Arm® Arm926EJ-S™ Processor-Based MPU, 800 MHz, MIPI DSI® or CSI-2, LVDS, RGB, 2D Graphics, Gigabit Ethernet with TSN, CAN-FD, Octal/Quad SPI, Crypto, PUF

SAM9X7 Series Data Sheet

www.microchip.com
Product Page Links

Introduction

The SAM9X7 Series microprocessors are high-performance and cost-optimized Arm926EJ-S CPU-based embedded microprocessors (MPU), running up to 800 MHz. The devices integrate powerful peripherals for connectivity and user interface applications, including MIPI DSI, LVDS, RGB and 2D graphics, MIPI-CSI-2, Gigabit Ethernet with TSN and CAN-FD. Advanced security functions are offered, such as tamper detection, secure boot, secure key storage, TRNG, PUF as well as high-performance crypto accelerators for AES and SHA. The SAM9X7 Series is supported by Microchip MPLAB®-X development tools, Harmony, Linux® distributions and the Ensemble Graphics Toolkit.

The SAM9X7 Series is available in ambient temperature ranges up to 105°C and is AEC-Q100 grade 2 qualified.

Reference Document

The SAM9X7 Series device conforms functionally to this data sheet, except for the anomalies described in the following document.

Type	Title	Literature No.	Available
Errata	SAM9X7 Series Silicon Errata and Data Sheet Clarification	DS80001082	www.microchip.com

Overview

1. Features

- CPU Running up to 800 MHz
  - Arm926EJ-S Arm Thumb® processor
  - 32-Kbyte data cache, 32-Kbyte instruction cache, Memory Management Unit (MMU)
- Memories
  - One 176-Kbyte internal ROM
    - 80-Kbyte internal ROM embedding a secure bootloader program supporting boot on NAND Flash, SD card, SPI or QSPI Flash; bootloader features selectable by OTP bits
    - 96-Kbyte ROM for NAND Flash BCH ECC table
  - One 64-Kbyte internal SRAM (SRAM0), single cycle access at system speed
  - DDR3(L)/DDR2 controller running at up to 266 MHz
  - External Bus Interface (EBI) supporting:
    - 16-bit 8/4-bank DDR3(L)/DDR2
    - 16-bit static memories
    - 8-bit NAND Flash with up to 24-bit programmable multi-bit error correcting code
  - One 10-Kbyte OTP memory for secure key storage with Emulation mode (OTP bits are emulated by a 4-Kbyte SRAM (SRAM1))
- System Running up to 266 MHz
  - Power-on reset cells, reset controller, shutdown controller, periodic interval timer, watchdog timer 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 (slow) and 12 MHz (fast)
  - Two crystal oscillators: 32.768 kHz (slow) and 20 to 50 MHz (fast)
  - One PLL for the system and one PLL optimized for USB high-speed operation (480 MHz)
    - One PLL for audio operations, with dedicated output clock
    - One PLL in LVDS I/F (LVDS usage only)
    - One PLL in MIPI D-PHY (MIPI DSI usage only)
  - One dual-port 16-channel DMA controller
  - Advanced interrupt controller and debug unit
  - JTAG port with disable bit in OTP memory
  - Two programmable clock output signals
- Low-Power Modes
  - Backup mode with RTC, eight 32-bit general purpose backup registers, and shutdown controller to control the external power supply
  - Clock generator and power management controller
  - Software-programmable ultra-low power modes: Very slow clock operating mode (ULP0), and no-clock operating mode (ULP1) with fast wake-up capabilities
  - Software programmable power optimization capabilities
- Peripherals

  - LCD controller with overlay, alpha-blending, rotation, scaling and color conversion; display size up to 1024x768 (XGA) with overlay (application-dependent); still images up to 1280x720 (720p)
  - RGB, LVDS, MIPI DSI interfaces (see Configuration Summary)
  - 2D graphics controller supporting fill BLT, copy BLT, transparent BLT, blend/alpha BLT, ROP4 BLT (raster operations) and command ring buffer
  - Image sensor controller with ITU-R BT; 601/656/1120 video interface support up to 5 Mpixels; support of raw Bayer 12, YCbCr, monochrome and JPEG compressed sensors up to 12 bits
    - MIPI CSI2 I/F support
    - 12-bit parallel I/F support
  - One high-speed USB device, three high-speed USB hosts with dedicated on-chip transceivers
  - One 10/100/1000 Mbps Ethernet Mac controller, with IEEE® -1588 and TSN support, RGMII and RMII support
  - Two 4-bit secure digital multimedia card controllers
  - Two CAN FD controllers with timestamping
  - One Quad/Octal SPI controller
  - Two 3-channel 32-bit timers/counters
  - Two high-resolution (64-bit) periodic interval timers
  - One synchronous serial controller
  - One inter-IC sound multi-channel controller with TDM support
  - One audio class D controller with single-ended or bridge-tied load connection to power stage
  - One 4-channel 16-bit PWM controller
  - Thirteen FLEXCOMs (USART, SPI and TWI/I2C)
  - One 8-channel, 12-bit, analog-to-digital converter with 4/5 wires resistive touchscreen support
- Hardware Cryptography
  - SHA (SHA1, SHA224, SHA256, SHA384, SHA512) and HMAC compliant with FIPS PUB 180
  - AES: 256-, 192-, 128-bit key algorithms compliant with FIPS PUB 197
  - AES/SHA tight coupling for IPsec hardware acceleration
  - TDES: 2-key or 3-key algorithms compliant with FIPS PUB 46
  - True random number generator compliant with NIST Special Publication 800-22 Test Suite and FIPS PUBs 140-2 and 140-3
  - Key bus providing private key transfers between AES, TDES, TRNG, OTPC
  - Physical Unclonable Function (PUF) including NIST SP 800-90B (DRNG) and embedding four Kbytes of SRAM (PUFSRAM)
- I/O Ports
  - Four parallel input/output controllers
  - Up to 106 programmable I/O lines multiplexed with up to four peripheral I/Os
  - Input change interrupt capability on each I/O line, optional Schmitt trigger input
  - Individually programmable open-drain, pull-up and pull-down resistors, synchronous output
  - 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
  - BGA power/ground ball assignment to provide optimum decoupling capacitors placement
- Operating Conditions
  - Junction temperature (T_J) range: -40°C to +125°C
  - SAM9X7x-I devices ambient temperature (T_A) range: -40°C to +85°C
  - SAM9X7x-V devices ambient temperature (T_A) range: -40°C to +105°C
- Qualification
  - AEC-Q100 Grade 2 ([-40°C to +105°C] ambient temperature) applies to -V/4PBVAO devices only
  - The AEC-Q006 set of tests applies, as only copper wire interconnections are used
- Packages
  - 11x11 mm^2, 0.65-mm pitch, 240-ball BGA optimized for standard class PCB layout (down to four layers)
  - 9x9 mm^2, 0.5-mm pitch, 256-ball BGA for space-constrained applications

2. Ordering Information

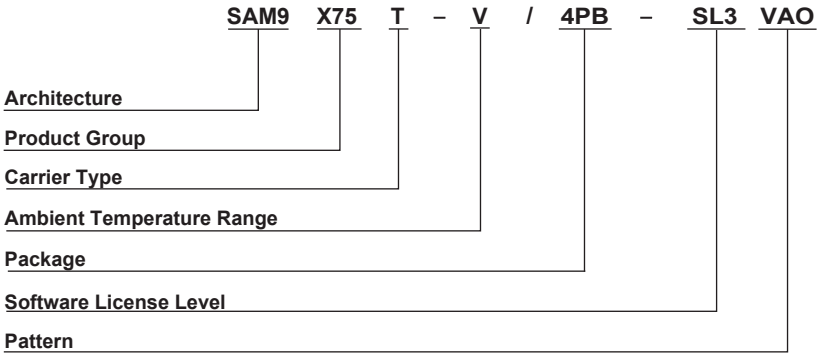
Ordering Code[1][2]	Tier	Package	Ambient Operating Temperature Range
SAM9X70(T)-I/4PB(-SLx)	Industrial	TFBGA240	-40°C to +85°C
SAM9X72(T)-I/4PB(-SLx)			
SAM9X75(T)-I/4PB(-SLx)			
SAM9X70(T)-V/4PB(-SLx)	Extended Industrial	TFBGA240	-40°C to +105°C
SAM9X72(T)-V/4PB(-SLx)			
SAM9X75(T)-V/4PB(-SLx)			
SAM9X70(T)-V/6GW(-SLx)	Extended Industrial	TFBGA256	-40°C to +105°C
SAM9X72(T)-V/6GW(-SLx)			
SAM9X75(T)-V/6GW(-SLx)			
SAM9X75(T)-V/4PB(-SLx)VAO	Automotive	TFBGA240	-40°C to +105°C

Notes:

1. For details on ordering codes, refer to Product Identification System.
2. For SL1, SL2 and SL3 device availability, contact a Microchip Sales representative.

3. Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture:	SAM9	= Arm926EJ-S Arm Thumb microprocessor
Product Group:	X70	= General purpose microprocessors with variable feature set
	X72	
	X75	
Carrier Type:	Blank	= Standard packaging (tray)
	T	= Tape and reel
Ambient Temperature Range:	I	= –40°C to +85°C (industrial)
	V	= –40°C to +105°C (extended industrial or automotive (AEC-Q100 grade 2))
Package:	4PB	= TFBGA240
	6GW	= TFBGA256
Software License Level:	Blank	= Standard
	SL1	= Level 1
	SL2	= Level 2
	SL3	= Level 3
Pattern:	Blank	= Standard device
	VAO	= Automotive flow/material without PPAP. Available upon request.
	Vxx	= Automotive flow/material with associated PPAP. Available upon request.

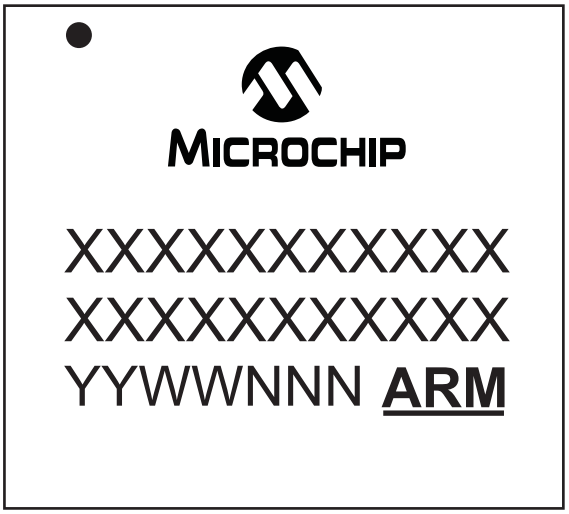
Example:

- SAM9X75T–V/4PB-SL3VAO = Arm926EJ-S Arm Thumb microprocessor, tape and reel, –40°C to +105°C temperature, 240-ball TFBGA package, software license level 3, Automotive flow/material without PPAP

Note: The Tape and Reel identifier and the Software License Level identifier only appear in the catalog part number description. These identifiers are used for ordering purposes and are not printed on the device package. Check with your Microchip Sales Office for package availability.

4. 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	SAM9X70 SAM9X72 SAM9X75
4	Temperature code / Packaging code, Jedec symbol	I/4PB e8 V/4PB e8 V/6GW e8
5	Lot traceability, Arm logo	YYWWNNN ARM

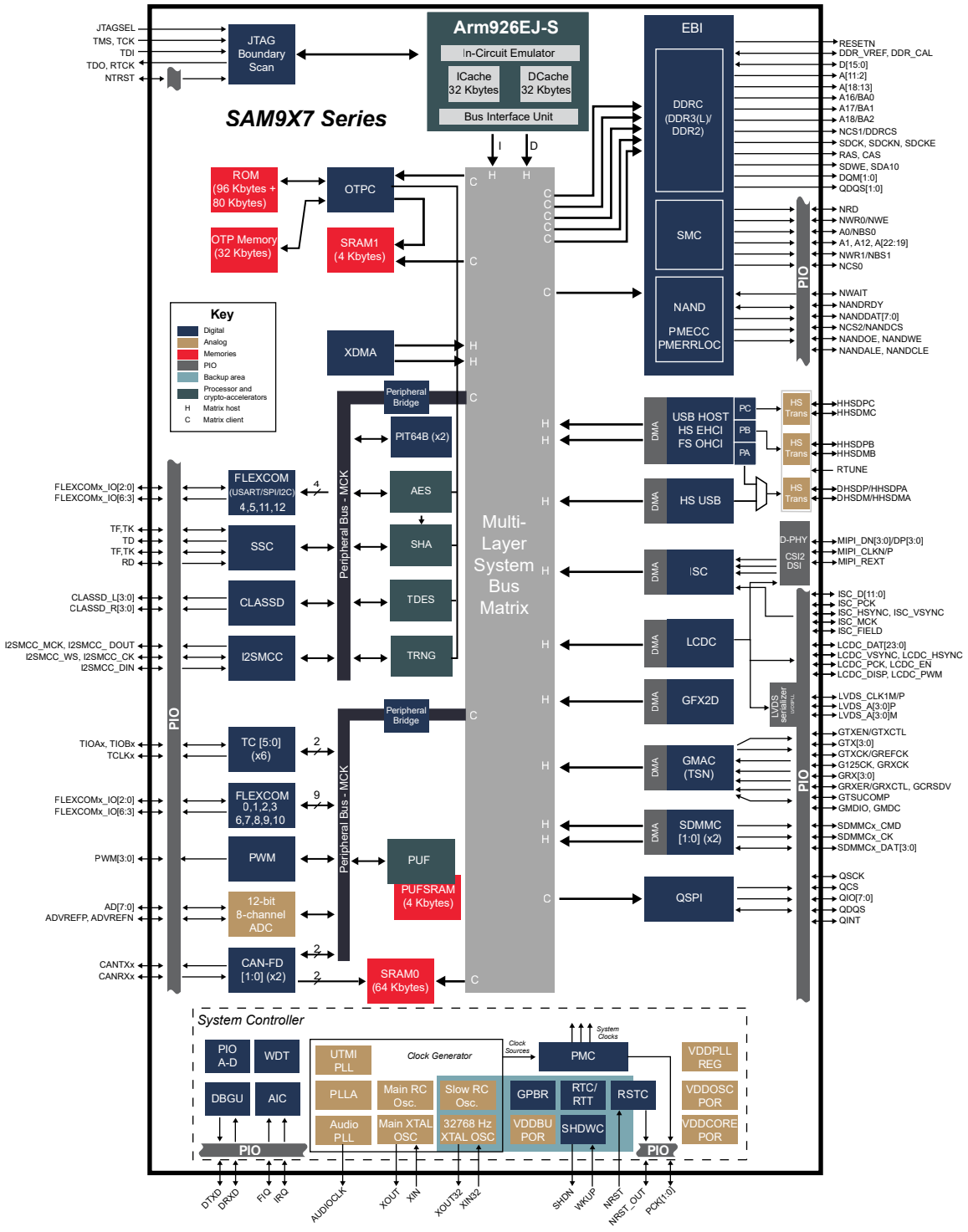
5. Configuration Summary

Table 5.1. SAM9X7 Series Configuration Summary

Feature	SAM9X70	SAM9X72	SAM9X75
Package	11x11 mm^2, 0.65-mm pitch, 240-ball BGA		
	9x9 mm^2, 0.5-mm pitch, 256-ball BGA		
Core	Arm926EJ-S @ 800 MHz		
SMC	Up to 16 bits		
MIPI D-PHY	No		Bidirectional, 4 lanes
PIOs	106		
SRAM0/SRAM1	64/4 Kbytes		
Quad/Octal SPI	1		
LCD, GFX2D	No	LVDS, parallel port	LVDS, MIPI DSI, parallel port
Camera Interface (ISC)	Parallel port		MIPI CSI2, parallel port
GMAC	10/100/1000 Mbps with IEEE-1588 and TSN support		
CAN-FD	No	2	
USB	3 (3 hosts or 2 hosts/1 device)		
UART/SPI/I2C	13		
SDIO/SD/MMC	2		
I2SMCC/SSC/CLASSD	1/1/1		
ADC Inputs	8 (8 channels)		
64-bit/32-bit Timer Counter Channels	2/6		
PWM	4 (PWMC)		
Cryptography	AES/TDES/SHA/TRNG		

6. Block Diagram

Figure 6.1. SAM9X7 Series Block Diagram



7. Signal Description

Table 7.1. Signal Description List

Signal Name	Function	Type	Comments	Active Level
Clocks, Oscillators and PLLs				
XIN	Main Crystal Oscillator Input	Input	–	–
XOUT	Main Crystal Oscillator Output	Output	–	–
XIN32	32.768 kHz Crystal Oscillator Input	Input	–	–
XOUT32	32.768 kHz Crystal Oscillator Output	Output	–	–
RTUNE	USB External Tune Resistor	Analog	–	–
PCK[1:0]	Programmable Clock Output	Output	–	–
AUDIOCLK	Audio Programmable Clock Output	Output	–	–
Shutdown, Wake-Up Logic				
SHDN	Shutdown Control	Output	–	–
WKUP0	Wake-Up Input	Input	–	–
ICE and JTAG				
TCK	Test Clock	Input	–	–
TDI	Test Data In	Input	–	–
TDO	Test Data Out	Output	–	–
TMS	Test Mode Select	Input	–	–
JTAGSEL	JTAG Selection	Input	–	–
RTCK	Return Test Clock	Output	–	–
Reset/Test				
NRST	External Reset Input	Input	–	Low
NRST_OUT	Reset Controller Output	Output	–	Low
TST	Test Mode Select	Input	–	–
NTRST	Test Reset Signal	Input	–	–
Debug Unit - DBGU				
DRXD	Debug Receive Data	Input	–	–
DTXD	Debug Transmit Data	Output	–	–
Advanced Interrupt Controller - AIC				
IRQ	External Interrupt Input	Input	–	–
FIQ	Fast Interrupt Input	Input	–	–
PIO Controller - PIOA - PIOB - PIOC - PIOD				
PA[31:0]	Parallel IO Controller A	I/O	–	–
PB[26:0]	Parallel IO Controller B	I/O	–	–
PC[31:0]	Parallel IO Controller C	I/O	–	–
PD[14:0]	Parallel IO Controller D	I/O	–	–
External Bus Interface - EBI				
A[22:0]	Address Bus	Output	–	–
NWAIT/NANDRDY	External Wait Signal/NAND Flash R/B Signal	Input	–	Low
Static Memory Controller - SMC				
NCS[2:0]	Chip Select Lines	Output	–	Low
NWR[1:0]	Write Signal	Output	–	Low
NRD	Read Signal	Output	–	Low
NWE	Write Enable	Output	–	Low

Table 7.1. Signal Description List (continued)

Signal Name	Function	Type	Comments	Active Level
NBS[1:0]	Byte Mask Signal	Output	–	Low
NAND Flash Controller				
NANDDAT[7:0]	NAND Flash I/O	I/O	–	–
NANDCS	NAND Flash Chip Select	Output	–	Low
NANDOE	NAND Flash Output Enable	Output	–	Low
NANDWE	NAND Flash Write Enable	Output	–	Low
NANDALE	NAND Address Latch Enable	Output	–	Low
NANDCLE	NAND Command Latch Enable	Output	–	Low
DDR2/DDR3(L) Controller				
SDCK	DRAM Clock	Output	–	–
SDCKN	DRAM Clock Bar	Output	–	–
SDCKE	DRAM Clock Enable	Output	–	High
DDRCS	DRAM Chip Select	Output	–	Low
BA[2:0]	Bank Select	Output	–	Low
SDWE	DRAM Write Enable	Output	–	Low
DDR_VREF	I/O Reference Voltage	I/O	–	–
DDR_CAL	Calibration Input	I/O	–	–
RAS - CAS	Row and Column Signal	Output	–	Low
A[22:0]	Address Bus	Output	–	–
SDA10	SDRAM Address 10 Line	Output	–	–
D[15:0]	Data Bus	I/O	–	–
DQS[1:0]	Positive Data Strobe	I/O	–	–
DQSN[1:0]	Negative Data Strobe (DDR2/3(L)-SDRAM only)	I/O	–	–
DQM[1:0]	Write Data Mask	Output	–	–
RESETN	DDR3-SDRAM Reset	Output	–	–
Secure Data Memory Card - SDMMCx [1:0]				
SDMMCx_CMD	SD Card/e.MMC Command Line	I/O	–	–
SDMMCx_CK	SD Card/e.MMC Clock Signal	Output	–	–
SDMMCx_DAT[3:0]	SD Card/e.MMC Data Lines	I/O	–	–
Flexible Serial Communication Controller - FLEXCOMx [12:0]				
FLEXCOMx_IO0	Transmit Data (TXD/MOSI/TWD)	I/O	–	–
FLEXCOMx_IO1	Receive Data (RXD/MISO/TWCK)	I/O	–	–
FLEXCOMx_IO2	Serial Clock (SCK/SPCK)	I/O	–	–
FLEXCOMx_IO3	Clear To Send/Peripheral Chip Select	I/O	–	–
FLEXCOMx_IO4	Request To Send/Peripheral Chip Select	Output	–	–
FLEXCOMx_IO5	Peripheral Chip Select	Output	–	–
FLEXCOMx_IO6	Peripheral Chip Select	Output	–	–
FLEXCOMx_IO7	LON Collision	Input	–	–
Synchronous Serial Controller - SSC				
TD	Transmit Data	Output	–	–
RD	Receive Data	Input	–	–
TK	Transmit Clock	I/O	–	–
RK	Receive Clock	I/O	–	–
TF	Transmit Frame Synchronization	I/O	–	–
RF	Receive Frame Synchronization	I/O	–	–

Table 7.1. Signal Description List (continued)

Signal Name	Function	Type	Comments	Active Level
Timer/Counter - TCx [5:0]				
TCLK[2:0]	External Clock Input	Input	–	–
TIOA[2:0]	I/O Line A	I/O	–	–
TIOB[2:0]	I/O Line B	I/O	–	–
Pulse Width Modulation Controller - PWMC				
PWM[3:0]	Pulse Width Modulation Output	Output	–	–
USB Host High Speed Port - UHPHS				
HHSDMA	USB Host Port A High Speed Data -	Analog	–	–
HHSDPA	USB Host Port A High Speed Data +	Analog	–	–
HHSDMB	USB Host Port B High Speed Data -	Analog	–	–
HHSDPB	USB Host Port B High Speed Data +	Analog	RTUNE	–
HHSDMC	USB Host Port C High Speed Data -	Analog	–	–
HHSDPC	USB Host Port C High Speed Data +	Analog	–	–
USB Device High Speed Port - UDPHS				
DHSDM	USB Device High Speed Data -	Analog	–	–
DHSDP	USB Device High Speed Data +	Analog	–	–
Gigabit Ethernet 10/100/1000 with IEEE-1588 and TSN (RGMII/RMII only) - GMAC				
GTXCK/GREFCK	Transmit Clock or Reference Clock	I/O	–	–
G125CK	125 MHz Reference Clock	Input	–	–
GRXCK	Receive Clock	Input	–	–
GTXEN/GTXCTL	Transmit Enable or Transmit Control	Output	–	–
GTX[3:0]	Transmit Data	Output	–	–
GCRSDV/GRXCTL	Receive Data Valid or Receive Control	Input	–	–
GRX[3:0]	Receive Data	Input	–	–
GRXER	Receive Error	Input	–	–
GMDC	Management Data Clock	Output	–	–
GMDIO	Management Data Input/Output	I/O	–	–
GTSUCOMP	TSU Timer Comparison Valid	Output	–	–
Analog-to-Digital Converter - ADC				
AD[7:0]	8 Analog Inputs	Input	–	–
ADTRG	ADC Trigger	Input	–	–
ADVREFN	ADC Negative Reference Voltage	Analog Input	–	–
ADVREFP	ADC Positive Reference Voltage	Analog Input	–	–
CAN Controller - CANx [1:0]				
CANRXx	CAN Receive	Input	–	–
CANTXx	CAN Transmit	Output	–	–
Class D Controller - CLASSD				
CLASSD_L0	Class D Controller Left Output 0	Output	–	–
CLASSD_L1	Class D Controller Left Output 1	Output	–	–
CLASSD_L2	Class D Controller Left Output 2	Output	–	–
CLASSD_L3	Class D Controller Left Output 3	Output	–	–
CLASSD_R0	Class D Controller Right Output 0	Output	–	–
CLASSD_R1	Class D Controller Right Output 1	Output	–	–
CLASSD_R2	Class D Controller Right Output 2	Output	–	–

Table 7.1. Signal Description List (continued)

Signal Name	Function	Type	Comments	Active Level
CLASSD_R3	Class D Controller Right Output 3	Output	–	–
Quad/Octal I/O SPI - QSPI				
QSCK	Quad IO SPI Serial Clock	Output	–	–
QCS	Quad IO SPI Chip Select	Output	–	–
QIO[7:0]	IO SPI I/O 0 to 7	I/O	–	–
QDQS	Octal IO Data Strobe	I/O	–	–
QINT	Interrupt	Input	–	–
Inter IC Sound Multi Channel Controller - I2SMCC				
I2SMCC_MCK	Main System Bus Clock	Output	–	–
I2SMCC_CK	Serial Clock	I/O	–	–
I2SMCC_WS	I2S Word Select	I/O	–	–
I2SMCC_DIN	Serial Data Input	Input	–	–
I2SMCC_DOUT	Serial Data Output	Output	–	–
MIPI D-PHY				
MIPI_DP[3:0] MIPI_DN[3:0]	MIPI D-PHY Differential Output Data Lane [3:0]	I/O	–	–
MIPI_CLKP MIPI_CLKN	MIPI D-PHY Differential Output Clock Lane	I/O	–	–
MIPI_REXT	Calibration Reference Resistor (4.02 KΩ E96)	I/O	–	–
Low Voltage Differential Signaling Controller (LVDS)				
LVDS_A[3:0]P LVDS_A[3:0]M	Differential LVDS Data Line Transceiver Output [3:0]	Output	–	–
LVDS_CLK1M LVDS_CLK1P	Differential LVDS Clock Line Transceiver Output	Output	–	–
Image Sensor Controller (ISC)				
ISC_MCK	Main System Bus Clock to Sensor	Output	–	–
ISC_PCK	Pixel Clock from Sensor	Input	–	–
ISC_D[11:0]	Data	Input	–	–
ISC_HSYNC	Horizontal Synchronization	Input	–	–
ISC_VSYNC	Vertical Synchronization	Input	–	–
ISC_FIELD	Field to Interface Video Streams	Input	–	–
LCD Controller (LCDC)				
LCDC_DAT[23:0]	Data Bus	Output	–	–
LCDC_PCK	Pixel Clock	Output	–	–
LCDC_HSYNC	Horizontal Synchronization	Output	–	–
LCDC_VSYNC	Vertical Synchronization	Output	–	–
LCDC_DEN	Data Enable	Output	–	–
LCDC_DISP	Display On/Off	Output	–	–
LCDC_PWM	PWM for Contrast Control	Output	–	–

8. Package and Pinout

8.1. Package

SAM9X7 Series devices are available in the following packages.

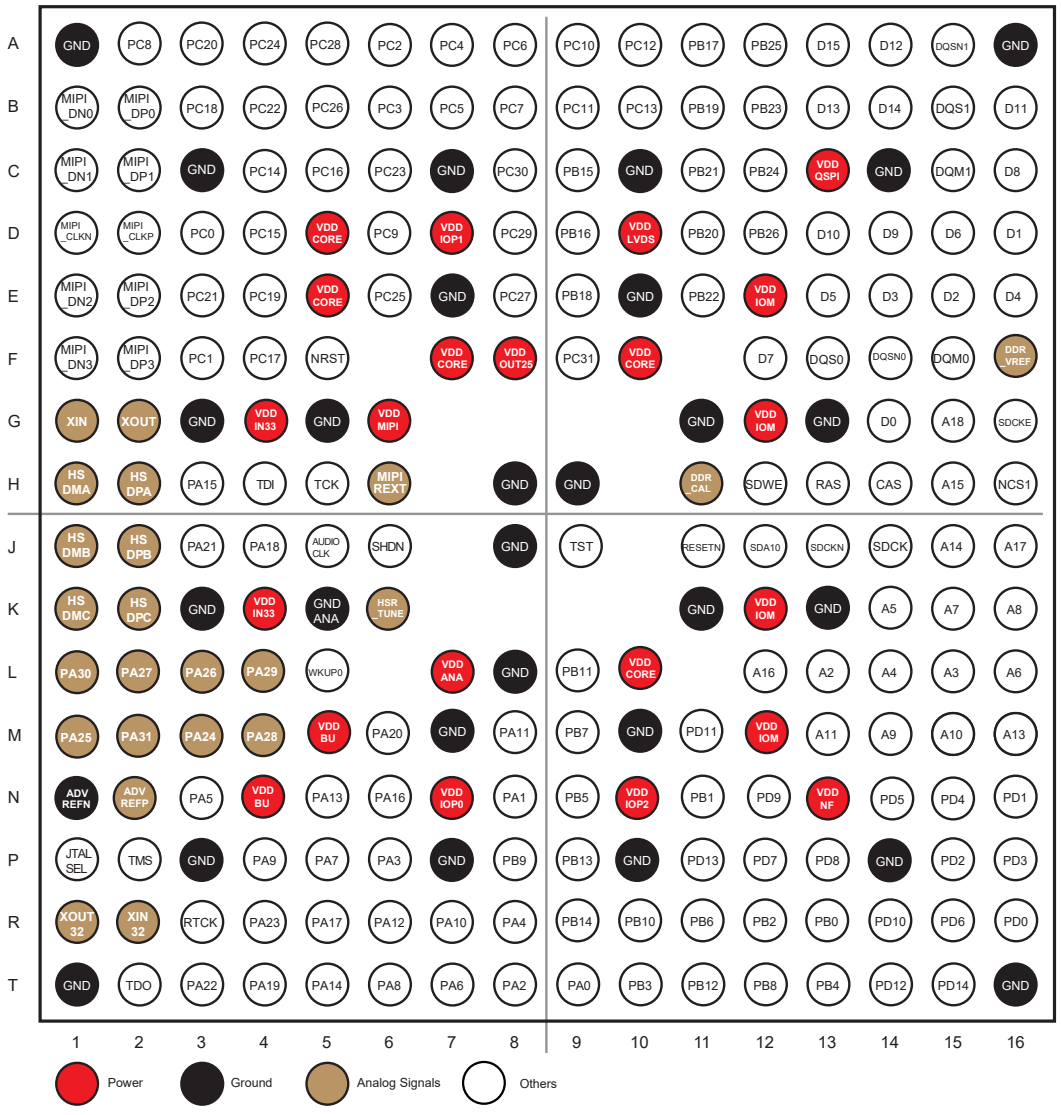
Package Name	Ball Count	Ball Pitch	Package Size
TFBGA240	240	0.65 mm	11 x 11 x 1.217 mm
TFBGA256	256	0.5 mm	9 x 9 x 1.2 mm

For further details, refer to Mechanical Characteristics.

8.2. Pinout

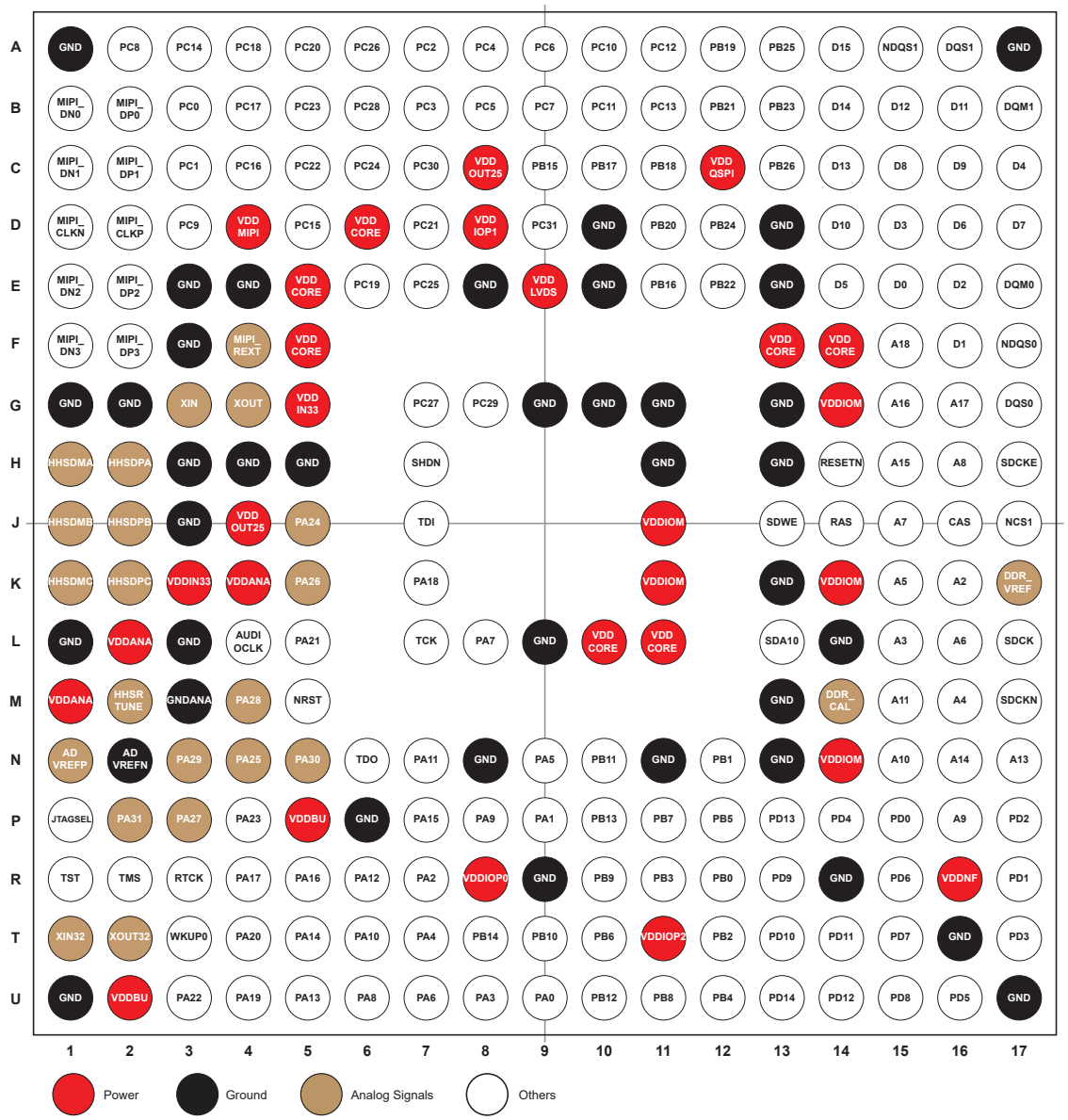
8.2.1. BGA240 Pinout

Figure 8.1. SAM9X7 Series BGA240 Pinout



8.2.2. BGA256 Pinout

Figure 8.2. SAM9X7 Series BGA256 Pinout



8.2.3. Pin Description

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 8.1. Pin Description[1]

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
T9	U9	VDDIOP0	GPIO	PA0	I/O	–	–	A	SDMMC0_DAT0	I/O	1	PIO, I, PU, ST
N8	P9	VDDIOP0	GPIO	PA1	I/O	–	–	A	SDMMC0_CMD	I/O	1	PIO, I, PU, ST
T8	R7	VDDIOP0	GPIO	PA2	I/O	WKUP1	–	A	SDMMC0_CK	I/O	1	PIO, I, PU, ST
P6	U8	VDDIOP0	GPIO	PA3	I/O	–	–	A	SDMMC0_DAT1	I/O	1	PIO, I, PU, ST
R8	T7	VDDIOP0	GPIO	PA4	I/O	–	–	A	SDMMC0_DAT2	I/O	1	PIO, I, PU, ST
N3	N9	VDDIOP0	GPIO	PA5	I/O	–	–	A	SDMMC0_DAT3	I/O	1	PIO, I, PU, ST
T7	U7	VDDIOP0	GPIO	PA6	I/O	–	–	A	FLEXCOM0_IO4	O	1	PIO, I, PU, ST
								B	SDMMC1_DAT1	I/O	1	
P5	L8	VDDIOP0	GPIO	PA7	I/O	WKUP2	–	A	FLEXCOM0_IO3	I/O	1	PIO, I, PU, ST
								B	SDMMC1_DAT2	I/O	1	
T6	U6	VDDIOP0	GPIO	PA8	I/O	WKUP3	–	A	FLEXCOM0_IO2	I/O	1	PIO, I, PU, ST
								B	SDMMC1_DAT3	I/O	1	
P4	P8	VDDIOP0	GPIO	PA9	I/O	–	–	A	FLEXCOM4_IO1	I/O	1,2	PIO, I, PU, ST
								B	SDMMC1_DAT0	I/O	1	
R7	T6	VDDIOP0	GPIO	PA10	I/O	–	–	A	FLEXCOM4_IO0	I/O	1,2	PIO, I, PU, ST
								B	SDMMC1_CMD	I/O	1	
M8	N7	VDDIOP0	GPIO	PA11	I/O	–	–	A	FLEXCOM4_IO2	I/O	1,2	PIO, I, PU, ST
								B	SDMMC1_CK	I/O	1	
R6	R6	VDDIOP0	GPIO	PA12	I/O	–	–	A	FLEXCOM4_IO3	I/O	1,2	PIO, I, PU, ST
								C	FLEXCOM5_IO4	O	1	
N5	U5	VDDIOP0	GPIO	PA13	I/O	–	–	A	FLEXCOM2_IO0	I/O	1	PIO, I, PU, ST
								B	FLEXCOM4_IO4	O	1	
T5	T5	VDDIOP0	GPIO	PA14	I/O	–	–	A	FLEXCOM2_IO1	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO3	I/O	1,2	
								C	FLEXCOM4_IO5	O	1	
H3	P7	VDDIOP0	GPIO	PA15	I/O	–	–	A	TIOA0	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO1	I/O	1,2	
								C	CLASSD_R0	O	1	
N6	R5	VDDIOP0	GPIO	PA16	I/O	–	–	A	TIOA1	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO0	I/O	1,2	
								C	CLASSD_R1	O	1	
R5	R4	VDDIOP0	GPIO	PA17	I/O	–	–	A	TIOA2	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO2	I/O	1,2	
								C	CLASSD_R2	O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
J4	K7	VDDIOP0	GPIO	PA18	I/O	–	–	A	TCLK0	I	1	PIO, I, PU, ST
								B	TK	I/O	1	
								C	CLASSD_L0	O	1	
T4	U4	VDDIOP0	GPIO	PA19	I/O	–	–	A	TCLK1	I	1	PIO, I, PU, ST
								B	TF	I/O	1	
								C	CLASSD_L1	O	1	
M6	T4	VDDIOP0	GPIO	PA20	I/O	WKUP4	–	A	TCLK2	I	1	PIO, I, PU, ST
								B	TD	O	1	
								C	CLASSD_L2	O	1	
J3	L5	VDDIOP0	GPIO	PA21	I/O	–	–	A	TIOB0	I/O	1	PIO, I, PU, ST
								B	RD	I	1	
								C	CLASSD_L3	O	1	
T3	U3	VDDIOP0	GPIO	PA22	I/O	–	–	A	TIOB1	I/O	1	PIO, I, PU, ST
								B	RK	I/O	1	
								C	CLASSD_R3	O	1	
R4	P4	VDDIOP0	GPIO	PA23	I/O	–	–	A	TIOB2	I/O	1	PIO, I, PU, ST
								B	RF	I/O	1	
								C	FLEXCOM2_IO7	I	1	
M3	J5	VDDANA	GPIO	PA24	I/O	AD0	–	A	FLEXCOM6_IO0	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO6	O	1	
M1	N4	VDDANA	GPIO	PA25	I/O	AD1	–	A	FLEXCOM6_IO1	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO5	O	1	
L3	K5	VDDANA	GPIO	PA26	I/O	AD2	–	A	DRXD	I	1	PIO, I, PU, ST
								B	CANRX0	I	1	
L2	P3	VDDANA	GPIO	PA27	I/O	AD3	–	A	DTXD	O	1	PIO, I, PU, ST
								B	CANTX0	O	1	
M4	M4	VDDANA	GPIO	PA28	I/O	AD4	–	A	FLEXCOM1_IO0	I/O	1	PIO, I, PU, ST
								B	CANTX1	O	1	
L4	N3	VDDANA	GPIO	PA29	I/O	AD5	–	A	FLEXCOM1_IO1	I/O	1	PIO, I, PU, ST
								B	CANRX1	I	1	
L1	N5	VDDANA	GPIO	PA30	I/O	AD6	–	A	FLEXCOM0_IO0	I/O	1	PIO, I, PU, ST
								B	FLEXCOM5_IO4	O	2	
								C	FLEXCOM4_IO4	O	2	
M2	P2	VDDANA	GPIO	PA31	I/O	AD7	–	A	FLEXCOM0_IO1	I/O	1	PIO, I, PU, ST
								B	FLEXCOM4_IO5	O	2	
								C	GTSUCOMP	O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
R13	R12	VDDIOP2	GPIO	PB0	I/O	WKUP5	–	A	GRX2	I	1	PIO, I, PU, ST
								B	FLEXCOM2_IO4	O	1	
								C	GRXER	I	1	
N11	N12	VDDIOP2	GPIO	PB1	I/O	–	–	A	GRX3	I	1	PIO, I, PU, ST
								B	FLEXCOM2_IO3	I/O	1	
R12	T12	VDDIOP2	GPIO	PB2	I/O	–	–	A	G125CK	I	1	PIO, I, PU, ST
								B	FLEXCOM2_IO2	I/O	1	
T10	R11	VDDIOP2	GPIO	PB3	I/O	WKUP6	–	A	GCRSDV/GRXCTL	I	1	PIO, I, PU, ST
								B	FLEXCOM4_IO6	O	1	
T13	U12	VDDIOP2	GPIO	PB4	I/O	–	–	A	GTX2	O	1	PIO, I, PU, ST
								B	FLEXCOM8_IO0	I/O	1	
N9	P12	VDDIOP2	GPIO	PB5	I/O	–	–	A	GTX3	O	1	PIO, I, PU, ST
								B	FLEXCOM8_IO1	I/O	1	
R11	T10	VDDIOP2	GPIO	PB6	I/O	–	–	A	GTXCK/GREFCK	I/O	1	PIO, I, PU, ST
								B	FLEXCOM0_IO7	I	1	
M9	P11	VDDIOP2	GPIO	PB7	I/O	–	–	A	GTXEN/GTXCTL	O	1	PIO, I, PU, ST
								C	FLEXCOM6_IO2	I/O	1	
T12	U11	VDDIOP2	GPIO	PB8	I/O	–	–	A	GRXCK	I	1	PIO, I, PU, ST
								C	FLEXCOM6_IO3	I/O	1	
P8	R10	VDDIOP2	GPIO	PB9	I/O	–	–	A	GMDIO	I/O	1	PIO, I, PU, ST
								B	PCK1	O	1	
								C	FLEXCOM6_IO4	O	1	
R10	T9	VDDIOP2	GPIO	PB10	I/O	–	–	A	GMDC	O	1	PIO, I, PU, ST
								B	PCK0	O	1	
								C	FLEXCOM8_IO2	I/O	1	
L9	N10	VDDIOP2	GPIO	PB11	I/O	–	–	A	GRX0	I	1	PIO, I, PU, ST
								B	PWM0	O	2	
								C	FLEXCOM8_IO3	I/O	1	
T11	U10	VDDIOP2	GPIO	PB12	I/O	–	–	A	GRX1	I	1	PIO, I, PU, ST
								B	PWM1	O	2	
								C	FLEXCOM8_IO4	O	1	
P9	P10	VDDIOP2	GPIO	PB13	I/O	–	–	A	GTX0	O	1	PIO, I, PU, ST
							–	B	PWM2	O	2	
R9	T8	VDDIOP2	GPIO	PB14	I/O	–	–	A	GTX1	O	1	PIO, I, PU, ST
								B	PWM3	O	2	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
C9	C9	VDDQSPI	GPIO	PB15	I/O	–	–	A	QIO5	I/O	1	PIO, I, PU, ST
								B	–	–	–	
								C	FLEXCOM11_IO0	I/O	1	
								D	I2SMCC_WS	I/O	1	
D9	E11	VDDQSPI	GPIO	PB16	I/O	–	–	A	QIO6	I/O	1	PIO, I, PU, ST
								B	–	–	–	
								C	FLEXCOM11_IO1	I/O	1	
								D	I2SMCC_DIN	I	1	
A11	C10	VDDQSPI	GPIO	PB17	I/O	–	–	A	QIO7	I/O	1	PIO, I, PU, ST
								C	FLEXCOM12_IO0	I/O	1	
								D	I2SMCC_DOUT	O	1	
E9	C11	VDDQSPI	GPIO	PB18	I/O	WKUP7	–	A	QDQS	I	1	PIO, I, PU, ST
								B	ADTRG	I	1	
								C	FLEXCOM12_IO1	I/O	1	
								D	IRQ	I	1	
B11	A12	VDDQSPI	GPIO	PB19	I/O	–	–	A	QSCK	O	1	PIO, I, PU, ST
								C	FLEXCOM12_IO2	I/O	1	
D11	D11	VDDQSPI	GPIO	PB20	I/O	–	–	A	QCS	O	1	PIO, I, PU, ST
								C	FLEXCOM12_IO3	I/O	1	
C11	B12	VDDQSPI	GPIO	PB21	I/O	–	–	A	QIO0	I/O	1	PIO, I, PU, ST
								C	FLEXCOM12_IO4	O	1	
E11	E12	VDDQSPI	GPIO	PB22	I/O	–	–	A	QIO1	I/O	1	PIO, I, PU, ST
								C	FLEXCOM9_IO2	I/O	1	
B12	B13	VDDQSPI	GPIO	PB23	I/O	–	–	A	QIO2	I/O	1	PIO, I, PU, ST
								C	FLEXCOM9_IO3	I/O	1	
C12	D12	VDDQSPI	GPIO	PB24	I/O	–	–	A	QIO3	I/O	1	PIO, I, PU, ST
								C	FLEXCOM9_IO4	O	1	
A12	A13	VDDQSPI	GPIO	PB25	I/O	WKUP8	–	A	QINT	I	1	PIO, I, PU, ST
								D	I2SMCC_MCK	O	1	
D12	C13	VDDQSPI	GPIO	PB26	I/O	–	–	A	QIO4	I/O	1	PIO, I, PU, ST
								D	I2SMCC_CK	I/O	1	
D3	B3	VDDIOP1	GPIO	PC0	I/O	–	–	A	LCDC_DAT0	O	1	PIO, I, PU, ST
								B	ISC_D0	I	1	
								C	FLEXCOM7_IO0	I/O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
F3	C3	VDDIOP1	GPIO	PC1	I/O	–	–	A	LCDC_DAT1	O	1	PIO, I, PU, ST
								B	ISC_D1	I	1	
								C	FLEXCOM7_IO1	I/O	1	
A6	A7	VDDLVDS	GPIO	PC2	I/O	LVDS_A0M	–	A	LCDC_DAT2	O	1	PIO, I, PU, ST
								B	ISC_D2	I	1	
								C	TIOA3	I/O	1	
B6	B7	VDDLVDS	GPIO	PC3	I/O	LVDS_A0P	–	A	LCDC_DAT3	O	1	PIO, I, PU, ST
								B	ISC_D3	I	1	
								C	TIOB3	I/O	1	
A7	A8	VDDLVDS	GPIO	PC4	I/O	LVDS_A1M	–	A	LCDC_DAT4	O	1	PIO, I, PU, ST
								B	ISC_D4	I	1	
								C	TCLK3	I	1	
B7	B8	VDDLVDS	GPIO	PC5	I/O	LVDS_A1P	–	A	LCDC_DAT5	O	1	PIO, I, PU, ST
								B	ISC_D5	I	1	
								C	TIOA4	I/O	1	
A8	A9	VDDLVDS	GPIO	PC6	I/O	LVDS_A2M	–	A	LCDC_DAT6	O	1	PIO, I, PU, ST
								B	ISC_D6	I	1	
								C	TIOB4	I/O	1	
B8	B9	VDDLVDS	GPIO	PC7	I/O	LVDS_A2P	–	A	LCDC_DAT7	O	1	PIO, I, PU, ST
								B	ISC_D7	I	1	
								C	TCLK4	I	1	
A2	A2	VDDIOP1	GPIO	PC8	I/O	–	–	A	LCDC_DAT8	O	1	PIO, I, PU, ST
								B	ISC_D8	I	1	
								C	FLEXCOM9_IO0	I/O	1	
D6	D3	VDDIOP1	GPIO	PC9	I/O	–	–	A	LCDC_DAT9	O	1	PIO, I, PU, ST
								B	ISC_D9	I	1	
								C	FLEXCOM9_IO1	I/O	1	
A9	A10	VDDLVDS	GPIO	PC10	I/O	LVDS_CLK1M	–	A	LCDC_DAT10	O	1	PIO, I, PU, ST
								B	ISC_D10	I	1	
								C	PWM0	O	3	
B9	B10	VDDLVDS	GPIO	PC11	I/O	LVDS_CLK1P	–	A	LCDC_DAT11	O	1	PIO, I, PU, ST
								B	ISC_D11	I	1	
								C	PWM1	O	3	
A10	A11	VDDLVDS	GPIO	PC12	I/O	LVDS_A3M	–	A	LCDC_DAT12	O	1	PIO, I, PU, ST
								B	ISC_PCK	I	1	
								C	TIOA5	I/O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B10	B11	VDDLVDS	GPIO	PC13	I/O	LVDS_A3P	–	A	LCDC_DAT13	O	1	PIO, I, PU, ST
								B	ISC_VSYNC	I	1	
								C	TIOB5	I/O	1	
C4	A3	VDDIOP1	GPIO	PC14	I/O	–	–	A	LCDC_DAT14	O	1	PIO, I, PU, ST
								B	ISC_HSYNC	I	1	
								C	TCLK5	I	1	
D4	D5	VDDIOP1	GPIO	PC15	I/O	–	–	A	LCDC_DAT15	O	1	PIO, I, PU, ST
								B	ISC_MCK	O	1	
								C	PCK0	O	2	
C5	C4	VDDIOP1	GPIO	PC16	I/O	–	–	A	LCDC_DAT16	O	1	PIO, I, PU, ST
								B	ISC_FIELD	I	1	
								C	FLEXCOM10_IO0	I/O	1	
F4	B4	VDDIOP1	GPIO	PC17	I/O	–	–	A	LCDC_DAT17	O	1	PIO, I, PU, ST
								B	FLEXCOM1_IO7	I	1	
								C	FLEXCOM10_IO1	I/O	1	
B3	A4	VDDIOP1	GPIO	PC18	I/O	–	–	A	LCDC_DAT18	O	1	PIO, I, PU, ST
								B	FLEXCOM10_IO2	I/O	1	
								C	PWM0	O	1	
E4	E6	VDDIOP1	GPIO	PC19	I/O	–	–	A	LCDC_DAT19	O	1	PIO, I, PU, ST
								B	FLEXCOM10_IO3	I/O	1	
								C	PWM1	O	1	
A3	A5	VDDIOP1	GPIO	PC20	I/O	–	–	A	LCDC_DAT20	O	1	PIO, I, PU, ST
								B	FLEXCOM10_IO4	O	1	
								C	PWM2	O	1	
E3	D7	VDDIOP1	GPIO	PC21	I/O	–	–	A	LCDC_DAT21	O	1	PIO, I, PU, ST
								C	PWM3	O	1	
B4	C5	VDDIOP1	GPIO	PC22	I/O	–	–	A	LCDC_DAT22	O	1	PIO, I, PU, ST
								B	FLEXCOM3_IO0	I/O	1	
C6	B5	VDDIOP1	GPIO	PC23	I/O	WKUP9	–	A	LCDC_DAT23	O	1	PIO, I, PU, ST
								B	FLEXCOM3_IO1	I/O	1	
A4	C6	VDDIOP1	GPIO	PC24	I/O	WKUP10	–	A	LCDC_DISP	O	1	PIO, I, PU, ST
								B	FLEXCOM3_IO4	O	1	
E6	E7	VDDIOP1	GPIO	PC25	I/O	WKUP12	–	A	NTRST	I	1	NRST_OUT, O, PD
								B	FLEXCOM3_IO3	I/O	1	
								C	NRST_OUT	O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
B5	A6	VDDIOP1	GPIO	PC26	I/O	WKUP13	–	A	LCDC_PWM	O	1	PIO, I, PU, ST
								B	FLEXCOM3_IO2	I/O	1	
E8	G7	VDDIOP1	GPIO	PC27	I/O	–	–	A	LCDC_VSYNC	O	1	PIO, I, PU, ST
								C	FLEXCOM1_IO4	O	1	
A5	B6	VDDIOP1	GPIO	PC28	I/O	–	–	A	LCDC_HSYNC	O	1	PIO, I, PU, ST
								C	FLEXCOM1_IO3	I/O	1	
D8	G8	VDDIOP1	GPIO	PC29	I/O	–	–	A	LCDC_DEN	O	1	PIO, I, PU, ST
								C	FLEXCOM1_IO2	I/O	1	
C8	C7	VDDIOP1	GPIO	PC30	I/O	–	–	A	LCDC_PCK	O	1	PIO, I, PU, ST
								C	FLEXCOM3_IO7	I	1	
F9	D9	VDDIOP1	GPIO	PC31	I/O	WKUP11	–	A	FIQ	I	1	PIO, I, PU, ST
							–	C	PCK1	O	2	
R16	P15	VDDNF	GPIO	PD0	I/O	–	–	A	NANDOE	O	1	PIO, I, PU
								C	FLEXCOM7_IO2	I/O	1	
N16	R17	VDDNF	GPIO	PD1	I/O	–	–	A	NANDWE	O	1	PIO, I, PU
								C	FLEXCOM7_IO3	I/O	1	
P15	P17	VDDNF	GPIO	PD2	I/O	–	–	A	A21/NANDALE	O	1	A21,O, PD
								C	FLEXCOM7_IO4	O	1	
P16	T17	VDDNF	GPIO	PD3	I/O	–	–	A	A22/NANDCLE	O	1	A22,O, PD
								C	FLEXCOM11_IO2	I/O	1	
N15	P14	VDDNF	GPIO	PD4	I/O	–	–	A	NCS2/NANDCS	O	1	PIO, I, PU
								C	FLEXCOM11_IO3	I/O	1	
N14	U16	VDDNF	GPIO	PD5	I/O	–	–	A	–			PIO, I, PU
								B	NCS0	O	1	
								C	FLEXCOM11_IO4	O	1	
R15	R15	VDDNF	GPIO	PD6	I/O	–	–	A	NANDDAT0	I/O	1	PIO, I, PU
								B	A1	O	1	
P12	T15	VDDNF	GPIO	PD7	I/O	–	–	A	NANDDAT1	I/O	1	PIO, I, PU
								B	A12	O	1	
P13	U15	VDDNF	GPIO	PD8	I/O	–	–	A	NANDDAT2	I/O	1	PIO, I, PU
								B	A19	O	1	
N12	R13	VDDNF	GPIO	PD9	I/O	–	–	A	NANDDAT3	I/O	1	PIO, I, PU
								B	A20	O	1	
R14	T13	VDDNF	GPIO	PD10	I/O	–	–	A	NANDDAT4	I/O	1	PIO, I, PU
								B	NRD	O	1	

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
M11	T14	VDDNF	GPIO	PD11	I/O	–	–	A	NANDDAT5	I/O	1	PIO, I, PU
								B	NWR0/NWE	O	1	
T14	U14	VDDNF	GPIO	PD12	I/O	–	–	A	NANDDAT6	I/O	1	PIO, I, PU
								B	A0/NBS0	O	1	
P11	P13	VDDNF	GPIO	PD13	I/O	–	–	A	NANDDAT7	I/O	1	PIO, I, PU
								B	NWR1/NBS1	O	1	
T15	U13	VDDNF	GPIO	PD14	I/O	–	–	A	NWAIT/NANDRDY	I	1	PIO, I, PU
E12	G14	VDDIOM	Power	VDDIOM	I	–	–	–	–	–	–	–
G12	J11	VDDIOM	Power	VDDIOM	I	–	–	–	–	–	–	–
K12	K11	VDDIOM	Power	VDDIOM	I	–	–	–	–	–	–	–
M12	K14	VDDIOM	Power	VDDIOM	I	–	–	–	–	–	–	–
G6	D4	VDDMIPI	Power	VDDMIPI	I	–	–	–	–	–	–	–
D10	E9	VDDLVDS	Power	VDDLVDS	I	–	–	–	–	–	–	–
N13	R16	VDDNF	Power	VDDNF	I	–	–	–	–	–	–	–
N7	R8	VDDIOP0	Power	VDDIOP0	I	–	–	–	–	–	–	–
D7	D8	VDDIOP1	Power	VDDIOP1	I	–	–	–	–	–	–	–
N10	T11	VDDIOP2	Power	VDDIOP2	I	–	–	–	–	–	–	–
C13	C12	VDDQSPI	Power	VDDQSPI	I	–	–	–	–	–	–	–
N4	U2	VDDBU	Power	VDDBU	I	–	–	–	–	–	–	–
M5	P5	VDDBU	Power	VDDBU	I	–	–	–	–	–	–	–
L7	K4	VDDANA	Power	VDDANA	I	–	–	–	–	–	–	–
K5	M3	GNDANA	Ground	GNDANA	I	–	–	–	–	–	–	–
F8	C8	VDDOUT25	Power	VDDOUT25	I	–	–	–	–	–	–	–
G4	G5	VDDIN33	Power	VDDIN33	I	–	–	–	–	–	–	–
K4	K3	VDDIN33	Power	VDDIN33	I	–	–	–	–	–	–	–
D5	D6	VDDCORE	Power	VDDCORE	I	–	–	–	–	–	–	–
E5	E5	VDDCORE	Power	VDDCORE	I	–	–	–	–	–	–	–
F7	F5	VDDCORE	Power	VDDCORE	I	–	–	–	–	–	–	–
F10	F13	VDDCORE	Power	VDDCORE	I	–	–	–	–	–	–	–
L10	F14	VDDCORE	Power	VDDCORE	I	–	–	–	–	–	–	–
A1	A1	GND	Ground	GND	I	–	–	–	–	–	–	–
T1	J3	GND	Ground	GND	I	–	–	–	–	–	–	–
C3	D10	GND	Ground	GND	I	–	–	–	–	–	–	–
G3	H3	GND	Ground	GND	I	–	–	–	–	–	–	–
K3	L1	GND	Ground	GND	I	–	–	–	–	–	–	–
P3	H4	GND	Ground	GND	I	–	–	–	–	–	–	–

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
G5	F3	GND	Ground	GND	I	–	–	–	–	–	–	–
C7	D13	GND	Ground	GND	I	–	–	–	–	–	–	–
E7	E4	GND	Ground	GND	I	–	–	–	–	–	–	–
M7	G11	GND	Ground	GND	I	–	–	–	–	–	–	–
P7	H5	GND	Ground	GND	I	–	–	–	–	–	–	–
H8	–	GND	Ground	GND	I	–	–	–	–	–	–	–
J8	–	GND	Ground	GND	I	–	–	–	–	–	–	–
L8	G10	GND	Ground	GND	I	–	–	–	–	–	–	–
H9	–	GND	Ground	GND	I	–	–	–	–	–	–	–
C10	E10	GND	Ground	GND	I	–	–	–	–	–	–	–
E10	E8	GND	Ground	GND	I	–	–	–	–	–	–	–
M10	G13	GND	Ground	GND	I	–	–	–	–	–	–	–
P10	H11	GND	Ground	GND	I	–	–	–	–	–	–	–
G11	E13	GND	Ground	GND	I	–	–	–	–	–	–	–
K11	G2	GND	Ground	GND	I	–	–	–	–	–	–	–
G13	G1	GND	Ground	GND	I	–	–	–	–	–	–	–
K13	G9	GND	Ground	GND	I	–	–	–	–	–	–	–
C14	E3	GND	Ground	GND	I	–	–	–	–	–	–	–
P14	H13	GND	Ground	GND	I	–	–	–	–	–	–	–
A16	–	GND	Ground	GND	I	–	–	–	–	–	–	–
T16	K13	GND	Ground	GND	I	–	–	–	–	–	–	–
G14	E15	VDDIOM	DDRIO	D0	–	–	–	–	–	–	–	O, PD
D16	F16	VDDIOM	DDRIO	D1	–	–	–	–	–	–	–	O, PD
E15	E16	VDDIOM	DDRIO	D2	–	–	–	–	–	–	–	O, PD
E14	D15	VDDIOM	DDRIO	D3	–	–	–	–	–	–	–	O, PD
E16	C17	VDDIOM	DDRIO	D4	–	–	–	–	–	–	–	O, PD
E13	E14	VDDIOM	DDRIO	D5	–	–	–	–	–	–	–	O, PD
D15	D16	VDDIOM	DDRIO	D6	–	–	–	–	–	–	–	O, PD
F12	D17	VDDIOM	DDRIO	D7	–	–	–	–	–	–	–	O, PD
C16	C15	VDDIOM	DDRIO	D8	–	–	–	–	–	–	–	O, PD
D14	C16	VDDIOM	DDRIO	D9	–	–	–	–	–	–	–	O, PD
D13	D14	VDDIOM	DDRIO	D10	–	–	–	–	–	–	–	O, PD
B16	B16	VDDIOM	DDRIO	D11	–	–	–	–	–	–	–	O, PD
A14	B15	VDDIOM	DDRIO	D12	–	–	–	–	–	–	–	O, PD
B13	C14	VDDIOM	DDRIO	D13	–	–	–	–	–	–	–	O, PD
B14	B14	VDDIOM	DDRIO	D14	–	–	–	–	–	–	–	O, PD

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
A13	A14	VDDIOM	DDRIO	D15	–	–	–	–	–	–	–	O, PD
L13	K16	VDDIOM	DDRIO	A2	–	–	–	–	–	–	–	O, PD
L15	L15	VDDIOM	DDRIO	A3	–	–	–	–	–	–	–	O, PD
L14	M16	VDDIOM	DDRIO	A4	–	–	–	–	–	–	–	O, PD
K14	K15	VDDIOM	DDRIO	A5	–	–	–	–	–	–	–	O, PD
L16	L16	VDDIOM	DDRIO	A6	–	–	–	–	–	–	–	O, PD
K15	J15	VDDIOM	DDRIO	A7	–	–	–	–	–	–	–	O, PD
K16	H16	VDDIOM	DDRIO	A8	–	–	–	–	–	–	–	O, PD
M14	P16	VDDIOM	DDRIO	A9	–	–	–	–	–	–	–	O, PD
M15	N15	VDDIOM	DDRIO	A10	–	–	–	–	–	–	–	O, PD
M13	M15	VDDIOM	DDRIO	A11	–	–	–	–	–	–	–	O, PD
M16	N17	VDDIOM	DDRIO	A13	–	–	–	–	–	–	–	O, PD
J15	N16	VDDIOM	DDRIO	A14	–	–	–	–	–	–	–	O, PD
H15	H15	VDDIOM	DDRIO	A15	–	–	–	–	–	–	–	O, PD
L12	G15	VDDIOM	DDRIO	A16	–	BA0	–	–	–	–	–	O, PD
J16	G16	VDDIOM	DDRIO	A17	–	BA1	–	–	–	–	–	O, PD
G15	F15	VDDIOM	DDRIO	A18	–	BA2	–	–	–	–	–	O, PD
H16	J17	VDDIOM	DDRIO	NCS1	–	DDRCS	–	–	–	–	–	O, PU
J14	L17	VDDIOM	DDRIO	SDCK	–	–	–	–	–	–	–	O, PD
J13	M17	VDDIOM	DDRIO	SDCKN	–	–	–	–	–	–	–	O, PU
G16	H17	VDDIOM	DDRIO	SDCKE	–	–	–	–	–	–	–	O, PU
H13	J14	VDDIOM	DDRIO	RAS	–	–	–	–	–	–	–	O, PU
H14	J16	VDDIOM	DDRIO	CAS	–	–	–	–	–	–	–	O, PU
H12	J13	VDDIOM	DDRIO	SDWE	–	–	–	–	–	–	–	O, PU
J12	L13	VDDIOM	DDRIO	SDA10	–	–	–	–	–	–	–	O, PU
F15	E17	VDDIOM	DDRIO	DQM0	–	–	–	–	–	–	–	O, PU
C15	B17	VDDIOM	DDRIO	DQM1	–	–	–	–	–	–	–	O, PU
F13	G17	VDDIOM	DDRIO	DQS0	–	–	–	–	–	–	–	O, PD
F14	F17	VDDIOM	DDRIO	NDQS0	–	–	–	–	–	–	–	O, PU
B15	A16	VDDIOM	DDRIO	DQS1	–	–	–	–	–	–	–	O, PD
A15	A15	VDDIOM	DDRIO	NDQS1	–	–	–	–	–	–	–	O, PU
H11	M14	VDDIOM	analog	DDR_CAL	I	–	–	–	–	–	–	I
F16	K17	VDDIOM	analog	DDR_VREF	I	–	–	–	–	–	–	I
J11	H14	VDDIOM	DDRIO	RESETN	–	–	–	–	–	–	–	O
N2	N1	VDDANA	GPIO	ADVREFP	–	–	–	–	–	–	–	I
N1	N2	VDDANA	GPIO	ADVREFN	–	–	–	–	–	–	–	I

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
K6	M2	VDDIN33	USBHS	HHSRTUNE	–	–	–	–	–	–	–	I
H2	H2	VDDIN33	USBHS	HHSDPA	–	DHSDP	–	–	–	–	–	O, PD
H1	H1	VDDIN33	USBHS	HHSDMA	–	DHSDM	–	–	–	–	–	O, PD
J2	J2	VDDIN33	USBHS	HHSDPB	–	–	–	–	–	–	–	O, PD
J1	J1	VDDIN33	USBHS	HHSDMB	–	–	–	–	–	–	–	O, PD
K2	K2	VDDIN33	USBHS	HHSDPC	–	–	–	–	–	–	–	O, PD
K1	K1	VDDIN33	USBHS	HHSDMC	–	–	–	–	–	–	–	O, PD
L5	T3	VDDBU	GPIO	WKUP0	–	–	–	–	–	–	–	I, ST
J6	H7	VDDBU	GPIO	SHDN	–	–	–	–	–	–	–	O, PD
P1	P1	VDDBU	GPIO	JTAGSEL	–	–	–	–	–	–	–	I, PD
J9	R1	VDDBU	GPIO	TST	–	–	–	–	–	–	–	I, PD, ST
H5	L7	VDDIOP0	GPIO	TCK	–	–	–	–	–	–	–	I, ST
H4	J7	VDDIOP0	GPIO	TDI	–	–	–	–	–	–	–	I, ST
T2	N6	VDDIOP0	GPIO	TDO	–	–	–	–	–	–	–	O
P2	R2	VDDIOP0	GPIO	TMS	–	–	–	–	–	–	–	I, ST
R3	R3	VDDIOP0	GPIO	RTCK	–	–	–	–	–	–	–	O
F5	M5	VDDIOP0	GPIO	NRST	–	–	–	–	–	–	–	I, PU, ST
R2	T1	VDDBU	CLOCK	XIN32	–	–	–	–	–	–	–	I
R1	T2	VDDBU	CLOCK	XOUT32	–	–	–	–	–	–	–	O
G1	G3	VDDIN33	CLOCK	XIN	–	–	–	–	–	–	–	I
G2	G4	VDDIN33	CLOCK	XOUT	–	–	–	–	–	–	–	O
B1	B1	VDDMIPI	Analog	MIPI_DN0	I/O	–	–	–	–	–	–	HiZ[5]
B2	B2	VDDMIPI	Analog	MIPI_DP0	I/O	–	–	–	–	–	–	HiZ[5]
C1	C1	VDDMIPI	Analog	MIPI_DN1	I/O	–	–	–	–	–	–	HiZ[5]
C2	C2	VDDMIPI	Analog	MIPI_DP1	I/O	–	–	–	–	–	–	HiZ[5]
E1	E1	VDDMIPI	Analog	MIPI_DN2	I/O	–	–	–	–	–	–	HiZ[5]
E2	E2	VDDMIPI	Analog	MIPI_DP2	I/O	–	–	–	–	–	–	HiZ[5]
F1	F1	VDDMIPI	Analog	MIPI_DN3	I/O	–	–	–	–	–	–	HiZ[5]
F2	F2	VDDMIPI	Analog	MIPI_DP3	I/O	–	–	–	–	–	–	HiZ[5]
D1	D1	VDDMIPI	Analog	MIPI_CLKN	O	–	–	–	–	–	–	HiZ[5]
D2	D2	VDDMIPI	Analog	MIPI_CLKP	O	–	–	–	–	–	–	HiZ[5]
H6	F4	VDDMIPI	Analog	MIPI_REXT	I	–	–	–	–	–	–	I[5]
J5	L4	VDDANA	GPIO	AUDIOCLK	–	–	–	–	–	–	–	O
–	L10	VDDCORE	Power	–	–	–	–	–	–	–	–	–
–	L11	VDDCORE	Power	–	–	–	–	–	–	–	–	–
–	J4	VDDOUT25	Power	–	–	–	–	–	–	–	–	–

Table 8.1. Pin Description[1] (continued)

240-pin BGA	256-pin BGA	Power Rail	I/O Type[2]	Primary		Alternate		PIO Peripheral				Reset State[4]
				Signal	Dir	Signal	Dir	Func	Signal	Dir	I/O Set[3]	Signal, Dir, PU, PD, HiZ, ST, SEC, FILTER
–	L2	VDDANA	Power	–	–	–	–	–	–	–	–	–
–	M1	VDDANA	Power	–	–	–	–	–	–	–	–	–
–	N14	VDDIOM	Power	–	–	–	–	–	–	–	–	–
–	L3	GND	Ground	–	–	–	–	–	–	–	–	–
–	L9	GND	Ground	–	–	–	–	–	–	–	–	–
–	L14	GND	Ground	–	–	–	–	–	–	–	–	–
–	M13	GND	Ground	–	–	–	–	–	–	–	–	–
–	N8	GND	Ground	–	–	–	–	–	–	–	–	–
–	N11	GND	Ground	–	–	–	–	–	–	–	–	–
–	N13	GND	Ground	–	–	–	–	–	–	–	–	–
–	P6	GND	Ground	–	–	–	–	–	–	–	–	–
–	R9	GND	Ground	–	–	–	–	–	–	–	–	–
–	R14	GND	Ground	–	–	–	–	–	–	–	–	–
–	T16	GND	Ground	–	–	–	–	–	–	–	–	–
–	U1	GND	Ground	–	–	–	–	–	–	–	–	–
–	U17	GND	Ground	–	–	–	–	–	–	–	–	–

Notes:

1. When using an I/O line with the Analog-to-Digital Converter (ADC), the PIO line configuration (pull-up, pull-down) programmed before assigning this line to the ADC peripheral is not modified by this peripheral.
2. Refer to the Electrical Characteristics section for further details.
3. 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.
4. Signal = 'PIO' if GPIO; Dir = Direction; PU = Pull-up; PD = Pull-down; HiZ = High impedance; ST = Schmitt Trigger
5. On SAM9X70 and SAM9X72: to be tied to GND.

9. Microchip Recommended Power Management Solutions

MCP16502 and MCP16501 are multi-channel Power Management Integrated Circuits (PMICs) recommended for the SAM9X7 series.

9.1. MCP16502 PMIC

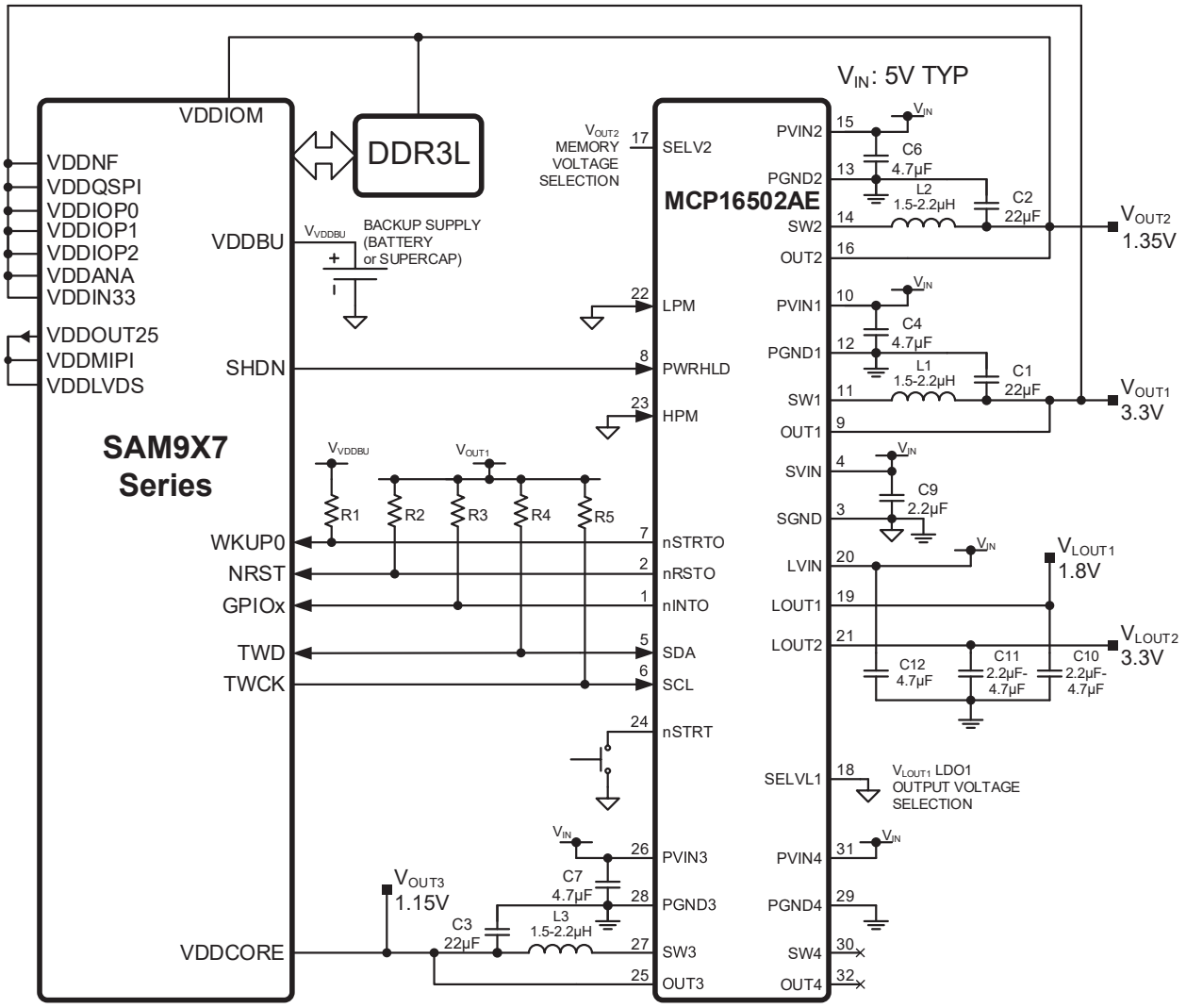
MCP16502 features four 1A DC-DC buck regulators and two 0.3A auxiliary LDO regulators, and provides a comprehensive interface to the MPU, which includes an interrupt flag and a 1-MHz I²C interface.

The PMIC-processor interface is optimized so that it remains leakage-free in Backup mode. The following figure gives an application schematic example of a SAM9X7 device with a DDR3L-SDRAM system, powered by MCP16502**AE**. This variant is specifically tailored for SAM9X7 systems with CPU frequency up to 800 MHz. The 3.3V, 1.35V and 1.15V supply rails are fed from DC-DC converters for maximum efficiency. The fourth DC-DC converter (Buck4) of MCP16502**AE** is left off by default during start-up and its components may be removed, if not needed for other purposes.

The two LDO regulator outputs LOUT1 and LOUT2 are auxiliary power rails available for the application. LOUT1 output is on by default at power-up and its default voltage is set to 1.8V, 2.5V or 3.3V depending on the SELV1 pin connection. Buck4 and LOUT2, off by default at power-up, can be started by software through the I²C control bus to the necessary voltage.

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

Figure 9.1. MCP16502 Simplified Application Block Diagram



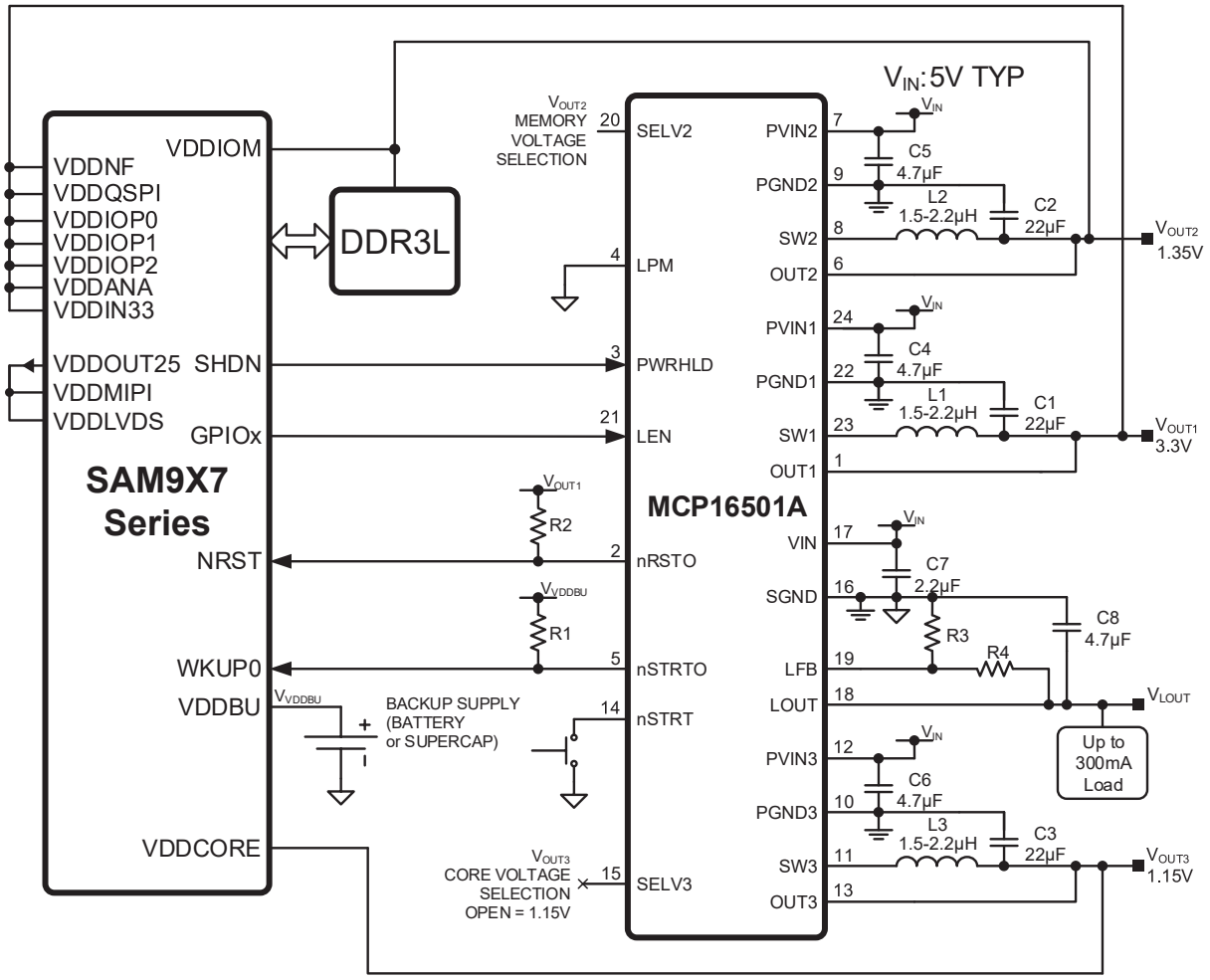
9.2. MCP16501 PMIC

MCP16501 is a 4-channel PMIC designed for PCB area constrained applications. In a 4x4mm QFN24 package, it features three 1A DC-DC buck regulators and one 0.3A auxiliary LDO regulator, and provides a simple, leakage-free interface with SAM9X7.

The following figure gives an application schematic example of a SAM9X7 device with a DDR3L-SDRAM system, powered by MCP16501**A**. This variant is specifically tailored for SAM9X7 systems with CPU frequency up to 800 MHz. The 3.3V, 1.35V and 1.15V supply rails are fed from DC-DC converters for maximum efficiency. The LDO regulator output LOUT is controlled with the LEN input and its output voltage is set with the resistive divider R3/R4.

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

Figure 9.2. MCP16501 Simplified Application Block Diagram



10. Safety and Security Features

10.1. Design for Safety and IEC60730 Class B Certification

10.1.1. 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 SAM9X7 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. Table 10.1 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.

10.2. Design for Security

The SAM9X7 embeds peripherals with security features to prevent counterfeiting, to secure external communication, and to authenticate the system.

Table 10.2 provides the list of peripherals and an overview of their security function. For more information, refer to the sections on each peripheral.

10.3. Safety and IEC 60730 Features

Table 10.1. Safety and IEC 60730 Features

Peripheral	Component	Fault/Error/Feature	Requirements for Class B IEC 60730(1)	General Safety
PMC	Clock	MCK frequency monitor - MCK out-of-range operation	–	X
		32.768 kHz crystal oscillator frequency monitor - Abnormal frequency deviation	X	X
		Main crystal oscillator failure detector - Crystal failure detection	X	X
System Controller	All	Safety critical peripherals and/or counters are fed by the always-on slow RC oscillator - WDT, RSTC, start-up counters, timeout counters, etc.	–	X
PIOC	I/O lines	Digital I/O - Plausibility check	X	–
ADCC		Analog I/O and ADC converter - Plausibility check	X	–
NAND Flash Controller ECC	Memory	Non-volatile memory - Multiple error detection (2 to 24)	–	X

Table 10.1. Safety and IEC 60730 Features (continued)

Peripheral	Component	Fault/Error/Feature	Requirements for Class B IEC 60730(1)	General Safety
WDT, RSTC	Watchdog	Watchdog is driven by an internal always on clock - Program counter stuck at faults	X	X
		Watchdog configuration can be locked until the next reset - Errant writes (programming errors, errors introduced by system or hardware failures)	–	X
		Watchdog overflow generates a system reset	X	X
Arm926EJ-S MMU	Memory Management Unit	Arm926EJ-S Memory Management Unit	–	X
MATRIX, AIC, RTC, RTT, RSTC, SHDWC, SDRAM, PMC, PIOC, MPDDRC, SMC, CLASSD, SSC, FLEXCOM, QSPI, TC, I2SMCC, ADC	Peripherals	Configuration, Interrupt Enable/ Disable, Control registers can be independently write-protected - Errant writes (programming errors, errors introduced by system or hardware failures)	–	X
AES, TDES, SHA, PIT64B, TC, MPDDRC	Peripherals	Embedded integrity checker with reports in status registers	–	X

Note:

1. Class B IEC 60730 Requirements. Annex H - Table H.1 (H.11.12.7 in Edition 3).

10.4. Security Features

Table 10.2. Security Features

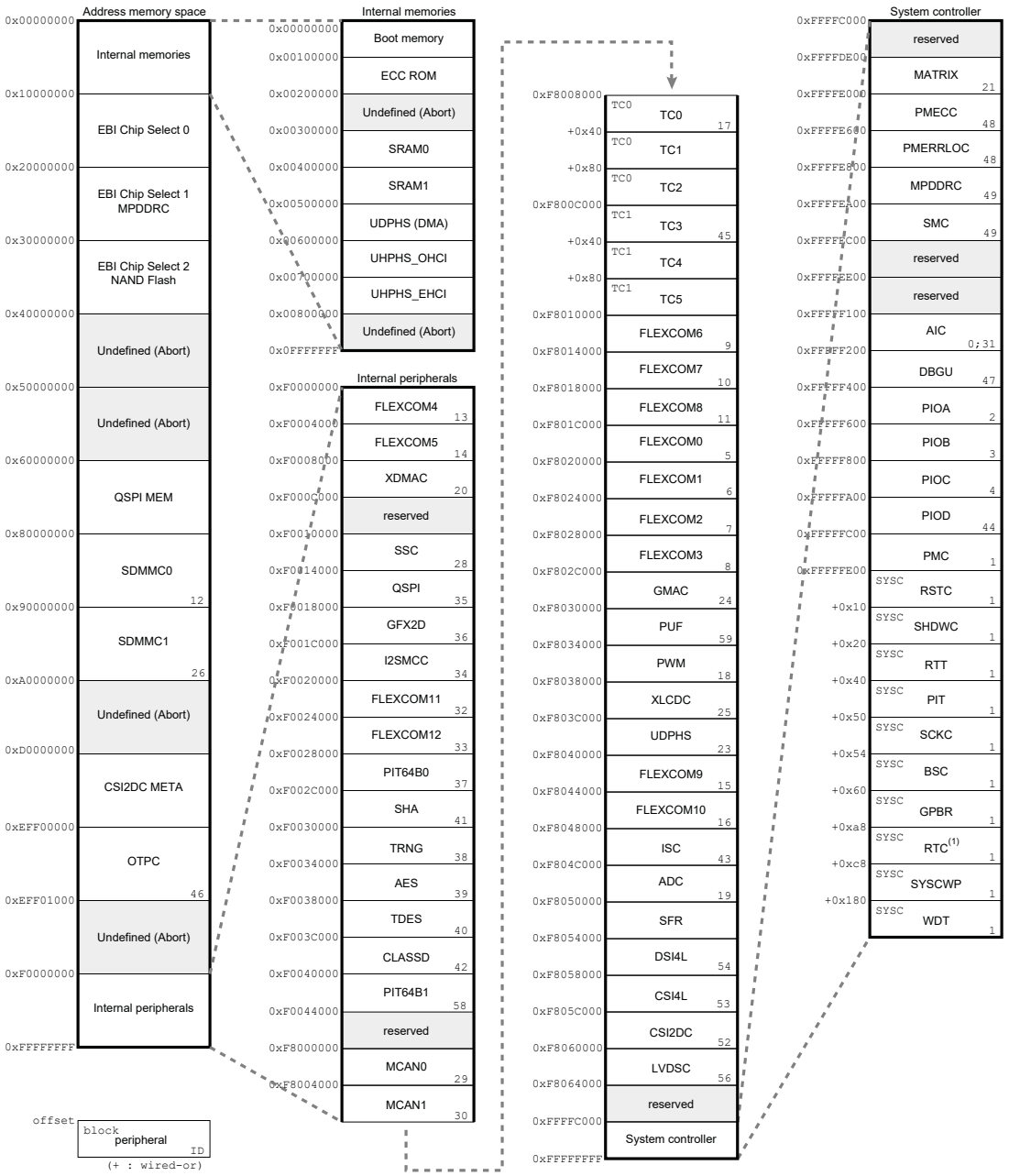
Peripheral	Function	Description	Comments
Arm926EJ-S MMU	Memory Management Unit	Memory Management Unit	–
PIO	I/O Control/ Peripheral Access	When a peripheral is not selected (PIO-controlled), I/O lines have no access to the peripheral.	–
AES	Cryptography Standards	Hardware-accelerated AES up to 256 bits	FIPS-compliant
SHA		SHA up to 512 and HMAC-SHA	
TDES		Hardware-accelerated Triple DES	
TRNG		True Random Number Generator	
AES, TDES	Cryptography Tamper	Immediate clear of keys in case of external tamper event detection (if enabled)	–
AES, TDES, SHA	Cryptography Integrity Checks	AES/TDES/SHA embed integrity checks on configuration registers and algorithm circuitries and a specific flag in status register. If this specific flag is set, an integrity error has been detected. This can occur only on abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)	–

Table 10.2. Security Features (continued)

Peripheral	Function	Description	Comments
OTPC, AES, TDES, TRNG	Cryptography Private Key Bus	Capability to transfer a key to AES/TDES in a totally invisible manner from software	–
Secure Boot	Secure Boot	Code encrypted/decrypted, Trusted Code Authentication	Hardware SHA (HMAC) + Software RSA or AES Hardware (CMAC)
Memories	Scrambling	On-the-fly scrambling/ unscrambling for memories	All external memories such as QSPI, DDR, and all memories on SMC
Physical Unclonable Function	Key Generation	Key creation, derivation, wrapping and management	Includes NIST SP 800-90B compliant DRNG
RTC	IO Tamper Pin	Eight tamper detection pins	VDDCORE WKUP1 to WKUP8 pins can be selected as a source of tamper, performing an immediate clear of AES/TDES keys (if enabled), immediate clear of scrambling keys in DDR/QSPI/ SMC, and immediate clear of General Purpose Backup Registers (if enabled)
	Timestamping	Timestamping of tamper events	All events are logged in the RTC. Timestamping gives the source of the reset/erase memory/interruption
	Configuration	Protection against bad configuration (invalid entry for date and time are impossible)	–
	Glitch Robustness	Glitch on 32 KHz does not corrupt the downstream counters	Glitch on 32 KHz can only create a phase shift of the downstream counters
	Integrity Check	If RTC Status flag TDERR is set, counters integrity have been corrupted	–
Secure OTP	JTAG Access Control	Disable JTAG access by OTP bit	–
PIT64B, TC	Integrity Checks	PIT64B/TC embed integrity checks on configuration registers and algorithm circuitries and a specific flag in status register. If this specific flag is set, an integrity error has been detected. This can occur only on abnormal operating conditions (electromagnetic attacks, VDD glitches, etc.)	–
GPBR	Access Protection	GPBR can be write-protected and/or read-protected	–
	Tamper	GBPR can be immediately cleared on tamper detection (if enabled)	–

11. Memory Mapping

Figure 11.1. Memory Mapping



(1) Refer to the table System Controller Peripheral Mapping in section System Controller Write Protection (SYSCWP) for RTC detailed mapping.

12. Peripheral Identifiers

Table 12.1. Peripheral Identifiers

Instance ID	Instance Name	Internal Interrupt	External Interrupt	Wired-OR	PMC Clock Control	Main System Bus Clock	Generic Clock	f_{GCLK} (Max)	PLLADIV2CLK	UPLLCLK	AUDIOPLL	Instance Description
0	AIC	–	FIQ	–	–	MCK	–	–	–	–	–	Advanced Interrupt Controller
1	SYSC	X	–	SYSC, PMC, WDT, PIT, RSTC, RTT, RTC	–	MCK	–	–	–	–	–	Logical-OR Interrupt of SYSC, PMC, WDT, PIT, RSTC, RTT, RTC
2	PIOA	X	–	–	X	MCK	–	–	–	–	–	Parallel I/O Controller A
3	PIOB	X	PB18	–	X	MCK	–	–	–	–	–	Parallel I/O Controller B
4	PIOC	X	–	–	X	MCK	–	–	–	–	–	Parallel I/O Controller C
5	FLEXCOM0	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 0
6	FLEXCOM1	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 1
7	FLEXCOM2	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 2
8	FLEXCOM3	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 3
9	FLEXCOM6	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 6
10	FLEXCOM7	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 7
11	FLEXCOM8	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 8
12	SDMMC0	X	–	–	X	MCK	X	105	X	–	X	Secure Data Memory Card Controller 0
13	FLEXCOM4	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 4
14	FLEXCOM5	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 5
15	FLEXCOM9	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 9
16	FLEXCOM10	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	–	FLEXCOM 10
17	TC0	X	–	–	X	MCK	X	$f_{MCK}/3$	X	–	X	Timer Counters 0,1,2
18	PWM	X	–	–	X	MCK	–	–	–	–	–	Pulse Width Modulation Controller
19	ADC	X	–	–	X	MCK	X	$f_{MCK}/3$	X	X	–	ADC Controller
20	XDMAC	X	–	–	X	MCK	–	–	–	–	–	Extended DMA Controller
21	MATRIX	X	–	–	–	MCK	–	–	–	–	–	Matrix
22	UHPHS	X	–	–	X	MCK	–	–	–	–	–	USB Host High Speed
23	UDPHS	X	–	–	X	MCK	–	–	–	–	–	USB Device High Speed
24	GMAC	X	–	–	X	MCK	X	50	X	–	X	Gigabit Ethernet MAC
25	LCDC	X	–	–	X	MCK	X	75	X	–	X	LCD Controller
26	SDMMC1	X	–	–	X	MCK	X	105	X	–	X	Secure Data Memory Card Controller 1

Table 12.1. Peripheral Identifiers (continued)

Instance ID	Instance Name	Internal Interrupt	External Interrupt	Wired-OR	PMC Clock Control	Main System Bus Clock	Generic Clock	f_{GCLK} (Max)	PLLADIV2CLK	UPLLCLK	AUDIOPLL	Instance Description
27	Reserved	–	–	–	–	MCK	–	–	–	–	–	–
28	SSC	X	–	–	X	MCK	–	–	–	–	–	Synchronous Serial Controller
29	MCAN0	X	–	–	X	MCK	X	80	X	X	–	CAN Controller 0
30	MCAN1	X	–	–	X	MCK	X	80	X	X	–	CAN Controller 1
31	AIC	–	IRQ	–		MCK	–	–	–	–	–	Advanced Interrupt Controller
32	FLEXCOM11	X	–	–	X	MCK	X	f_{MCK}/3	X	–	–	FLEXCOM 11
33	FLEXCOM12	X	–	–	X	MCK	X	f_{MCK}/3	X	–	–	FLEXCOM 12
34	I2SMCC	X	–	–	X	MCK	X	100	X	–	X	I2S Multi Channel Controller
35	QSPI	X	–	–	X	MCK(1)	X	200	X	–	X	Quad I/O SPI Controller
36	GFX2D	X	–	–	X	MCK	–	–	–	–	–	2D Graphics Controller
37	PIT64B0	X	–	–	X	MCK	X	f_{MCK}/3	X	–	–	64-bit Timer
38	TRNG	X	–	–	X	MCK	–	–	–	–	–	True Random Number Generator
39	AES	X	–	–	X	MCK	–	–	–	–	–	Advanced Encryption Standard
40	TDES	X	–	–	X	MCK	–	–	–	–	–	Triple Data Encryption Standard
41	SHA	X	–	–	X	MCK	–	–	–	–	–	Secure Hash Algorithm
42	CLASSD	X	–	–	X	MCK	X	100	X	–	X	CLASS D Controller
43	ISI	X	–	–	X	MCK	–	–	–	–	–	Image Sensor Interface
44	PIOD	X	–	–	X	MCK	–	–	–	–	–	Parallel I/O Controller D
45	TC1	X	–	–	X	MCK	X	f_{MCK}/3	X	–	X	Timer Counter 3, 4, 5
46	OTPC	X	–	–	–	MCK	–	–	–	–	–	OTP Controller
47	DBGU	X	–	–	X	MCK	X	f_{MCK}/3	X	–	–	Debug Unit
48	PMECC	X	–	PMECC, PMERRLOC	–	MCK	–	–	–	–	–	Logical-OR Interrupt of PMECC and PMERRLOC
49	MPDDRC	X	–	MPDDRC, SMC	X	MCK	–	–	–	–	–	Logical-OR Interrupt of MPDDRC and HSMC
50	UTMI	–	–	–	–	MCK	–	–	–	–	–	UTMI Controller
51	Reserved	–	–	–	–	–	–	–	–	–	–	–

Table 12.1. Peripheral Identifiers (continued)

Instance ID	Instance Name	Internal Interrupt	External Interrupt	Wired-OR	PMC Clock Control	Main System Bus Clock	Generic Clock	f_{GCLK} (Max)	PLLADIV2CLK	UPLLCLK	AUDIOPLL	Instance Description
52	CSI2DC	X	–	–	X	MCK	–	–	–	–	–	CSI to Demultiplexer Controller
53	CSI	X	–	–	X	MCK	–	–	–	–	–	Camera Serial Interface between ISC and MIPI D-PHY
54	DSI	X	–	–	X	MCK	–	–	–	–	–	Display Serial Interface between LCDC and MIPI D-PHY
55	MIPIPHY	–	–	–	–	–	X	27	X	–	–	MIPI D-PHY interface
56	LVDSC	–	–	–	X	X	–	–	–	–	–	Display Serial Interface between LCDC and LVDS Interface
57	LVDSPHY	–	–	–	–	–	–	–	–	–	–	LVDS Physical Interface
58	PIT64B1	X	–	–	X	MCK	X	$f_{MCK/3}$	X	–	X	64-bit Timer 1
59	PUF	–	–	–	X	MCK	–	–	–	–	–	PUF controller
60	GMAC	Q1	–	–	–	–	–	–	–	–	–	GMAC Queue 1 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 1
61	GMAC	Q2	–	–	–	–	–	–	–	–	–	GMAC Queue 2 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 2
62	GMAC	Q3	–	–	–	–	–	–	–	–	–	GMAC Queue 3 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 3

Table 12.1. Peripheral Identifiers (continued)

Instance ID	Instance Name	Internal Interrupt	External Interrupt	Wired-OR	PMC Clock Control	Main System Bus Clock	Generic Clock	f_{GCLK} (Max)	PLLADIV2CLK	UPLLCLK	AUDIOPLL	Instance Description
63	GMAC	Q4	–	–	–	–	–	–	–	–	–	GMAC Queue 4 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 4
64	GMAC	Q5	–	–	–	–	–	–	–	–	–	GMAC Queue 5 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 5
65	GMAC	EMAC	–	–	–	–	–	–	–	–	–	Express MAC
66	GMAC	MMSL	–	–	–	–	–	–	–	–	–	MAC Merge Sublayer
67	Reserved	–	–	–	–	–	–	–	–	–	–	–
68	MCAN0	INT1	–	–	–	–	–	–	–	–	–	MCAN0 Interrupt 1
69	MCAN1	INT1	–	–	–	–	–	–	–	–	–	MCAN1 Interrupt 1

Note:

1. This QSPI GCLK is a 2x clock. It must be set to 200 MHz to reach 100 MHz on the data.

CPU and Interconnect

13. Arm926EJ-S Processor

The Arm926EJ-S processor is a member of the Arm9™ family of general-purpose microprocessors. The Arm926EJ-S implements Arm architecture version 5TEJ and is targeted at multi-tasking applications where full memory management, high performance, low die size and low power are important features.

The Arm926EJ-S processor supports the 32-bit Arm and 16-bit Thumb instruction sets, enabling the user to trade off between high performance and high code density. It also supports the 8-bit Java instruction set and includes features for efficient execution of Java bytecode, providing a Java performance similar to JITs (Just-In-Time compilers), for the next generation of Java-powered wireless and embedded devices. It includes an enhanced multiplier design for improved DSP performance.

The Arm926EJ-S processor supports the Arm debug architecture and includes logic to assist in both hardware and software debug.

The Arm926EJ-S provides a complete high performance processor subsystem, including:

- an Arm9EJ-S™ integer core,
- a Memory Management Unit (MMU),
- separate instruction and data AMBA AHB bus interfaces,
- a 32-Kbyte L1 instruction cache and a 32-Kbyte data cache.

For information on the Arm926EJ-S processor, refer to the ARM926EJ-S Technical Reference Manual on www.arm.com.

14. Debug and Test

14.1. 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. The Debug Unit provides a two-pin UART that can be used to upload an application into the internal SRAM. It manages the interrupt handling of the internal COMMTX and COMMRX signals that trace the activity of the Debug Communication Channel.

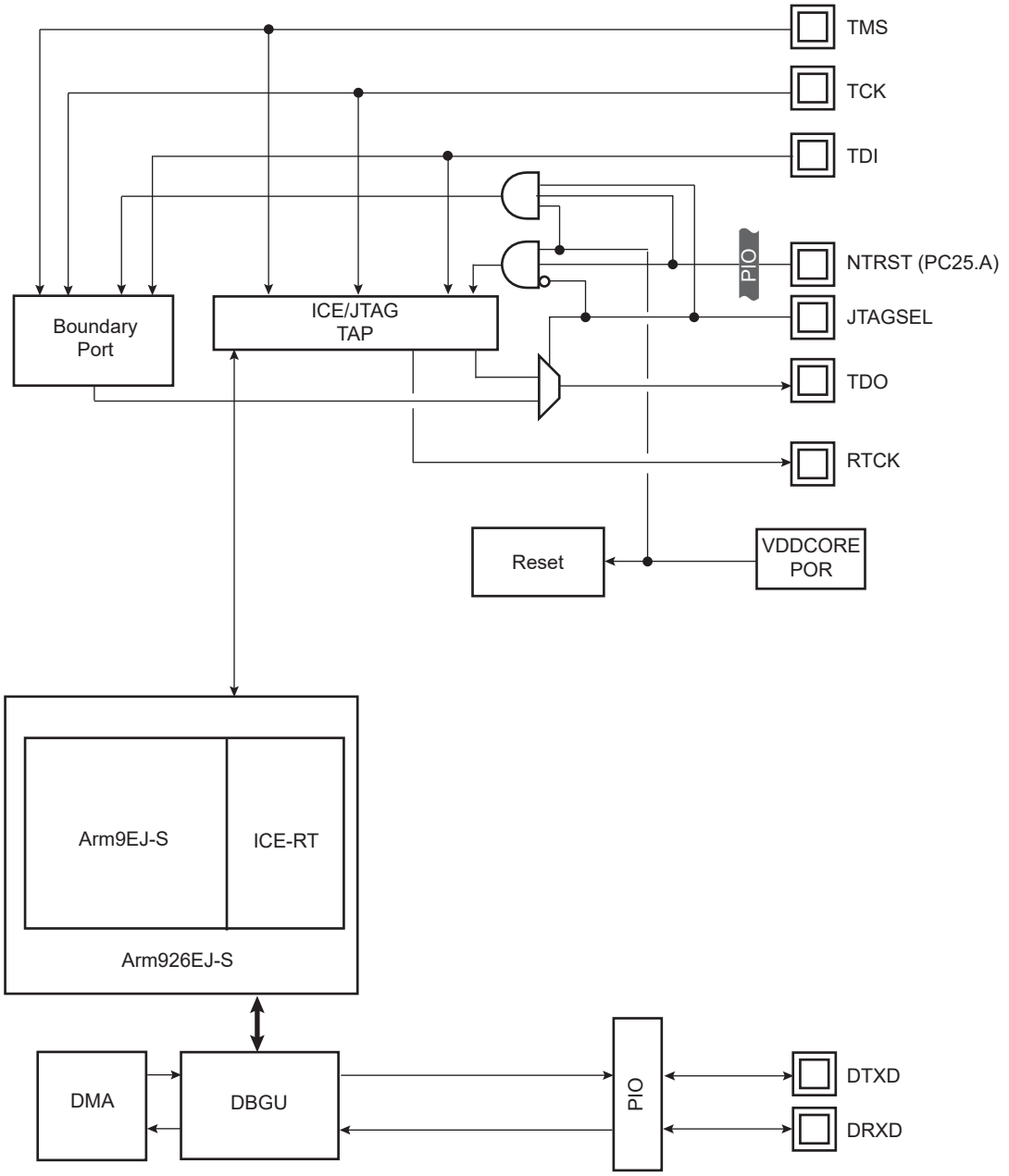
A set of dedicated debug and test input/output pins gives direct access to these capabilities from a PC-based test environment.

14.2. Embedded Characteristics

- Arm926 Real-Time In-Circuit Emulator
  - Two real-time watchpoint units
  - Two independent registers: Debug Control register and Debug Status register
  - Test access port accessible through JTAG protocol
  - Debug communications channel
- Debug Unit
  - Two-pin UART
  - Debug Communication Channel interrupt handling
  - Chip ID register
- IEEE® 1149.1 JTAG Boundary Scan on All Digital Pins

14.3. Block Diagram

Figure 14.1. Debug and Test Block Diagram



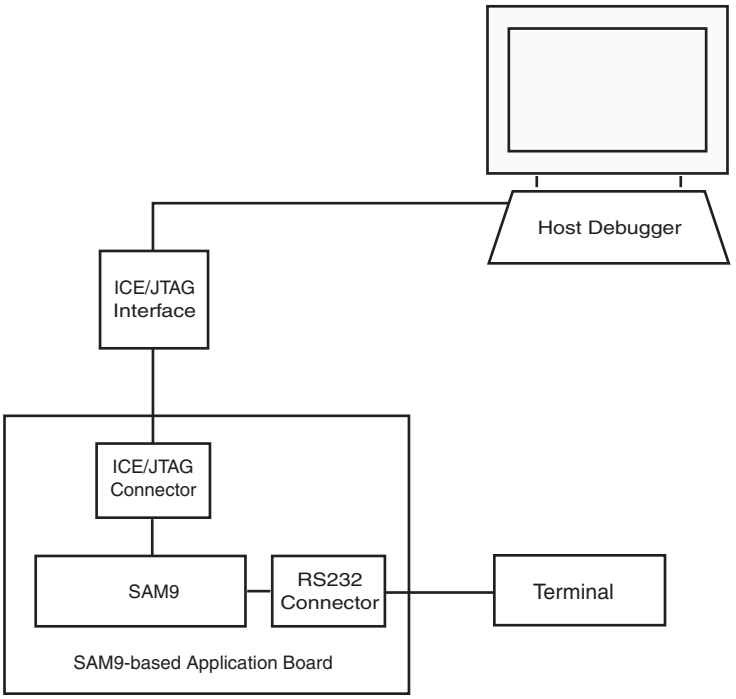
TAP: Test Access Port

14.4. Application Examples

14.4.1. Debug Environment

The following figure shows a complete debug environment example. The ICE/JTAG interface is used for standard debugging functions, such as downloading code and single-stepping through the program. A software debugger running on a personal computer provides the user interface for configuring a Trace Port interface utilizing the ICE/JTAG interface.

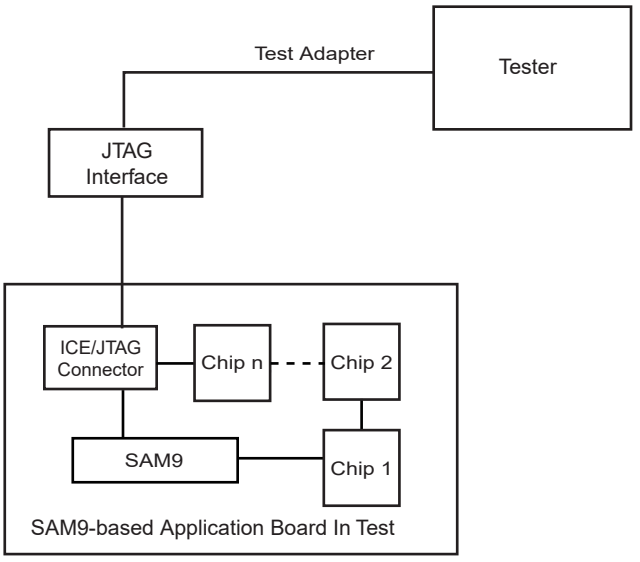
Figure 14.2. Application Debug and Trace Environment Example



14.4.2. Test Environment

The following figure shows a test environment example. Test vectors are sent and interpreted by the tester. In this example, the “board in test” is designed using a number of JTAG-compliant devices. These devices can be connected to form a single scan chain.

Figure 14.3. Application Test Environment Example



14.5. Debug and Test Pin Description

Table 14.1. Debug and Test Pin List

Pin Name	Function	Type	Active Level
Reset/Test			

Table 14.1. Debug and Test Pin List (continued)

Pin Name	Function	Type	Active Level
NRST	Microcontroller Reset	Input	Low
ICE and JTAG			
NTRST	Test Reset Signal	Input	Low
TCK	Test Clock	Input	–
TDI	Test Data In	Input	–
TDO	Test Data Out	Output	–
TMS	Test Mode Select	Input	–
RTCK	Returned Test Clock	Output	–
JTAGSEL	JTAG Selection	Input	–
Debug Unit			
DRXD	Debug Receive Data	Input	–
DTXD	Debug Transmit Data	Output	–

14.6. Functional Description

14.6.1. EmbeddedICE™

The Arm9EJ-S EmbeddedICE-RT is supported via the ICE/JTAG port. It is connected to a host computer via an ICE interface. Debug support is implemented using an Arm9EJ-S core embedded within the Arm926EJ-S. The internal state of the Arm926EJ-S 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 Arm9EJ-S registers. This data can be serially shifted out without affecting the rest of the system.

There are two scan chains inside the Arm9EJ-S 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-RT, refer to the ARM9EJ-S Technical Reference Manual (DDI 0222) on www.arm.com.

14.6.2. 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 mandatory in Arm cores and used to reset the debug logic. On Microchip Arm926EJ-S-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. Note the maximum JTAG clock rate on Arm926EJ-S cores is 1/6th the clock of the CPU. This gives 5.45 kHz maximum initial JTAG clock rate for an Arm9E running from the 32.768 kHz slow clock.

RTCK is the Return Test Clock. This return signal (non-IEEE Standard 1149.1) improves clock handling by emulators. From some ICE interface probes, it can be used to synchronize the TCK clock and it does not care about the given ratio between the ICE interface clock and system clock equal to 1/6th. The RTCK signal is available in JTAG ICE mode only (not in Boundary Scan mode).

14.6.3. Debug Unit

The Debug Unit manages the interrupt handling of the COMMTX and COMMRX signals coming from the ICE and tracing the activity of the Debug Communication Channel. The Debug Unit is used to block access to the system through the ICE interface.

For further details on the Debug Unit, refer to Debug Unit (DBGU).

14.6.4. 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 and NTRST are 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 test.

14.6.5. JTAG ID Code Register

Access: Read-only

Bit	31	30	29	28	27	26	25	24
	VERSION				PART NUMBER			
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PART NUMBER							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PART NUMBER				MANUFACTURER IDENTITY			
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	MANUFACTURER IDENTITY							1

- **VERSION[31:28]: Product Version Number**

Set to 0x0.

- **PART NUMBER[27:12]: Product Part Number**

Product part number is 0x5B46

- **MANUFACTURER IDENTITY[11:1]**

Set to 0x01F.

Bit[0] required by IEEE Std. 1149.1.

Set to 0x1.

JTAG ID Code value is 0x05B4_603F.

15. Bus Matrix (MATRIX)

15.1. Description

The Bus Matrix (MATRIX) implements a multi-layer AHB, based on the AHB-Lite protocol, that enables parallel access paths between multiple hosts and clients in a system, thus increasing the overall bandwidth.

The MATRIX interconnects 14 hosts to 12 clients. The normal latency to connect a host to a client is one cycle, except for the default host of the accessed client which is connected directly (zero cycle latency).

The MATRIX user interface is compliant with the Arm Advanced Peripheral Bus.

15.1.1. MATRIX Hosts

The MATRIX manages the 14 hosts listed in the table below. Each host can perform an access, concurrently with others, to an available client. The MATRIX operates at the main system bus clock (MCK) frequency. Each host has its own decoder, which is defined specifically for each host. In order to simplify the addressing, all the hosts have the same decodings.

Table 15.1. List of MATRIX Hosts

Host No.	Description
0	ISC DMA with QoS support
1	LCDC DMA with QoS support
2	GMAC DMA
3, 4	XDMA controller with QoS support
5	GFX2D DMA
6	SDMMC0 DMA
7	SDMMC1 DMA
8	USB high-speed device port (UDPHS) DMA
9	USB high-speed host port (UHPHS) EHCI DMA
10	USB high-speed host port (UHPHS) OHCI DMA
11	Reserved
12	Arm926 instruction
13	Arm926 data

15.1.2. MATRIX Clients

The MATRIX manages the 12 clients listed in the table below. Each client has its own arbiter, providing a dedicated arbitration per client.

Table 15.2. List of MATRIX Clients

Client No.	Description
0	MPDDRC port 4 with QoS support
1	EBI/MPDDRC port 0
2	MPDDRC port 1
3	MPDDRC port 2
4	MPDDRC port 3
5	SRAM0
6	OTPC client interface (ROM and OTP memory)
7	CSI2DC
8	APB 0

Table 15.2. List of MATRIX Clients (continued)

Client No.	Description
9	APB 1
10	QSPI
11	UDPHS dual port RAM
	UHPHS OHCI configuration registers
	UHPHS EHCI configuration registers
	SDMMC0 configuration registers
	SDMMC1 configuration registers
	SRAM1 port 1

Notes:

- APB0 is running @ MCK (up to 266 MHz) and includes TC0-1, FLEXCOM1,2,3,6,7,8,9,10, PWM, ADC, CAN0-1, PUF
- APB1 is running @ MCK (up to 266 MHz) and includes high-performance peripherals: SSC, CLASSD, I2SMCC, TRNG, TDES, SHA, AES PIT64B0-1, FLEXCOM4,5,11,12

15.1.3. Host to Client Access

MATRIX interconnections are versatile and can be modified depending on MPDDRC multiport configuration and NAND Flash usage.

15.1.3.1. MATRIX Configuration for MPDDRC Single Port Setting (Default)

The default configuration is single port, with all devices accessed on D0-D15: SDRAM and/or NAND Flash and/or SMC. This configuration is the Reset state SFR_CCFG_EBICSA.DDR_MP_EN = 0.

As the DDRC multiport feature is disabled, all hosts access DDR-SDRAM using MATRIX Client 1 (MPDDRC port 0), sharing it with EBI accesses.

The NAND Flash can be located either on D0-D7 or on D16-D23 depending on SFR_CCFG_EBICSA.NFD0_ON_D16, respectively 0 (default) or 1.

Table 15.3. MATRIX Interconnections with MPDDRC Single Port Configuration (Default)

Hosts		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Clients		ISC DMA	XLCDC DMA	GMAC DMA	XDMAC IF0	XDMAC IF1	GFX2D	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	Reserved	Arm926 Instruc.	Arm926 Data
0	MPDDRC port 4	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1	EBI (NAND Flash, SMC)	–	–	–	✓	✓	–	–	–	–	–	–	–	✓	✓
	MPDDRC port 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓
2	MPDDRC port 1	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3	MPDDRC port 2	–	–	–	–	–	–	–	–	–	–	–	–	–	–
4	MPDDRC port 3	–	–	–	–	–	–	–	–	–	–	–	–	–	–
5	SRAM0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓
6	OTPC client I/F	–	–	–	✓	✓	–	–	–	✓	–	–	–	✓	✓
7	CSI2DC	–	–	–	✓	–	–	–	–	–	–	–	–	–	✓
8	APB0	–	–	–	✓	✓	–	–	–	–	–	–	–	–	✓
9	APB1	–	–	–	✓	✓	–	–	–	–	–	–	–	–	✓
10	QSPI	–	✓	–	✓	✓	–	–	–	–	–	–	–	✓	✓

Table 15.3. MATRIX Interconnections with MPDDRC Single Port Configuration (Default) (continued)

Hosts		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Clients		ISC DMA	XLCDC DMA	GMAC DMA	XDMAC IF0	XDMAC IF1	GFX2D	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	Reserved	Arm926 Instruc.	Arm926 Data
11	UDPHS DPRAM UHPHS EHCI config. register UHPHS OHCI config. reg. SDMMC0 config. reg. SDMMC1 config. reg. SRAM1 Port 1	–	–	–	–	–	–	–	–	–	–	–	–	✓	✓

15.1.3.2.MATRIX Configuration for MPDDRC Multiport Setting

If the NAND Flash is located on D16-D23, the MPDDRC multiport feature can be enabled. The NAND Flash has its own dedicated port (Client 1) and other peripherals are distributed across four other MATRIX clients (0, 2, 3 and 4), respectively on DDR ports 4, 1, 2 and 3.

This configuration is achieved by setting the bits SFR_CCFG_EBICSA.NFD0_ON_D16 and SFR_CCFG_EBICSA.DDR_MP_EN.

By default, the round-robin arbitration scheme is enabled in MPDDRC. If the system is latency-sensitive, the quality of service (QoS) policy can be enabled in MPDDRC by setting MPDDRC_CONF_ARBITER.ARB to 3.

Table 15.4. MPDDRC Port MATRIX Interconnections with MPDDRC Multiport Configuration

Hosts		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Clients		ISC DMA	XLCDC DMA	GMAC DMA	XDMAC IF0	XDMAC IF1	GFX2D	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	Reserved	Arm926 Instruc.	Arm926 Data
0	MPDDRC port 4	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–
1	NAND Flash	–	–	–	✓	✓	–	–	–	–	–	–	–	✓	✓
	MPDDRC port 0	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2	MPDDRC port 1	–	–	–	✓	–	–	✓	–	–	–	–	–	✓	–
3	MPDDRC port 2	–	–	–	–	✓	✓	–	✓	–	–	–	–	–	✓
4	MPDDRC port 3	–	–	–	–	–	–	–	–	✓	✓	✓	–	–	–
5-11	No change														

15.1.3.3.MATRIX Alternate Configurations

If the NAND Flash is located on D16-D23 (SFR_CCFG_EBICSA.NFD0_ON_D16 = 1) and more flexibility is required, an alternate configuration can be achieved with one bit per host to control port distribution.

Configure MPDDRC in single port (SFR_CCFG_EBICSA.DDR_MP_EN = 0) and remap individual ports via the Remap Multiport DDR register (SFR_REMAP_MP_DDR).

Note: SFR_REMAP_MP_DDR = 14'b11111111111111 is equivalent to MPDDRC multi-port setting with SFR_CCFG_EBICSA.DDR_MP_EN = 1.

Configuration Example

When SFR_REMAP_MP_DDR = 14'b11111111011111, all hosts have a multiport MATRIX client configuration except GFX2D (Host_5), which keeps the single port MATRIX client configuration (MATRIX Client 1).

Performance is increased, since GFX2D uses MATRIX Client 1 (used only for NAND Flash in this configuration), avoiding an arbitration stage with Arm926 data on MATRIX Client 3.

Table 15.5. MPDDRC Port MATRIX Interconnections Example with Dedicated Port Remapping

Hosts		0	1	2	3	4	5	6	7	8	9	10	11	12	13
Clients		ISC DMA	XLCDC DMA	GMAC DMA	XDMAC0	XDMAC1	GFX2D	SDMMC0 DMA	SDMMC1 DMA	UDPHS DMA	UHPHS EHCI	UHPHS OHCI	Reserved	Arm926 Instruc.	Arm926 Data
0	MPDDRC port 4	✓	✓	✓	–	–	–	–	–	–	–	–	–	–	–
1	NAND Flash	–	–	–	✓	✓	–	–	–	–	–	–	–	✓	✓
	MPDDRC port 0	–	–	–	–	–	✓	–	–	–	–	–	–	–	–
2	MPDDRC port 1	–	–	–	✓	–	–	✓	–	–	–	–	–	✓	–
3	MPDDRC port 2	–	–	–	–	✓	–	–	✓	–	–	–	–	–	✓
4	MPDDRC port 3	–	–	–	–	–	–	–	–	✓	✓	✓	–	–	–
5-11	No change														

15.2. Embedded Characteristics

- 14 Configurable Host Ports
- 12 Configurable Client Ports
- One Decoder for Each Host
- One Remap Function for Each Host
- Support for Long Bursts of Length 32, 64, 128 and Up to the Limit of 256-bit Burst Beats of Words
- 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
- Register Write Protection of User Interface Registers

15.3. 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. See MATRIX_SCFGx.

15.4. No Default Host

After the end of the current access, if no other request is pending, the client is disconnected from all hosts. This saves power by preventing useless bus activity, in particular when the bus target clock is disabled by software.

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.

15.5. 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.

15.6. 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.

15.7. 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 Arbitration Scheduling.

15.7.1. 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 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 Slot Cycle Limit Arbitration.

15.7.1.1. 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.

15.7.1.2. 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.

The default reset value of MATRIX_SCFGx.SLOT_CYCLE does not need to be changed.

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

15.7.2. 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 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 MATRIX_PRASx and MATRIX_PRBSx.

Table 15.6. 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.

15.7.2.1. 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.

15.7.2.2. 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.

15.8. 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

MATRIX_MCFGx
MATRIX Host Configuration Register x

MATRIX_SCFGx
MATRIX Client Configuration Register x

MATRIX_PRASx
MATRIX Priority Register A For Clients x

MATRIX_PRBSx
MATRIX Priority Register B For Clients x

MATRIX_MRCR
MATRIX Host Remap Control Register

MATRIX_MEIER
MATRIX Host Error Interrupt Enable Register

MATRIX_MEIDR
MATRIX Host Error Interrupt Disable Register

15.9. 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]		
...										
0x34	MATRIX_MCFG13	31:24								
		23:16								
		15:8								
		7:0						ULBT[2:0]		
0x38 ... 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]							
...										
0x6C	MATRIX_SCFG11	31:24								
		23:16			FIXED_DEFMSTR[3:0]				DEFMSTR_TYPE[1:0]	
		15:8								SLOT_CYCLE[8]
		7:0	SLOT_CYCLE[7:0]							
0x70 ... 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								
		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								
		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								
		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								
		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]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	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								
		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]	
0xAC	MATRIX_PRBS5	31:24								
		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								
		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								
		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								
		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								
		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	MATRIX_PRAS10	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]	
0xD4	MATRIX_PRBS10	31:24								
		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]	
0xD8	MATRIX_PRAS11	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]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xDC	MATRIX_PRBS11	31:24								
		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]	
0xE0 ... 0xFF	Reserved									
0x0100	MATRIX_MRCR	31:24								
		23:16								
		15:8			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			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			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			MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
		7:0	MERR7	MERR6	MERR5	MERR4	MERR3	MERR2	MERR1	MERR0
0x015C	MATRIX_MESR	31:24								
		23:16								
		15:8			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]							
...										
0x0194	MATRIX_MEAR13	31:24	ERRADD[31:24]							
		23:16	ERRADD[23:16]							
		15:8	ERRADD[15:8]							
		7:0	ERRADD[7:0]							
0x0198 ... 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

15.9.1. MATRIX Host Configuration Register x

Name: MATRIX_MCFGx
Offset: 0x00 + x*0x04 [x=0..13]
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.

15.9.2. MATRIX Client Configuration Register x

Name: MATRIX_SCFGx
Offset: 0x40 + x*0x04 [x=0..11]
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 Slot Cycle Limit Arbitration for details.

15.9.3. MATRIX Priority Register A For Clients x

Name: MATRIX_PRASx
Offset: 0x80 + x*0x08 [x=0..11]
Reset: –
Property: Read/Write

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

Table 15.7. MATRIX_PRASx Register Reset Values

Registers	Reset Values
PRAS0	0x00000777
PRAS1, PRAS5	0x00077777
PRAS2	0x00007700
PRAS3	0x00070000
PRAS4	0x00000077
PRAS6, PRAS8, PRAS9	0x00077000
PRAS7	0x00007000
PRAS10	0x00077070
PRAS11	0x00000000

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		–	–	–		–	–	–

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		–	–	–		–	–	–

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		–	–	–		–	–	–

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		–	–	–		–	–	–

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 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.

15.9.4. MATRIX Priority Register B For Clients x

Name: MATRIX_PRBSx
Offset: 0x84 + x*0x08 [x=0..11]
Reset: –
Property: Read/Write

Table 15.8. MATRIX_PRBSx Register Reset Values

Registers	Reset Values
PRBS0, PRBS4, PRBS7	0x00000000
PRBS1, PRBS5, PRBS6, PRBS10, PRBS11	0x00110000
PRBS2	0x00010000
PRBS3, PRBS8, PRBS9	0x00100000

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
		LQOSEN13	M13PR[1:0]			LQOSEN12	M12PR[1:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		–	–	–		–	–	–

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		–	–	–		–	–	–

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		–	–	–		–	–	–

Bits 2, 6, 10, 14, 18, 22 – 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 – 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 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.

15.9.5. 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
			RCB13	RCB12	RCB11	RCB10	RCB9	RCB8
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
	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 – 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.

15.9.6. 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
			MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access			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 – MERRx Host x Access Error

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

15.9.7. 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
			MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access			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 – MERRx Host x Access Error

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

15.9.8. 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
			MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access			R	R	R	R	R	R
Reset			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 – 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.

15.9.9. 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
			MERR13	MERR12	MERR11	MERR10	MERR9	MERR8
Access			R	R	R	R	R	R
Reset			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 – 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.

15.9.10. MATRIX Host Error Address Register x

Name: MATRIX_MEARx
Offset: 0x0160 + x*0x04 [x=0..13]
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

15.9.11. 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 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).

15.9.12. 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.

16. DMA Controller (XDMAC)

16.1. Description

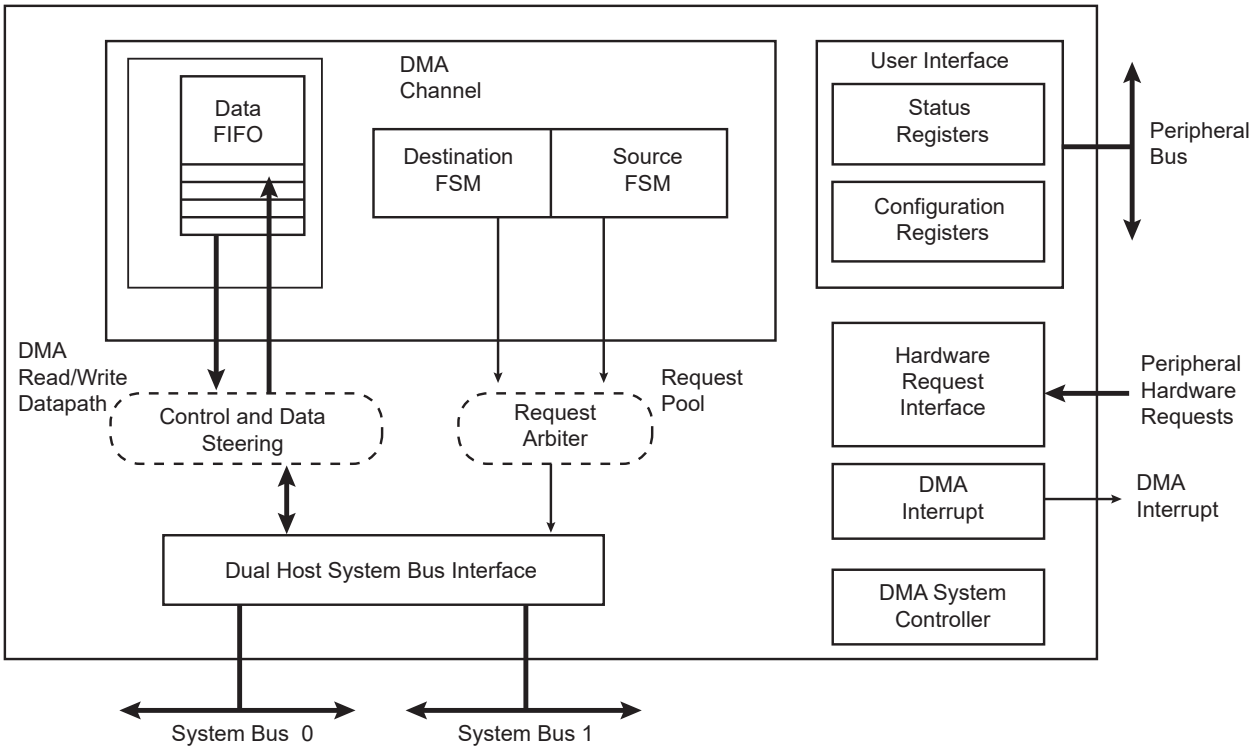
The DMA Controller (XDMAC) is a central direct memory access controller. It performs peripheral data transfer and memory move operations over one or two bus ports through the unidirectional communication channel. Each channel is fully programmable and provides both peripheral or memory-to-memory transfers.

16.2. Embedded Characteristics

- 2 System Bus Host Interfaces
- 16 DMA Channels
- 51 Hardware Requests
- 4 Kbytes Embedded FIFO
- 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) and Word (32 -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 AHB 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

16.3. Block Diagram

Figure 16.1. XDMAC Block Diagram



16.4. DMA Controller Peripheral Connections

DMA Controller 0 manages transfers between peripherals and memory, and receives the triggers from the peripherals listed in the following table.

Table 16.1. DMA Channels Definitions (XDMAC0)

Instance Name	Channel T/R	Interface Number
FLEXCOM0	Transmit	0
FLEXCOM0	Receive	1
FLEXCOM1	Transmit	2
FLEXCOM1	Receive	3
FLEXCOM2	Transmit	4
FLEXCOM2	Receive	5
FLEXCOM3	Transmit	6
FLEXCOM3	Receive	7
FLEXCOM4	Transmit	8
FLEXCOM4	Receive	9
FLEXCOM5	Transmit	10
FLEXCOM5	Receive	11
FLEXCOM6	Transmit	12
FLEXCOM6	Receive	13
FLEXCOM7	Transmit	14
FLEXCOM7	Receive	15
FLEXCOM8	Transmit	16

Table 16.1. DMA Channels Definitions (XDMAC0) (continued)

Instance Name	Channel T/R	Interface Number
FLEXCOM8	Receive	17
FLEXCOM9	Transmit	18
FLEXCOM9	Receive	19
FLEXCOM10	Transmit	20
FLEXCOM10	Receive	21
FLEXCOM11	Transmit	22
FLEXCOM11	Receive	23
FLEXCOM12	Transmit	24
FLEXCOM12	Receive	25
QSPI	Transmit	26
QSPI	Receive	27
DBGU	Transmit	28
DBGU	Receive	29
TDES	Receive	30
TDES	Transmit	31
AES	Transmit	32
AES	Receive	33
SHA	Transmit	34
CLASSD	Transmit	35
I2SMCC	Transmit	36
I2SMCC	Receive	37
SSC	Transmit	38
SSC	Receive	39
ADC	Receive	40
TC0	Receive	41
TC1	Receive	42
TC1_CPA	Compare Counter A, Timer Channel 1	43
TC4_CPA	Compare Counter A, Timer Channel 4	44
TC1_CPB	Compare Counter B, Timer Channel 1	45
TC4_CPB	Compare Counter B Timer Channel 4	46
TC1_CPC	Compare Counter C, Timer Channel 1	47
TC4_CPC	Compare Counter C, Timer Channel 4	48
TC1_ETRG	External Event trigger, timer channel 1 for TC1_ETRG	49
TC4_ETRG	External Event trigger, timer channel 1 for TC4_ETRG	50

16.5. Functional Description

16.5.1. 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.

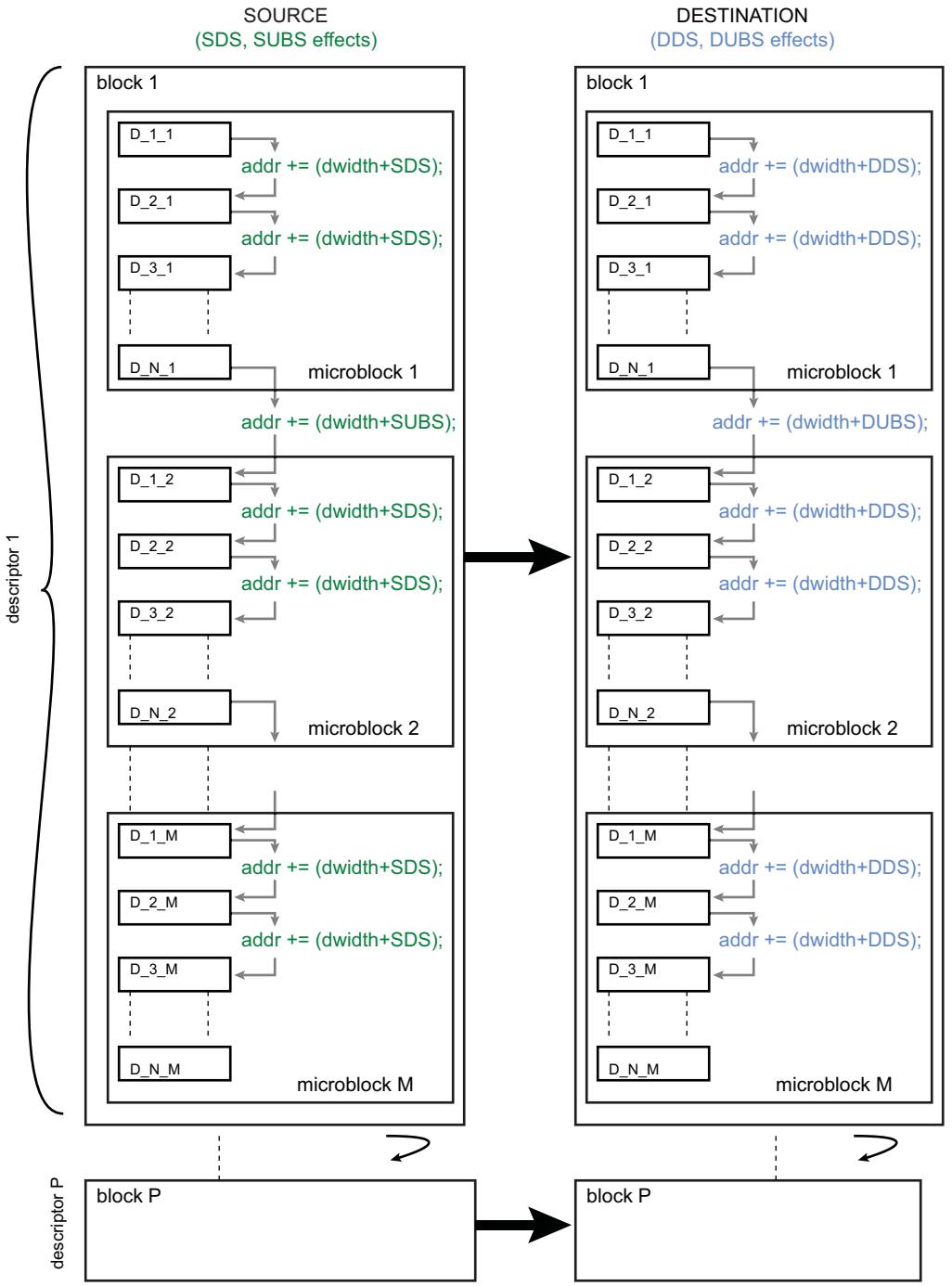
XDMAC Microblock: The microblock is composed of a programmable number of data. The channel configuration registers remain unchanged at the data boundary. The data address may be fixed (a FIFO location, a peripheral transmit or receive register), incrementing (a memory-mapped area) by a programmable signed 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.

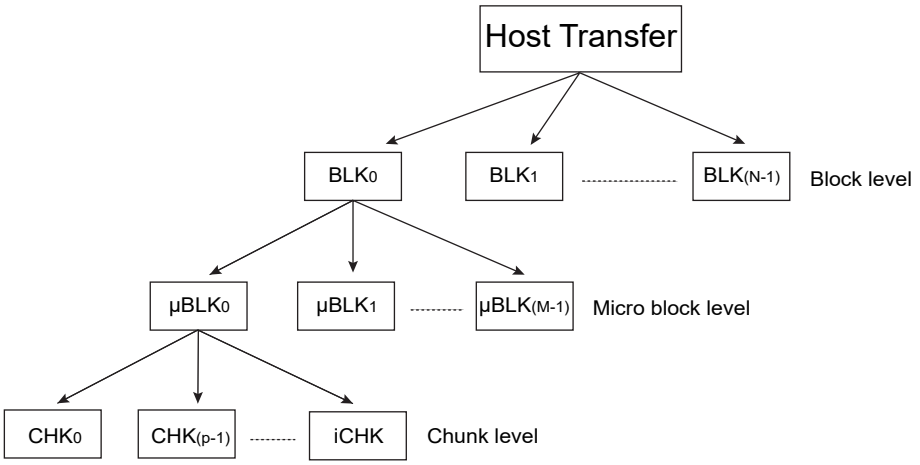
16.5.2. Data Striding Diagram

Figure 16.2. Data Striding Diagram



16.5.3. Transfer Hierarchy Diagrams

Figure 16.3. XDMAC Memory Transfer Hierarchy



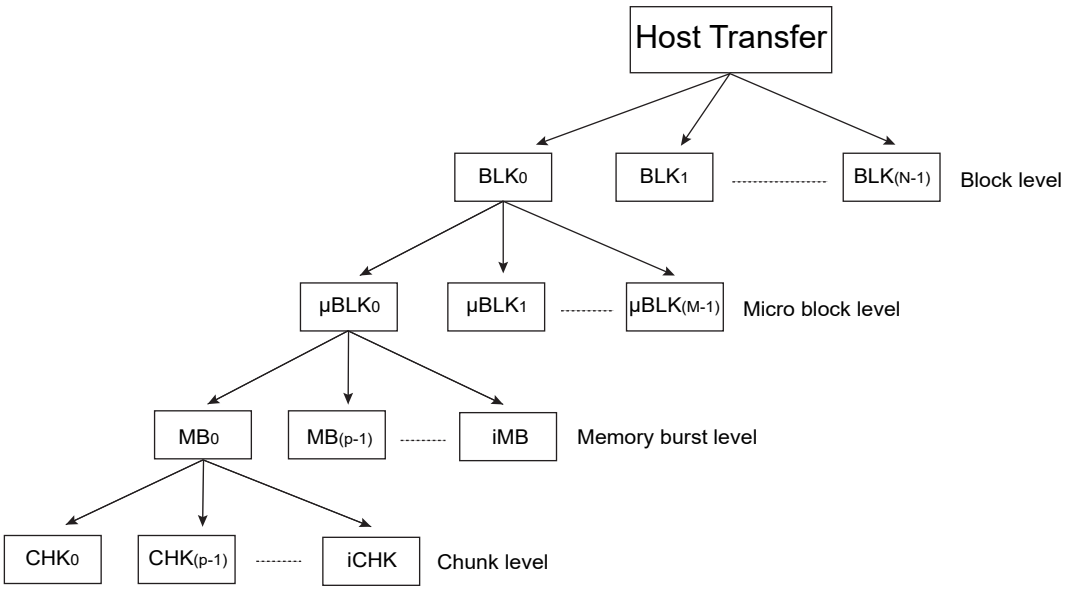
16.5.4. Peripheral Synchronized Transfer

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.

16.5.4.1. Peripheral to Memory Transfer

XDMAC reads data from the source peripheral and writes to the destination memory location.

Figure 16.4. 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 a total of 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.

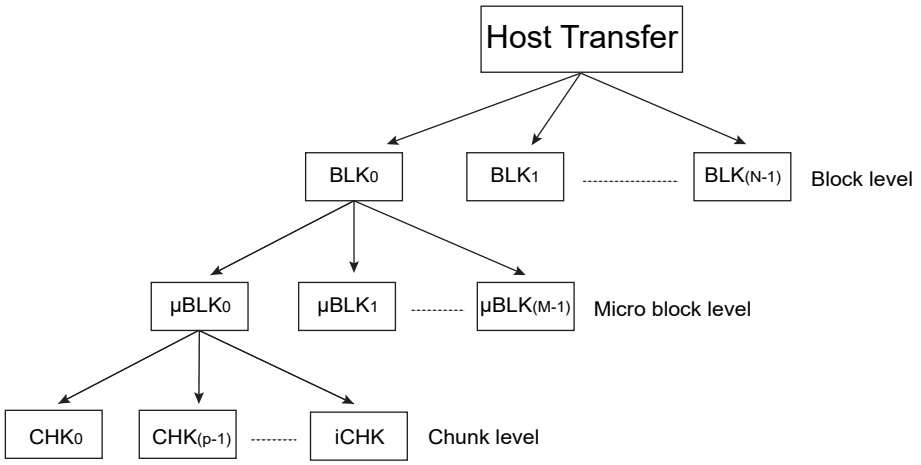
XDMAC Chunk and Incomplete Chunk: When a peripheral to memory transfer is activated, the burst level transaction is further split into a number of data chunks. The chunk size is configured in CSIZE field of XDMAC Channel Configuration Register (XDMAC_CCx). The chunk size denotes the number of 'data' 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.

16.5.4.2. Memory to Peripheral Transfer

XDMAC reads data from the source memory location and writes to the destination peripheral.

Figure 16.5. Memory to Peripheral Transfer Hierarchy



Memory to Peripheral transfer is also a peripheral synchronized transfer. It has a total of 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 a 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 the 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 chosen is 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. 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 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.

16.5.4.3. 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.

16.5.5. 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.

16.5.5.1. 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. Configure XDMAC_CCx.CSIZE to configure the channel chunk size (only relevant for peripheral synchronized transfer).
   f. Configure XDMAC_CCx.DWIDTH to configure the transfer data width.
   g. Configure XDMAC_CCx.SIF, XDMAC_CCx.DIF to configure the host interface used to read data and write data, respectively.
   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.

16.5.5.2.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.

16.5.5.3.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 and bit XDMAC_CNDAx.NDAIF with the host interface identifier.
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.

16.5.5.4. 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.

16.6. Linked List Descriptor Operation

16.6.1. Linked List Descriptor View

16.6.1.1. Channel Next Descriptor View 0–3 Structures

Table 16.2. Channel Next Descriptor View 0–3 Structures

Channel Next Descriptor	Offset	Structure Member	Name
View 0 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	XDMAC_MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	XDMAC_MBR_UBC
	DSCR_ADDR+0x08	Transfer Address Member	XDMAC_MBR_TA
View 1 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	XDMAC_MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	XDMAC_MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	XDMAC_MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	XDMAC_MBR_DA
View 2 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	XDMAC_MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	XDMAC_MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	XDMAC_MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	XDMAC_MBR_DA
	DSCR_ADDR+0x10	Configuration Register	XDMAC_MBR_CFG
View 3 Structure	DSCR_ADDR+0x00	Next Descriptor Address Member	XDMAC_MBR_NDA
	DSCR_ADDR+0x04	Microblock Control Member	XDMAC_MBR_UBC
	DSCR_ADDR+0x08	Source Address Member	XDMAC_MBR_SA
	DSCR_ADDR+0x0C	Destination Address Member	XDMAC_MBR_DA
	DSCR_ADDR+0x10	Configuration Member	XDMAC_MBR_CFG
	DSCR_ADDR+0x14	Block Control Member	XDMAC_MBR_BC
	DSCR_ADDR+0x18	Data Stride Member	XDMAC_MBR_DS
	DSCR_ADDR+0x1C	Source Microblock Stride Member	XDMAC_MBR_SUS
	DSCR_ADDR+0x20	Destination Microblock Stride Member	XDMAC_MBR_DUS

16.6.2. Descriptor Structure Members Description

16.6.2.1.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 “Bus Matrix (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.

16.7. XDMAC Maintenance Software Operations

16.7.1. 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.

16.7.2. 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.

16.7.3. 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.

16.7.4. Maintenance Operation Priority

16.7.4.1. 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.

16.7.4.2. 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.

16.7.4.3. Suspend Operation Priority

If the suspend operation is performed after a disable request, the write suspend operation is ignored.

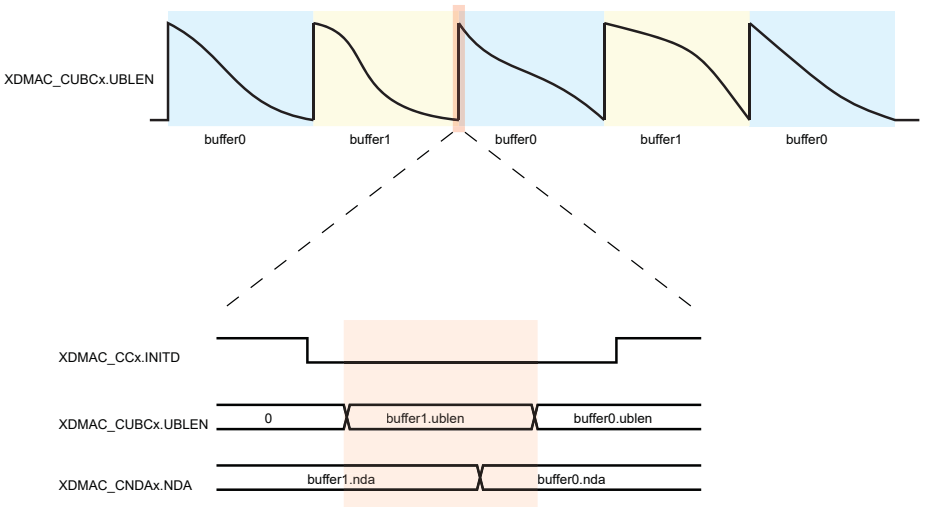
16.8. XDMAC Software Requirements

- 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 or 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.

Figure 16.6. INITD Timing Diagram



16.9. 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								
		23:16								
		15:8								BXKBEN
		7:0					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								
		23:16								
		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								
		23:16								
		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								
		23:16								
		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								
		23:16								
		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								
		23:16								
		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								
		23:16								
		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								
		23:16								
		15:8	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
		7:0	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
0x28	XDMAC_GRS	31:24								
		23:16								
		15:8	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
		7:0	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
0x2C	XDMAC_GWS	31:24								
		23:16								
		15:8	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
		7:0	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
0x30	XDMAC_GRWS	31:24								
		23:16								
		15:8	RWS15	RWS14	RWS13	RWS12	RWS11	RWS10	RWS9	RWS8
		7:0	RWS7	RWS6	RWS5	RWS4	RWS3	RWS2	RWS1	RWS0
0x34	XDMAC_GRWR	31:24								
		23:16								
		15:8	RWR15	RWR14	RWR13	RWR12	RWR11	RWR10	RWR9	RWR8
		7:0	RWR7	RWR6	RWR5	RWR4	RWR3	RWR2	RWR1	RWR0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	XDMAC_GSWR	31:24								
		23:16								
		15:8	SWREQ15	SWREQ14	SWREQ13	SWREQ12	SWREQ11	SWREQ10	SWREQ9	SWREQ8
		7:0	SWREQ7	SWREQ6	SWREQ5	SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0
0x3C	XDMAC_GSWS	31:24								
		23:16								
		15:8	SWRS15	SWRS14	SWRS13	SWRS12	SWRS11	SWRS10	SWRS9	SWRS8
		7:0	SWRS7	SWRS6	SWRS5	SWRS4	SWRS3	SWRS2	SWRS1	SWRS0
0x40	XDMAC_GSWF	31:24								
		23:16								
		15:8	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
		7:0	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
0x44 ... 0x4F	Reserved									
0x50	XDMAC_CIE0	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x54	XDMAC_CID0	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x58	XDMAC_CIM0	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x5C	XDMAC_CIS0	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x60	XDMAC_CSA0	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x64	XDMAC_CDA0	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x68	XDMAC_CNDA0	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x6C	XDMAC_CNDC0	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x70	XDMAC_CUBC0	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x74	XDMAC_CBC0	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x78	XDMAC_CC0	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x7C	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]							
0x80	XDMAC_CSUS0	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x84	XDMAC_CDUS0	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x88 ... 0x8F	Reserved									
0x90	XDMAC_CIE1	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x94	XDMAC_CID1	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x98	XDMAC_CIM1	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x9C	XDMAC_CIS1	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xA0	XDMAC_CSA1	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0xA4	XDMAC_CDA1	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0xA8	XDMAC_CNDA1	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0xAC	XDMAC_CNDC1	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0xB0	XDMAC_CUBC1	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0xB4	XDMAC_CBC1	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0xB8	XDMAC_CC1	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xBC	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]							
0xC0	XDMAC_CSUS1	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0xC4	XDMAC_CDUS1	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0xC8 ... 0xCF	Reserved									
0xD0	XDMAC_CIE2	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0xD4	XDMAC_CID2	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0xD8	XDMAC_CIM2	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0xDC	XDMAC_CIS2	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0xE0	XDMAC_CSA2	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0xE4	XDMAC_CDA2	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0xE8	XDMAC_CNDA2	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0xEC	XDMAC_CNDC2	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0xF0	XDMAC_CUBC2	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0xF4	XDMAC_CBC2	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0xF8	XDMAC_CC2	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xFC	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]							
0x0100	XDMAC_CSUS2	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0104	XDMAC_CDUS2	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0108 ... 0x010F	Reserved									
0x0110	XDMAC_CIE3	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0114	XDMAC_CID3	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0118	XDMAC_CIM3	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x011C	XDMAC_CIS3	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0120	XDMAC_CSA3	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0124	XDMAC_CDA3	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0128	XDMAC_CNDA3	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x012C	XDMAC_CNDC3	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0130	XDMAC_CUBC3	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0134	XDMAC_CBC3	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0138	XDMAC_CC3	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x013C	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]							
0x0140	XDMAC_CSUS3	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0144	XDMAC_CDUS3	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0148 ... 0x014F	Reserved									
0x0150	XDMAC_CIE4	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0154	XDMAC_CID4	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0158	XDMAC_CIM4	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x015C	XDMAC_CIS4	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0160	XDMAC_CSA4	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0164	XDMAC_CDA4	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0168	XDMAC_CNDA4	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x016C	XDMAC_CNDC4	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0170	XDMAC_CUBC4	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0174	XDMAC_CBC4	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0178	XDMAC_CC4	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x017C	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]							
0x0180	XDMAC_CSUS4	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0184	XDMAC_CDUS4	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0188 ... 0x018F	Reserved									
0x0190	XDMAC_CIE5	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0194	XDMAC_CID5	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0198	XDMAC_CIM5	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x019C	XDMAC_CIS5	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01A0	XDMAC_CSA5	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01A4	XDMAC_CDA5	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01A8	XDMAC_CNDA5	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x01AC	XDMAC_CNDC5	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x01B0	XDMAC_CUBC5	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x01B4	XDMAC_CBC5	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x01B8	XDMAC_CC5	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01BC	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]							
0x01C0	XDMAC_CSUS5	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x01C4	XDMAC_CDUS5	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x01C8 ... 0x01CF	Reserved									
0x01D0	XDMAC_CIE6	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x01D4	XDMAC_CID6	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x01D8	XDMAC_CIM6	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x01DC	XDMAC_CIS6	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x01E0	XDMAC_CSA6	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x01E4	XDMAC_CDA6	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x01E8	XDMAC_CNDA6	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x01EC	XDMAC_CNDC6	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x01F0	XDMAC_CUBC6	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x01F4	XDMAC_CBC6	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x01F8	XDMAC_CC6	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01FC	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]							
0x0200	XDMAC_CSUS6	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0204	XDMAC_CDUS6	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0208 ... 0x020F	Reserved									
0x0210	XDMAC_CIE7	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0214	XDMAC_CID7	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0218	XDMAC_CIM7	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x021C	XDMAC_CIS7	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0220	XDMAC_CSA7	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0224	XDMAC_CDA7	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0228	XDMAC_CNDA7	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x022C	XDMAC_CNDC7	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0230	XDMAC_CUBC7	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0234	XDMAC_CBC7	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0238	XDMAC_CC7	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x023C	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]							
0x0240	XDMAC_CSUS7	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0244	XDMAC_CDUS7	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0248 ... 0x024F	Reserved									
0x0250	XDMAC_CIE8	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0254	XDMAC_CID8	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0258	XDMAC_CIM8	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x025C	XDMAC_CIS8	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0260	XDMAC_CSA8	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0264	XDMAC_CDA8	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0268	XDMAC_CNDA8	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x026C	XDMAC_CNDC8	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0270	XDMAC_CUBC8	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0274	XDMAC_CBC8	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0278	XDMAC_CC8	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x027C	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]							
0x0280	XDMAC_CSUS8	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0284	XDMAC_CDUS8	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0288 ... 0x028F	Reserved									
0x0290	XDMAC_CIE9	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0294	XDMAC_CID9	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0298	XDMAC_CIM9	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x029C	XDMAC_CIS9	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02A0	XDMAC_CSA9	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02A4	XDMAC_CDA9	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02A8	XDMAC_CNDA9	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x02AC	XDMAC_CNDC9	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x02B0	XDMAC_CUBC9	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x02B4	XDMAC_CBC9	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x02B8	XDMAC_CC9	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02BC	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]							
0x02C0	XDMAC_CSUS9	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x02C4	XDMAC_CDUS9	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x02C8 ... 0x02CF	Reserved									
0x02D0	XDMAC_CIE10	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x02D4	XDMAC_CID10	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x02D8	XDMAC_CIM10	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x02DC	XDMAC_CIS10	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x02E0	XDMAC_CSA10	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x02E4	XDMAC_CDA10	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x02E8	XDMAC_CNDA10	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x02EC	XDMAC_CNDC10	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x02F0	XDMAC_CUBC10	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x02F4	XDMAC_CBC10	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x02F8	XDMAC_CC10	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02FC	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]							
0x0300	XDMAC_CSUS10	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0304	XDMAC_CDUS10	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0308 ... 0x030F	Reserved									
0x0310	XDMAC_CIE11	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0314	XDMAC_CID11	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0318	XDMAC_CIM11	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x031C	XDMAC_CIS11	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0320	XDMAC_CSA11	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0324	XDMAC_CDA11	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0328	XDMAC_CNDA11	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x032C	XDMAC_CNDC11	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0330	XDMAC_CUBC11	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0334	XDMAC_CBC11	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0338	XDMAC_CC11	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x033C	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]							
0x0340	XDMAC_CSUS11	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0344	XDMAC_CDUS11	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0348 ... 0x034F	Reserved									
0x0350	XDMAC_CIE12	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0354	XDMAC_CID12	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0358	XDMAC_CIM12	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x035C	XDMAC_CIS12	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0360	XDMAC_CSA12	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0364	XDMAC_CDA12	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0368	XDMAC_CNDA12	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x036C	XDMAC_CNDC12	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0370	XDMAC_CUBC12	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0374	XDMAC_CBC12	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0378	XDMAC_CC12	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x037C	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]							
0x0380	XDMAC_CSUS12	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0384	XDMAC_CDUS12	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0388 ... 0x038F	Reserved									
0x0390	XDMAC_CIE13	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0394	XDMAC_CID13	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0398	XDMAC_CIM13	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x039C	XDMAC_CIS13	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03A0	XDMAC_CSA13	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03A4	XDMAC_CDA13	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03A8	XDMAC_CNDA13	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x03AC	XDMAC_CNDC13	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x03B0	XDMAC_CUBC13	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x03B4	XDMAC_CBC13	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x03B8	XDMAC_CC13	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03BC	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]							
0x03C0	XDMAC_CSUS13	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x03C4	XDMAC_CDUS13	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x03C8 ... 0x03CF	Reserved									
0x03D0	XDMAC_CIE14	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x03D4	XDMAC_CID14	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x03D8	XDMAC_CIM14	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x03DC	XDMAC_CIS14	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x03E0	XDMAC_CSA14	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x03E4	XDMAC_CDA14	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x03E8	XDMAC_CNDA14	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x03EC	XDMAC_CNDC14	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x03F0	XDMAC_CUBC14	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x03F4	XDMAC_CBC14	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x03F8	XDMAC_CC14	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03FC	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]							
0x0400	XDMAC_CSUS14	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0404	XDMAC_CDUS14	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							
0x0408 ... 0x040F	Reserved									
0x0410	XDMAC_CIE15	31:24								
		23:16								
		15:8								
		7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
0x0414	XDMAC_CID15	31:24								
		23:16								
		15:8								
		7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
0x0418	XDMAC_CIM15	31:24								
		23:16								
		15:8								
		7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
0x041C	XDMAC_CIS15	31:24								
		23:16								
		15:8								
		7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
0x0420	XDMAC_CSA15	31:24	SA[31:24]							
		23:16	SA[23:16]							
		15:8	SA[15:8]							
		7:0	SA[7:0]							
0x0424	XDMAC_CDA15	31:24	DA[31:24]							
		23:16	DA[23:16]							
		15:8	DA[15:8]							
		7:0	DA[7:0]							
0x0428	XDMAC_CNDA15	31:24	NDA[29:22]							
		23:16	NDA[21:14]							
		15:8	NDA[13:6]							
		7:0	NDA[5:0]							NDAIF
0x042C	XDMAC_CNDC15	31:24								
		23:16								
		15:8								
		7:0		QOS[1:0]		NDVIEW[1:0]		NDDUP	NDSUP	NDE
0x0430	XDMAC_CUBC15	31:24								
		23:16	UBLEN[23:16]							
		15:8	UBLEN[15:8]							
		7:0	UBLEN[7:0]							
0x0434	XDMAC_CBC15	31:24								
		23:16								
		15:8					BLEN[11:8]			
		7:0	BLEN[7:0]							
0x0438	XDMAC_CC15	31:24		PERID[6:0]						
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x043C	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]							
0x0440	XDMAC_CSUS15	31:24								
		23:16	SUBS[23:16]							
		15:8	SUBS[15:8]							
		7:0	SUBS[7:0]							
0x0444	XDMAC_CDUS15	31:24								
		23:16	DUBS[23:16]							
		15:8	DUBS[15:8]							
		7:0	DUBS[7:0]							

16.9.1. XDMAC Global Type Register

Name: XDMAC_GTYPE
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
		NB_REQ[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
	FIFO_SZ[10:3]							
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
	FIFO_SZ[2:0]			NB_CH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 22:16 – NB_REQ[6:0] Number of Peripheral Requests Minus One

Bits 15:5 – FIFO_SZ[10:0] Number of Bytes

Bits 4:0 – NB_CH[4:0] Number of Channels Minus One

16.9.2. XDMAC Global Configuration Register

Name: XDMAC_GCFG
Offset: 0x04
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								R/W
Reset								0

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

Bit 8 – BXKBEN Boundary X Kilobyte Enable

Value	Description
0	XDMAC generates a non-sequential attribute on the system bus when address crosses the 1-Kilobyte boundary with a burst access.
1	XDMAC does not generate a non-sequential attribute on the system bus when address crosses the 1-Kilobyte boundary with a burst access.

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.

16.9.3. XDMAC Global Weighted Arbiter Configuration Register

Name: XDMAC_GWAC
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
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
This field indicates the weight of pool 3 in the arbitration scheme of the DMA scheduler.

Bits 11:8 – PW2[3:0] Pool Weight 2
This field indicates the weight of pool 2 in the arbitration scheme of the DMA scheduler.

Bits 7:4 – PW1[3:0] Pool Weight 1
This field indicates the weight of pool 1 in the arbitration scheme of the DMA scheduler.

Bits 3:0 – PW0[3:0] Pool Weight 0
This field indicates the weight of pool 0 in the arbitration scheme of the DMA scheduler.

16.9.4. XDMAC Global Interrupt Enable Register

Name: XDMAC_GIE
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
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 – 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.

16.9.5. XDMAC Global Interrupt Disable Register

Name: XDMAC_GID
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
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 – 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.

16.9.6. XDMAC Global Interrupt Mask Register

Name: XDMAC_GIM
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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 – 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.

16.9.7. XDMAC Global Interrupt Status Register

Name: XDMAC_GIS
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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 – 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.

16.9.8. XDMAC Global Channel Enable Register

Name: XDMAC_GE
Offset: 0x1C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
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 – 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'.

16.9.9. XDMAC Global Channel Disable Register

Name: XDMAC_GD
Offset: 0x20
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
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 – DIx XDMAC Channel x Disable

Value	Description
0	This bit has no effect.
1	Disables channel x.

16.9.10. XDMAC Global Channel Status Register

Name: XDMAC_GS
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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 – 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.

16.9.11. XDMAC Global Channel Read Suspend Register

Name: XDMAC_GRS
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
	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
Access	R/W	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
	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Access	R/W	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 – RSx XDMAC Channel x Read Suspend

Value	Description
0	The read channel is not suspended.
1	The source requests for channel n are no longer serviced by the system scheduler.

16.9.12. XDMAC Global Channel Write Suspend Register

Name: XDMAC_GWS
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
	WS15	WS14	WS13	WS12	WS11	WS10	WS9	WS8
Access	R/W	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
	WS7	WS6	WS5	WS4	WS3	WS2	WS1	WS0
Access	R/W	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 – WSx XDMAC Channel x Write Suspend

Value	Description
0	The write channel is not suspended.
1	Destination requests are no longer routed to the scheduler.

16.9.13. XDMAC Global Channel Read Write Suspend Register

Name: XDMAC_GRWS
Offset: 0x30
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
	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 – RWSx XDMAC Channel x Read Write Suspend

Value	Description
0	No effect.
1	Read and write requests are suspended.

16.9.14. XDMAC Global Channel Read Write Resume Register

Name: XDMAC_GRWR
Offset: 0x34
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
	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 – RWRx XDMAC Channel x Read Write Resume

Value	Description
0	No effect.
1	Read and write requests are serviced.

16.9.15. XDMAC Global Channel Software Request Register

Name: XDMAC_GSWR
Offset: 0x38
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
	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 – SWREQx XDMAC Channel x Software Request

Value	Description
0	No effect.
1	Requests a DMA transfer for channel x.

16.9.16. XDMAC Global Channel Software Request Status Register

Name: XDMAC_GSWS
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
	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 – SWRSx XDMAC Channel x Software Request Status

Value	Description
0	Channel x source request is serviced.
1	Channel x source request is pending.

16.9.17. XDMAC Global Channel Software Flush Request Register

Name: XDMAC_GSWF
Offset: 0x40
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
	SWF15	SWF14	SWF13	SWF12	SWF11	SWF10	SWF9	SWF8
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	SWF7	SWF6	SWF5	SWF4	SWF3	SWF2	SWF1	SWF0
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 – 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.

16.9.18. XDMAC Channel x Interrupt Enable Register [x=0..15]

Name: XDMAC_CIE
Offset: 0x50 + n*0x40 [n=0..15]
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
		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

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.

16.9.19. XDMAC Channel x Interrupt Disable Register [x = 0..15]

Name: XDMAC_CID
Offset: 0x54 + n*0x40 [n=0..15]
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
		ROID	WBEID	RBEID	FID	DID	LID	BID
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

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.

16.9.20. XDMAC Channel x Interrupt Mask Register [x = 0..15]

Name: XDMAC_CIM
Offset: 0x58 + n*0x40 [n=0..15]
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
		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

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.

16.9.21. XDMAC Channel x Interrupt Status Register [x = 0..15]

Name: XDMAC_CIS
Offset: 0x5C + n*0x40 [n=0..15]
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
		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

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.

16.9.22. XDMAC Channel x Source Address Register [x = 0..15]

Name: XDMAC_CSA
Offset: 0x60 + n*0x40 [n=0..15]
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.

16.9.23. XDMAC Channel x Destination Address Register [x = 0..15]

Name: XDMAC_CDA
Offset: 0x64 + n*0x40 [n=0..15]
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.

16.9.24. XDMAC Channel x Next Descriptor Address Register [x = 0..15]

Name: XDMAC_CNDA
Offset: 0x68 + n*0x40 [n=0..15]
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]							NDAIF
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 – 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.

Bit 0 – NDAIF Channel x Next Descriptor Interface

Value	Description
0	The channel descriptor is retrieved through system interface 0.
1	The channel descriptor is retrieved through system interface 1.

16.9.25. XDMAC Channel x Next Descriptor Control Register [x = 0..15]

Name: XDMAC_CNDC
Offset: 0x6C + n*0x40 [n=0..15]
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 “Bus Matrix (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

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.

Value	Name	Description
1	DSCR_FETCH_EN	Descriptor fetch is enabled.

16.9.26. XDMAC Channel x Microblock Control Register [x = 0..15]

Name: XDMAC_CUBC
Offset: 0x70 + n*0x40 [n=0..15]
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.

16.9.27. XDMAC Channel x Block Control Register [x = 0..15]

Name: XDMAC_CBC
Offset: 0x74 + n*0x40 [n=0..15]
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.

16.9.28. XDMAC Channel x Configuration Register [x = 0..15]

Name: XDMAC_CC
Offset: 0x78 + n*0x40 [n=0..15]
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
		DIF	SIF	DWIDTH[1:0]		CSIZE[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	7	6	5	4	3	2	1	0
	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	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 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.

Bit 14 – DIF Channel x Destination Interface Identifier

Value	Name	Description
0	AHB_IF0	The data is written through system bus interface 0.
1	AHB_IF1	The data is written through system bus interface 1.

Bit 13 – SIF Channel x Source Interface Identifier

Value	Name	Description
0	AHB_IF0	The data is read through system bus interface 0.
1	AHB_IF1	The data is read through system bus interface 1.

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

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 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)

16.9.29. XDMAC Channel x Data Stride Memory Set Pattern Register [x = 0..15]

Name: XDMAC_CDS_MSP
Offset: 0x7C + n*0x40 [n=0..15]
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.

16.9.30. XDMAC Channel x Source Microblock Stride Register [x = 0..15]

Name: XDMAC_CSUS
Offset: 0x80 + n*0x40 [n=0..15]
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 16.2).
The SUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

16.9.31. XDMAC Channel x Destination Microblock Stride Register [x = 0..15]

Name: XDMAC_CDUS
Offset: 0x84 + n*0x40 [n=0..15]
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 16.2).
The DUBS field is only relevant when XDMAC_CCx.SAM=UBS_AM or XDMAC_CCx.SAM=UBS_DS_AM.

17. Boot Strategies

17.1. Standard Boot Strategy

17.1.1. Description

The chip always boots from the ROM memory at start-up or after a reset.

The ROM code is a boot program contained in the embedded ROM. It is also called “Bootloader”.

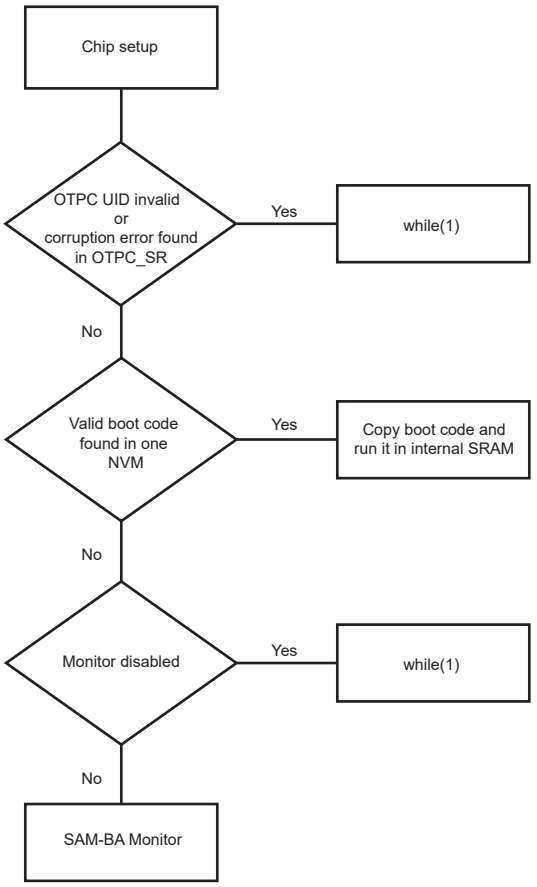
By default, the chip starts in Standard Boot mode. The user can choose to enable the Secure Boot mode. To know how to set up and configure the Secure Boot mode, see Secure Boot Strategy.

Note: JTAG access is disabled during the main part of the execution of the ROM code. It is re-enabled when the Standard SAM-BA® Monitor is executed or when jumping into a valid bootstrap.

17.1.2. Flow Diagram

The following figure shows the ROM code global flow.

Figure 17.1. ROM Code Flow Diagram



17.1.3. Chip Setup

When the chip is powered on, the system source clock is the Main RC oscillator. No external crystal or clock is needed during the external memories boot sequence.

The ROM code performs a low-level initialization that follows the steps described below:

1. PLLA initialization

2. Main system bus clock selection: when the PLLA is stabilized, the main system bus clock source is switched from the main clock to the PLLA clock. The PMC Status register is polled to wait for MCK Ready.

Once the clock configuration is done, the ROM code initializes the ROM code console interface and sends a `RomBOOT` character string on it.

For clock frequencies, see Table 17.14.

Note: The ROM code only seeks an external clock source before launching the SAM-BA Monitor to get a more accurate clock signal for USB.

17.1.4. Boot Configuration

The boot sequence is controlled using a Boot Configuration Packet, stored in the OTP User area and accessed through the OTP Controller (OTPC).

17.1.4.1. Default Boot Sequence

When no Boot Configuration Packet is available in the OTP User area, the ROM code uses a default boot configuration and tries to boot from the following external memories:

- SD Card/e.MMC (SDMMC 0) - PA0 to PA5
- SD Card/e.MMC (SDMMC 1) - PA6 to PA11
- QSPI NOR Flash (QSPI) - PB15 to PB26
- NAND Flash (SMC) - PD0 to PD14
- SPI (FLEXCOM 5) - PA14 to PA17

See Table 17.15 for further details.

If no bootable file is found in any of these memories, the ROM code runs the SAM-BA Monitor.

17.1.4.2. Using a Boot Configuration Packet

Several boot configuration parameters can be customized by writing a Boot Configuration Packet in the OTP User area:

- IO configuration of boot memories (see Hardware and Software Constraints for a description of the IOs)
- Boot sequence
- UART port for the ROM code console
- Monitor disable

See Boot Configuration Packet for a detailed description of all the fields in this packet.

Out of factory, the OTP User area is empty and thus contains no Boot Configuration Packet.

OTP Emulation Mode

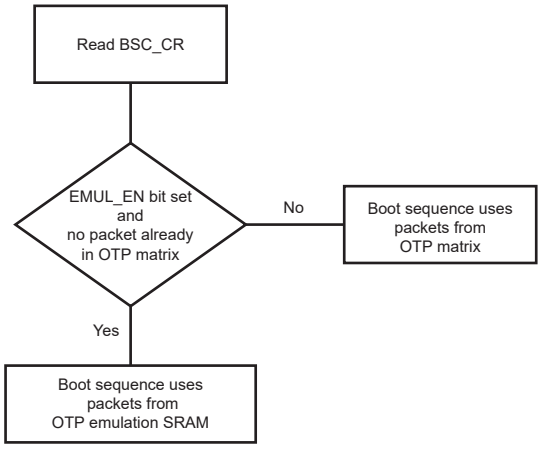
During a prototyping phase or for test purposes, it is possible to use the OTP Emulation mode. In this mode, the OTP matrix is replaced by an internal SRAM1. Once the Emulation mode is enabled, every OTP packet read and write is done in the internal SRAM1 instead of the OTP matrix. To instruct the ROM code to use this feature, set the bit EMUL_EN in the Boot Sequence Controller Configuration register (BSC_CR).

When using the emulated OTP, the user can test several boot configuration options, including the Secure Boot mode, without programming the OTP.

Notes:

- The emulation SRAM is not backed up. After a power off/power on sequence, the configuration and content are lost.
- Once a packet is written in the OTP matrix, the OTP Emulation mode is no more available. Any instruction from the user to enable this mode in BSC_CR will fail, and packets will be read or written in the OTP memory.

Figure 17.2. Boot Configuration Loading



17.1.4.3.Boot Sequence Controller Configuration Register

Name: BSC_CR

Address: 0xFFFFFE54

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 EMUL_EN. Always read as 0.

Bit 0 – EMUL_EN Request to use the OTP Emulation Mode for Boot Features

Value	Description
0	OTP Emulation mode is disabled.
1	The ROM code will use OTP Emulation mode for the boot features.

17.1.4.4.Boot Configuration Packet

Table 17.1. Boot Configuration Packet Structure

Offset	Name	Description
0x00	MON_DIS	Disables the SAM-BA Monitor
0x04	RESERVED	Reserved
0x08	CONSOLE_PIN	Defines the ROM code console pin muxing
0xC-0x44	MEM_CFGx[2]	Array of two 32-bit words used to set memory configurations for x=0..6 MEM_CFGx[0]: this word contains mandatory options for the connected external Flash memories. MEM_CFGx[1]: this word contains optional data specific to the memory described in word 0.

17.1.4.4.1.Monitor Disable Word

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] SAM-BA Monitor Disable

Value	Description
0	If no bootstrap file is found in an external bootable Flash memory, the SAM-BA Monitor is launched.
Non-Zero	The SAM-BA Monitor is never executed.

17.1.4.4.2.ROM Code Console Pin Muxing Word

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
	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	CONSOLE_IOSET[3:0]			
Access								
Reset								

Bits 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

Bits 3:0 – CONSOLE_IOSET[3:0] Selects the pins and UART interface used for the ROM code console terminal

Value	Name	Description
0	DBGU	Uses DBGU
1	FLEXCOM0	Uses FLEXCOM0 UART pins
2	FLEXCOM1	Uses FLEXCOM1 UART pins
3	FLEXCOM2	Uses FLEXCOM2 UART pins
4	FLEXCOM3	Uses FLEXCOM3 UART pins
5	FLEXCOM4	Uses FLEXCOM4 UART pins
6	FLEXCOM5	Uses FLEXCOM5 UART pins
7	FLEXCOM6	Uses FLEXCOM6 UART pins
8	FLEXCOM7	Uses FLEXCOM7 UART pins
9	FLEXCOM8	Uses FLEXCOM8 UART pins
10	FLEXCOM9	Uses FLEXCOM9 UART pins
11	FLEXCOM10	Uses FLEXCOM10 UART pins
12	FLEXCOM11	Uses FLEXCOM11 UART pins
13	FLEXCOM12	Uses FLEXCOM12 UART pins

17.1.4.4.3.QSPI Memory Boot Configuration (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	Memory boot interface disabled
1	Boot on QSPI memory enabled

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	QSPI0

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1

17.1.4.4.4.QSPI Memory Boot Configuration (Second Word)

Name: MEM_CFGx[1]

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	DNU
Access								
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 – DNU Do Not Use

17.1.4.4.5.SDMMC Memory Boot Configuration (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] Boot memory interface type

Value	Description
0	Memory boot interface disabled
3	Boot on SD card/e.MMC memory enabled

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	SDMMC0
1	SDMMC1

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1

17.1.4.4.6.SD Card/e.MMC Memory Boot Configuration (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
	USE_CD	PIO_ID[1:0]		PIN[4:0]				
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 other bit values are ignored.

Bit 7 – USE_CD Use a Card Detect pin

Value	Description
0	Use Card Detect disabled: the ROM code does not use any card detect pin and directly tries to boot from the memory connected to the SDMMC controller.
1	Use Card Detect enabled: the ROM code checks the state of the card detect pin. If the pin level is 0, the ROM code tries to boot from a memory connected to the SDMMC interface. If the pin level is 1, the ROM code skips the boot on SDMMC and jumps to the next interface in the boot sequence.

Bits 6:5 – PIO_ID[1:0] PIO Controller ID
ID of the PIO controller featuring the card detect pin.

Value	Description
0	PIOA
1	PIOB
2	PIOC
3	PIOD

Bits 4:0 – PIN[4:0] Card Detect Pin Index
Index of the card detect pin inside the PIO controller.

17.1.4.4.7.SPI Memory Boot Configuration (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	Memory boot interface disabled
2	Boot on SPI memory enabled

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	FLEXCOM0 SPI
1	FLEXCOM1 SPI
2	FLEXCOM2 SPI
3	FLEXCOM3 SPI
4	FLEXCOM4 SPI
5	FLEXCOM5 SPI

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1
1	PIO set 2
2	PIO set 3
3	PIO set 4
4	PIO set 5

Value	Description
5	PIO set 6

17.1.4.4.8.SPI Memory Boot Configuration (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	DNU
Access								
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 – DNU Do Not Use

17.1.4.4.9.NAND Flash Memory Boot Configuration (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	Memory boot interface disabled
4	Boot on NAND Flash memory enabled

Bits 7:4 – INSTANCE_ID[3:0] IP Instance ID

Value	Description
0	SMC

Bits 3:0 – IFACE_IOSET[3:0] Memory IOSET

Value	Description
0	PIO set 1

17.1.4.4.10.NAND Flash Memory Boot Configuration (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	DNU
Access								
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 – DNU DO NOT USE

17.1.4.5.NVM Boot Sequence

For each memory listed in the default boot sequence or enabled in the Boot Configuration Packet, the ROM code performs the flash initialization and valid code detection as described below.

Figure 17.3. NVM Bootloader Program

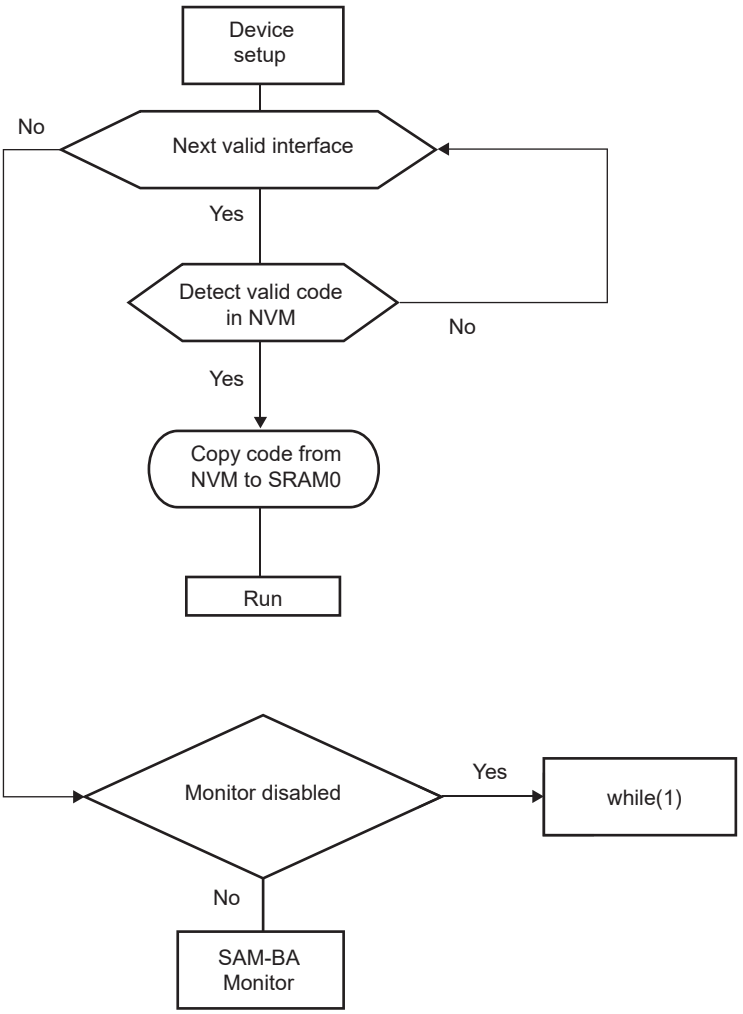
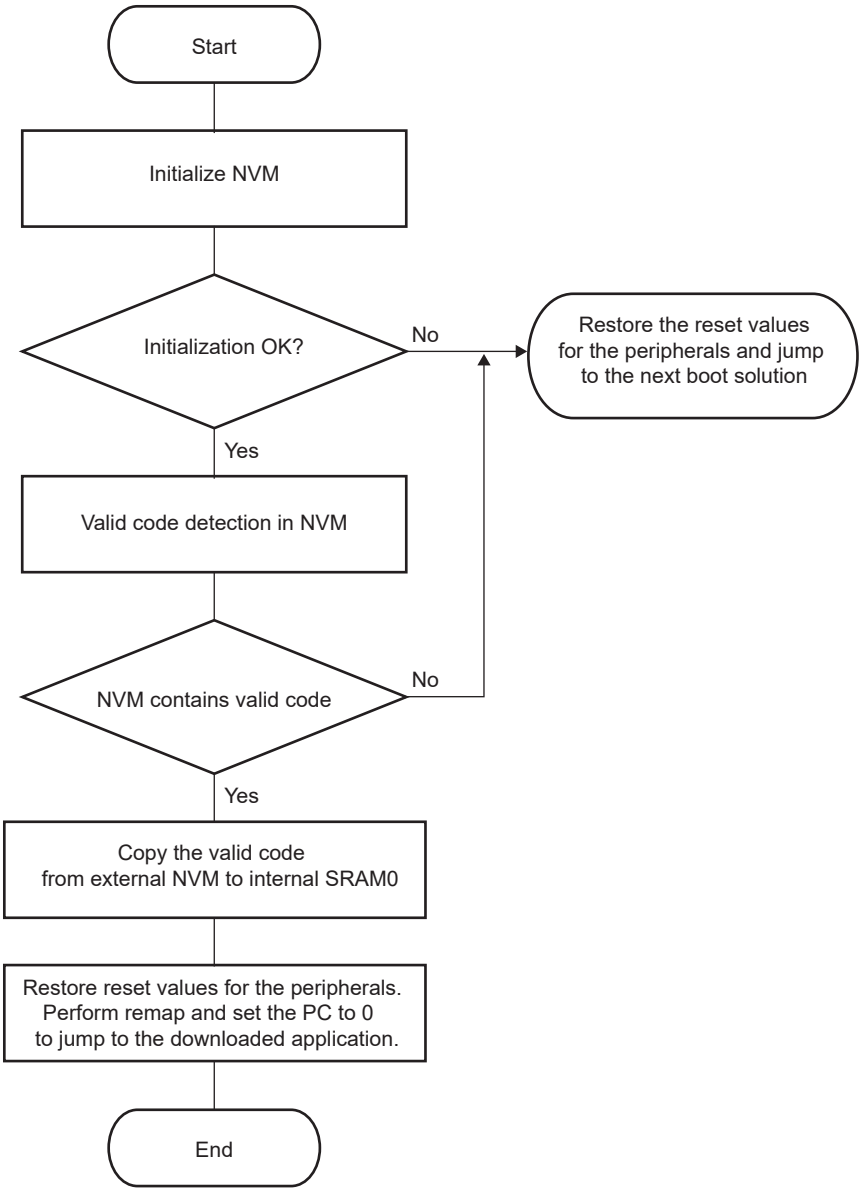


Figure 17.4. NVM Boot Diagram



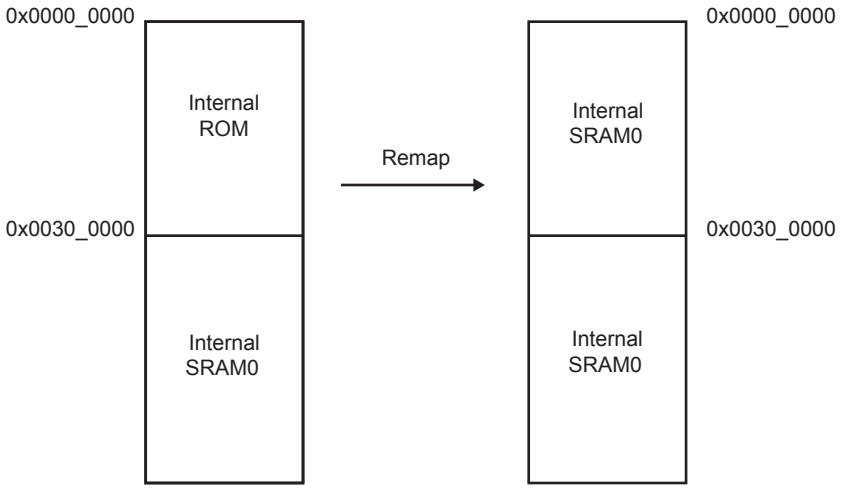
The NVM bootloader program first initializes the PIOs related to the NVM device. Then it configures the right peripheral depending on the NVM and tries to access this memory. If the initialization fails, it restores the reset values for the PIO and the peripheral, and then tries to perform the same operations on the next NVM of the sequence.

If the initialization is successful, the NVM bootloader program reads the beginning of the NVM and determines if the NVM contains a valid code.

If the NVM does not contain a valid code, the NVM bootloader program restores the reset value for the peripherals and then tries to perform the same operations on the next NVM of the sequence.

If a valid code is found, this code is loaded from the NVM into the internal SRAM0 and executed by branching at address 0x0000_0000 after remap. This code may be the application code or a second-level bootloader. The code size must be lower than 32 KB.

Figure 17.5. Remap Action after Download Completion



17.1.4.6.Valid Code Detection

Two types of valid code detection are available:

- Arm Exception Vectors Check
- boot.bin File Check

17.1.4.6.1.Arm Exception Vectors Check

The ROM code reads and analyzes the first 28 bytes corresponding to the first seven Arm exception vectors of the bootstrap.

These exception vectors must implement either a branch or a load with PC-relative addressing Arm instruction, except for the 6th vector.

More precisely, the ROM code must read either 0xEA or 0xE5 in bytes 3, 7, 11, 15 and 23.

The sixth vector (32 bits), at offset 0x14, must contain the size in bytes of the bootstrap to download from the external NVM into the internal SRAM.

The bootstrap size must not exceed the maximum bootstrap size allowed.

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              00008000                    ← Code size = 32 kbytes
18              eafffffe                    B 0x18
```

17.1.4.6.2.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 bootstrap size allowed.

17.1.4.7.Detailed Memory Boot Procedures

17.1.4.7.1.NAND Flash Boot: NAND Flash Detection

After the NAND Flash interface configuration, a reset command is sent to the memory.

The reset time of the NAND memory, after this reset command, must not be higher than 100 µs.

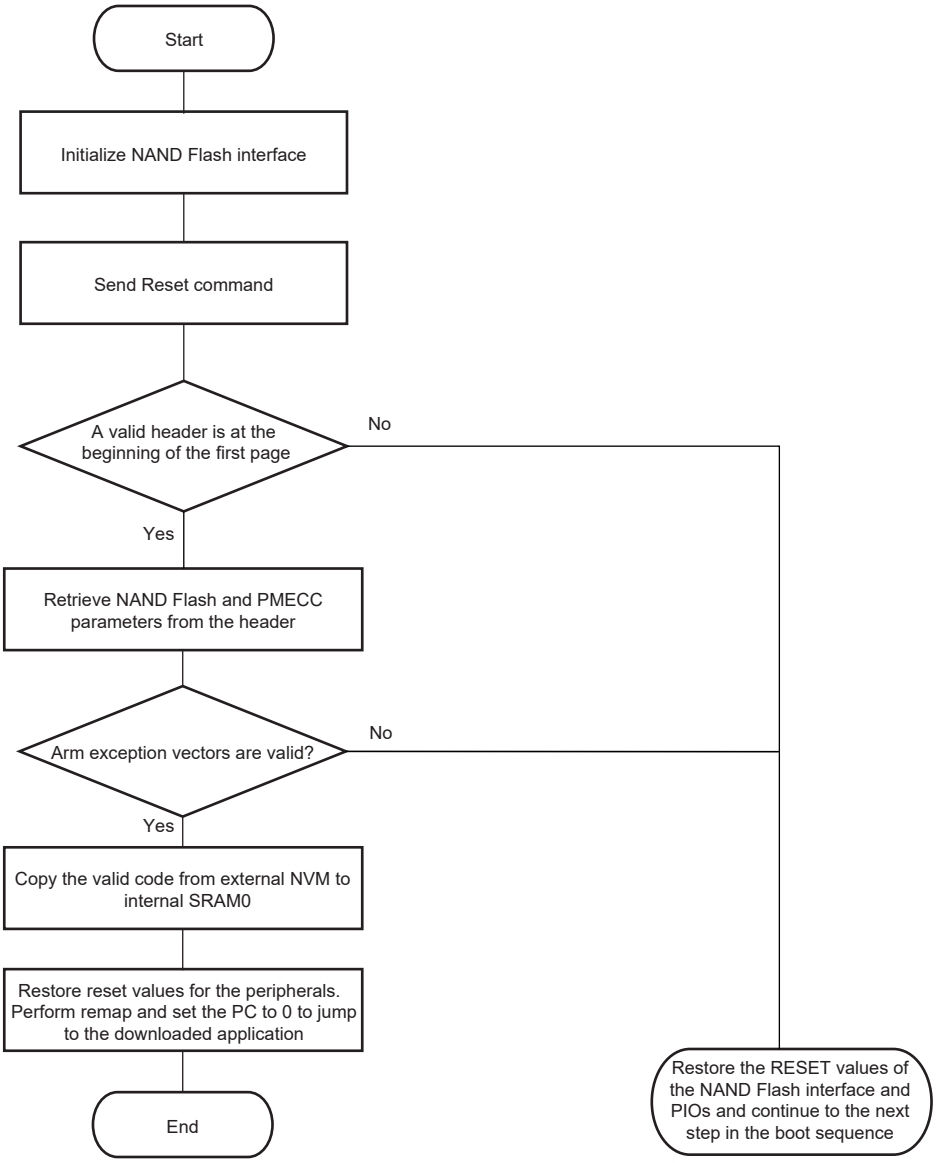
Hardware ECC detection and correction are provided by the PMECC peripheral. Refer to the section Functional Description in Programmable Multibit Error Correction Code Controller (PMECC) for more details.

The ROM code retrieves NAND Flash parameters and ECC requirements using a specific header store at the beginning of the bootstrap.

Once the ROM code has got the ECC parameters, it reads the first page again, with or without ECC, depending on the usePmecc parameter. Then it looks for a valid code programmed just after the header offset 0xD0. If the code is valid, the program is copied at the beginning of the internal SRAM.

Note: Booting on 16-bit NAND Flash is not possible; only 8-bit NAND Flash memories are supported.

Figure 17.6. Boot NAND Flash Download



NAND Flash Specific Header Detection

After receiving the Initialization and Reset command, the ROM code reads the first page without an ECC check, to determine whether the NAND parameter header is present. The header is made of 52 times the same 32-bit word (for redundancy reasons) which must contain NAND and PMECC parameters used to correctly perform the read of the rest of the data in the NAND. This 32-bit word is described below.

If the header is valid, the Boot program continues with the detection of a valid code.

[NAND Flash Specific Header Detection]

Name: NAND Flash Specific Header Detection

Bit	31	30	29	28	27	26	25	24
	key[3:0]					eccOffset[8:6]		
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	eccOffset[5:0]						sectorSize[1:0]	
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	eccBitReq[2:0]			spareSize[8:4]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	spareSize[3:0]				nbSectorPerPage[2:0]			usePmecc
Access								
Reset								

Bits 31:28 – key[3:0] Value 0xC Must be Written here to Validate the Content of the Whole Word

Bits 26:18 – eccOffset[8:0] Offset of the First ECC Byte in the Spare Zone
A value below 2 is not allowed and is considered as 2.

Bits 17:16 – sectorSize[1:0] Size of the ECC Sector

Value	Description
0	For 512 bytes
1	For 1024 bytes per sector
Other values	For future use

Bits 15:13 – eccBitReq[2:0] Number of ECC Bits Required

Value	Description
0	2-bit ECC
1	4-bit ECC
2	8-bit ECC
3	12-bit ECC
4	24-bit ECC

Bits 12:4 – spareSize[8:0] Size of the Spare Zone in Bytes

Bits 3:1 – nbSectorPerPage[2:0] Number of Sectors per Page

Value	Description
0	1 sector per page
1	2 sectors per page
2	4 sectors per page
3	8 sectors per page
4	16 sectors per page

Bit 0 – usePmecc Use PMECC

Value	Description
0	Do not use PMECC to detect and correct the data.
1	Use PMECC to detect and correct the data.

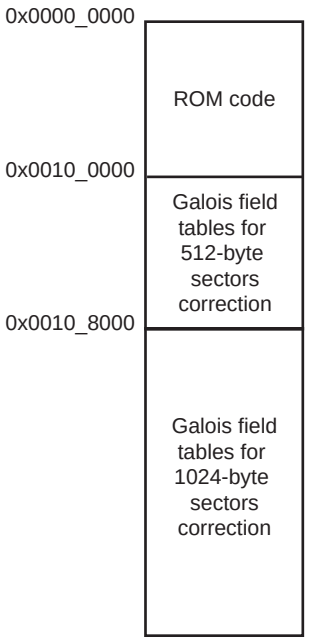
17.1.4.7.2.NAND Flash Boot: PMECC Error Detection and Correction

NAND Flash boot procedure uses PMECC to detect and correct errors during NAND Flash read operations

The ROM memory embeds the Galois field tables. The user does not need to embed or generate them in his/her own application.

The Galois field tables are mapped in the ROM just after the ROM code, as shown in the following figure.

Figure 17.7. Galois Field Table Mapping



17.1.4.8.SD Card/e.MMC Boot

When the Card Detect pin option is enabled in the Boot Configuration Packet, and if the level of the Card Detect pin is high (no card inserted), the ROM code will not try to initialize an SD card/e.MMC (no PIO will toggle). Otherwise, the SD card/e.MMC access is initiated (SDMMC interface PIO will toggle).

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 “maximum bootstrap size” bytes 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.

17.1.4.9.SPI Flash Boot

Two types of SPI Flash are supported: SPI Serial Flash and SPI DataFlash.

The ROM code tries to boot on SPI, first looking for SPI Serial Flash, and then for SPI DataFlash.

It uses the analysis of Arm exception vectors to detect a valid bootstrap code.

The SPI Flash read is done by means of a “Continuous Read” command from address 0x0. This command is 0xE8 for DataFlash devices and 0x0B for Serial Flash devices.

Supported DataFlash Devices

The ROM code supports the DataFlash devices listed in the following table.

Table 17.2. DataFlash Devices

Device	Density	Page Size (bytes)	Number of Pages
AT45DB011	1 Mbit	264	512
AT45DB021	2 Mbits	264	1024
AT45DB041	4 Mbits	264	2048
AT45DB081	8 Mbits	264	4096
AT45DB161	16 Mbits	528	4096
AT45DB321	32 Mbits	528	8192
AT45DB642	64 Mbits	1056	8192
AT45DB641	64 Mbits	264	37768

Supported Serial Flash Devices

The ROM code supports all SPI Serial Flash devices responding correctly to both “Get Status” and “Continuous Read” standard commands.

17.1.4.10.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 Serial Flash Discoverable Parameters (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 any of 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, this 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 17.3. 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 enter the 0-4-4 mode (XIP)	A0h	0h The ROM code first sets SIP bit[3] in the Volatile Configuration register (VCR)	0Fh	A5h	N/A
Value of mode cycles to exit the 0-4-4 mode (normal read)	00h	1h	00h	FFh	N/A
XIP supported	Yes	Yes	Yes	Yes	No

Those hard-coded parameters give a last chance to the ROM code to boot from a QSPI NOR memory in either Normal mode or XIP (Continuous Read) mode.

Table 17.4. QSPI NOR Memories Tested with and Supported by ROM Code (Non Exhaustive)

Manufacturer	Memories
Microchip (SST)	SST26VF016B SST26VF032B SST26VF032BA SST26VF064B

Table 17.4. QSPI NOR Memories Tested with and Supported by ROM Code (Non Exhaustive) (continued)

Manufacturer	Memories
Micron	N25Q128A N25Q128A13ESF N25Q256A13ESF N25Q512A13 MT25QL01G
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 (normal boot only; XIP fails) S25FL164 S25FL512
Winbond	W25M512

Note: For an updated list of memories, refer to the "Booting from External Non-Volatile Memory (NVM) on SAM9X7 Series MPUs" (AN4971) application note, available on www.microchip.com.

17.1.5. SAM-BA Monitor

This part of the ROM code is executed when no valid bootstrap is found in an external NVM during the boot sequence, and if the Monitor is not disabled in the Boot Configuration Packet.

The SAM-BA Monitor enables a serial communication link over either the USB Device peripheral or the ROM code console interface. It offers a set of simple commands to communicate with the chip using only a serial link controlled by an external device.

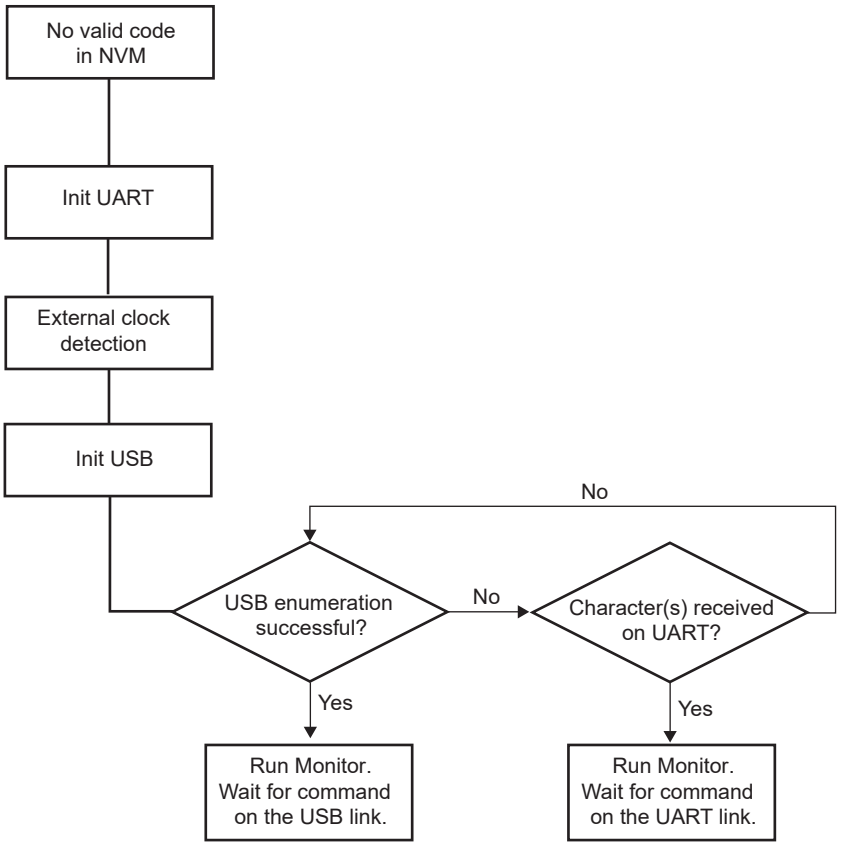
During the whole boot sequence, the Main RC oscillator is used as the main clock of the system. However, to use the USB Device interface, an accurate external clock must be supplied. Thus, before running the SAM-BA Monitor, the ROM code checks whether such an external clock is available. If a valid external clock or crystal is found, the UPLL is configured to allow communication on the USB Device link; otherwise, the USB Device interface is not activated and only the ROM code console is used for the SAM-BA Monitor.

SAM-BA Monitor steps:

- Initialize the ROM code console interface
- Initialize the USB Device interface (if a valid external clock is available)
- Enter a loop waiting for
  - the USB device enumeration by an external host, or
  - a character reception on the ROM code console interface

Once the communication interface is identified, the application runs in an infinite loop waiting for different commands as listed in Table 17.5.

Figure 17.8. SAM-BA Monitor



17.1.5.1.Command List

Table 17.5. SAM-BA Monitor Commands

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,#
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,NbOfBytes#	**S**200000,1234#
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 the SAM-BA Monitor to send/receive data in binary format.
  - Terminal mode configures the SAM-BA 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**): The host will send a buffer of data to be stored at a specified address in the device
  - Address: address in hexadecimal
  - NbOfBytes: number of bytes (in hexadecimal format) to be received from the host
  - Output: '>'
    Note: A timeout on this command is reached when the prompt '>' appears before the end of the command execution.
- Receive a file (**R**): The host requests to read a buffer of data from a specified address in the device.
  - Address: address in hexadecimal
  - NbOfBytes: number of bytes (in hexadecimal format) to be sent to the host
  - 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 manage the link register and does not return, the prompt is not displayed.
- Get ROM code version (**V**): Returns the ROM code version.
  - Output: version, date and time of ROM code followed by '>'

17.1.5.2.ROM Code Console Interface

The ROM code console interface (DBGU or UART) can be configured in the Boot Configuration Packet.

At the beginning of ROM code execution, the `RomBOOT` character string is sent on this same console interface.

Communication settings:

- 115200 bauds
- 8 bits of data
- No parity
- 1 stop bit

17.1.5.2.1.Xmodem Protocol

The "Send File" and "Receive File" commands use the Xmodem protocol. Any terminal using this protocol can be used to send data buffers to this device. The size of the binary file to send depends on the SRAM size embedded in the product. In all cases, the size of the binary file must be lower than 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 guarantee 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:

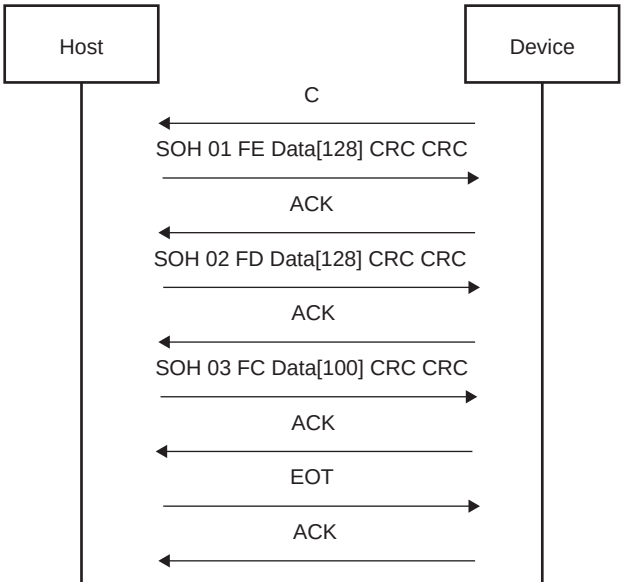
<SOH><blk #><255-blk #><--128 data bytes--><checksum> in which:

- <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 17.9. Xmodem transfer example



17.1.5.3.USB Device Interface

17.1.5.3.1.Supported External Crystal or External Clocks

The SAM-BA Monitor requires an accurate external clock or crystal to use a USB connection. In other cases, only a UART connection is available for the SAM-BA Monitor feature.

17.1.5.3.2.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 supported by Linux, macOS®, and Microsoft Windows®. 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.

17.1.5.3.3.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 17.6. 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

Table 17.6. Handled Standard Requests (continued)

Request	Definition
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 17.7. 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.

17.1.5.3.4.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.

SAM-BA 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.

17.2. Secure Boot Strategy

17.2.1. Description

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 and the AES-CMAC algorithm.
  - Boot file decryption is performed using the customer private CBC key stored in the OTP memory and 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 CBC key stored in the OTP memory and the AES-CBC algorithm.

Encryption is supported by an AES (using HW acceleration); in addition, either symmetric (AES-CMAC) or asymmetric (RSA-based using an X.509 certificate chain) authentication is supported.

17.2.2. Secure Boot Configuration

In addition to the Boot Configuration Packet content, the secure part of the boot sequence flow is controlled with data stored in the Secure Boot Configuration Packet.

The Secure Boot Configuration, stored in the Secure Boot Configuration Packet in the OTP user area, contains all the information required to boot in Secure mode:

- Secure Boot mode enable
- Authentication mode (AES-CMAC or RSA)
- Address of the regular packet containing the customer key wrapped with the PUF key

See Secure Boot Configuration Packet for a detailed description of the fields in the Secure Boot Configuration Packet.

During the prototyping phase, the OTPC Emulation mode can be used to test several configurations and the Secure Boot Mode without burning real OTP bits in the OTP matrix. See Boot Configuration Packet for more details.

17.2.3. Secure Boot Configuration Packet

Table 17.8. Secure Boot Configuration Packet Content

Offset	Name	Description
0x00	SEC_MODE_EN	Secure Boot mode enabled
0x04	AUTH_MODE	Authentication mode selection
0x08	DNU	Do Not Use (keep 0x0 value)
0x0C	KEY_WRITTEN	Key written
0x10	IV	AES Initialization Vector packet address
0x14	RSA_HASH	RSA root certificate public key hash packet address
0x18	CBC_KEY	AES CBC key packet address
0x1C	CMAC_KEY	AES CMAC key packet address

17.2.3.1.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.

Different from 0: Secure Boot mode is enabled.

17.2.3.2. 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.

17.2.3.3.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.

17.2.3.4.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

17.2.3.5.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

17.2.3.6.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

17.2.3.7.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

17.2.4. Valid Code Detection in Secure Boot Mode

The valid code detection in Secure Boot mode is similar to the one in Standard Boot mode. However, additional checks and operations are performed.

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 Vectors 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 size and the certificate chain size must be lower than the maximum bootstrap size. If these first validations fail, the ROM code restores the memory interface PIO and its settings to their reset values and then tries to boot on the next NVM in the boot sequence.

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 in the boot sequence.

If the boot file verification passes, the ROM code deciphers the boot file stored in the internal SRAM0.

Since the NVM interface is no longer needed, the ROM code restores the memory interface PIO and its settings to their reset values.

Finally, the ROM code locks access to the ROM area, enables JTAG and then branches to the beginning of the internal SRAM0 to execute the deciphered bootstrap.

17.2.4.1.6th 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

17.2.4.2.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 `boot.cip` file must contain the ciphered bootstrap, its signature and the chain of X.509 certificates, in this order.

17.2.5. Encryption, Decryption and Authentication

All customer secret keys are stored securely (wrapped using the PUF feature) in a regular packet. 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.

17.2.6. Secure Monitor

When no bootable program is found in any external NVM memory during the boot sequence and Secure Boot mode is enabled, the ROM code executes the Secure Monitor.

The set of commands implemented in the Secure Monitor enables the user to send commands to configure the Secure Boot mode, to write the customer key and the RSA root certificate public key hash, and to execute secure applets.

17.2.6.1.Generic Commands

The command format is composed of an 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).

17.2.6.2.Command Format

The commands are formatted as follows:

____,_____,_____,_____,__#

```
opcode addr 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 format 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 17.9. Read Version Command

Command	Description
RVER	Read ROM code version

Table 17.10. Key Management Commands

Command	Description
WCKY	Send customer keys and initialization vector (640 bits) for AES-CMAC
WCKP	Send customer key and RSA hash of root certificate public key for AES-RSA mode

Table 17.11. Applet-Related Commands

Command	Description
SAPT	Send applet
SMBX	Send applet mailbox content
RMBX	Read applet mailbox content
EAPP	Execute applet
SFIL	Send file to an already loaded applet
RFIL	Read file from the buffer filled by an applet after its execution

17.2.6.3. Monitor Answer

The Secure SAM-BA Monitor answers any command with an acknowledgment 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 17.12. Command List

Command	Full Name	Description
CACK	SEND_CMD_ACK	Returns the status of simple commands
ASTA	SEND_APPLET_STATUS	Returns the status after execution of an applet
SVER	SEND_ROM_VERSION	Returns the ROM code version

Table 17.13. 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

Table 17.13. Error Code (continued)

Error Name	Hex Value	Description
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 not written in OTP
secCmdCustKeyAlreadyWritten	0xFFFFFFF6	Customer key already written in OTP
secCmdCmacErr	0xFFFFFFF5	Message CMAC error
secCmdDecryptErr	0xFFFFFFF4	Error during message decryption
secCmdRsaHashAlreadyWritten	0xFFFFFFEE	RSA hash already written in OTP

17.2.6.4.Command Description

17.2.6.4.1.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 characters should be received.

(PC to device) >> `RVER,0,0,0,00#`

(Device to PC) << `SVER,00000000,00000030#v1.0.p1.Feb 23 2021.18:35:56...`

17.2.6.4.2.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 acknowledgment 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#`

17.2.6.4.3.Running a Secure Applet

Secure applets are small programs allowed by the ROM code to run in the internal SRAM of the device. They are ciphered and signed, and only provided by Microchip as part of the SAM-BA tool. They give the possibility to perform more operations than the one provided by the simple Secure Monitor commands.

Several applets are available. For example, external memories can be programmed using memory-programming algorithm applets, the Boot Configuration and Secure Boot Configuration Packets can be programmed using dedicated applet, etc.

The first step 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, the host computer requests sending an applet of size 0x9870, and this is acknowledged by the Secure Monitor. Then the host computer sends the applet ciphered binary file (applet_binary.cip), and after checking the signature and deciphering the applet in SRAM, the Secure Monitor sends the acknowledgment message and error code (CACK and error code 0x0: successful).

Once the applet is in the SRAM, before executing its code, its mailbox must be filled. The mailbox is a 32-word buffer used to exchange commands and arguments with the applet.

The number of commands and arguments in the mailbox can differ from one applet to another, depending on each applet purpose.

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 can be run with the Execute Applet command:

(PC to device) >> EAPP,0,0,0,00#

(Device to PC) << ASTA,00000000,00000000#

The Secure Monitor replies with an acknowledgment and a status of the applet execution (0x0: successful).

During this step, the ROM code retrieves the information that will be used when receiving the next Write File command:

- Status of the applet execution
- If applicable, address and size of a data buffer for data exchange between the host computer and the applet (for applets intended to program external Flash memories, for example)

A Send File command can then 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#

17.2.7. Secure Boot Mode Recommendations

This section provides recommendations to ensure a good security level when configuring and using the Secure Boot mode, and some bootstrap development information.

17.2.7.1. 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 an RSA signature is used).
5. Program the ciphered bootstrap.
6. Program the other application files.

7. **Disable invalidation of the Boot and Secure Boot Configuration Packets** by writing the corresponding bits in the User Hardware Configuration Packet (see OTP Memory Controller (OTPC)).
8. **Lock the Boot Configuration Packet and the Secure Boot Configuration Packet.**
9. **Disable the Secure SAM-BA Monitor** to avoid any further access.

Note: Keeping the Secure Monitor enabled in order to update the bootstrap in the field or in house is not recommended.

17.2.7.2.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 the number of updates.

17.2.7.2.1.Bootstrap Ciphering

Before update, the bootstrap must be ciphered in accordance with the Secure Boot mode selected.

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, they must be securely stored with restricted and controlled access.

17.2.7.2.2.Bootstrap Update in the Field

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 so, the program in charge of 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.

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 access to the external boot memory.
5. Program the bootstrap using the SAM-BA tool.

17.2.7.3. Monitor Disabling by the Bootstrap

It is highly recommended to disable the Secure Monitor feature for systems in production. To do so, the MON_DIS word must be filled and the Boot Configuration Packet must be locked at the production stage.

If that has not been done, writing the “MON_DIS” word can be performed by the bootstrap or by the customer application, but only if the Boot Configuration Packet has not been locked.

To write the “MON_DIS” word and lock the packet, 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

17.3. Software Considerations

17.3.1. Bootstrap Size

The maximum size allowed for the bootstrap for this chip is 0x8000 bytes (32 Kbytes).

17.3.2. PUF Activation Code Packet Copy in SRAM

When the ROM code launches a bootstrap or executes the SAM-BA Monitor, the PUF Activation Code is made available for the user application in SRAM0 at address 0x0030FBE8. The total size of this data is 0x3E8 bytes.

Structure:

- First 32 bits at address 0x0030FBE8 are 0x43414B51: “QKAC” in ASCII, "tag" used to identify the beginning of the buffer
- 0x0030FBEC to 0x0030FFCD: 0x3E4 bytes of actual activation code data

17.3.3. Boot Interface

Before jumping to the bootstrap entry address, the ROM code provides some information to the bootstrap in CPU registers:

Register	Content
R4	Boot memory

17.4. Hardware Considerations

The table below provides clock frequencies configured by the ROM code during boot.

Table 17.14. Clock Frequencies During External Memory Boot Sequence

Clock	Frequency
PLLA	1200 MHz
CPU_CLK	600 MHz
MCK	200 MHz
SDMMC (init/operational)	400 kHz/25 MHz
QSPI	50 MHz

The default UART used for ROM code console and SAM-BA Monitor UART link is given below:

Index Value in CONSOLE_IOSET[5:0] Bit	Interface

0	DBGU

The default can be changed by writing another index in the Boot Configuration Packet (refer to CONSOLE_PIN).

The NVM drivers use several PIOs in Peripheral mode to communicate with external Flash 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 High 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 17.15. PIO Driven During Boot Program Execution

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up enabled
SD Card/e.MMC	SDMMC_0	1	SDMMC0_DAT0	PIO_PA0A	Y
			SDMMC0_CMD	PIO_PA1A	Y
			SDMMC0_CK	PIO_PA2A	–
			SDMMC0_DAT1	PIO_PA3A	Y
			SDMMC0_DAT2	PIO_PA4A	Y
			SDMMC0_DAT3	PIO_PA5A	Y
	SDMMC_1	1	SDMMC1_DAT0	PIO_PA9B	Y
			SDMMC1_DAT1	PIO_PA6B	Y
			SDMMC1_DAT2	PIO_PA7B	Y
			SDMMC1_DAT3	PIO_PA8B	Y
			SDMMC1_CMD	PIO_PA10B	Y
			SDMMC1_CK	PIO_PA11B	–
NAND Flash	HSMC	1	NANDOE	PIO_PD0A	–
			NANDWE	PIO_PD1A	–
			NANDALE	PIO_PD2A	–
			NANDCLE	PIO_PD3A	–
			NANDCS	PIO_PD4A	–
			NAND WAIT	PIO_PD14A	–
			D0-D7	PIO_PD6A-PIO_PD13A	Y

Table 17.15. PIO Driven During Boot Program Execution (continued)

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up enabled
SPI Flash	FLEXCOM0	1	MOSI	PIO_PA31A	Y
			MISO	PIO_PA30A	–
			NPCS0	PIO_PA3A	–
			SPCK	PIO_PA8A	–
		2	MOSI	PIO_PA0A	Y
			MISO	PIO_PA1A	–
			NPCS1	PIO_PA2A	–
			SPCK	PIO_PA4A	–
	FLEXCOM1	1	MOSI	PIO_PA5A	Y
			MISO	PIO_PA6A	–
			NPCS0	PIO_PC28C	–
			SPCK	PIO_PC29C	–
		2	MOSI	PIO_PA5A	Y
			MISO	PIO_PA6A	–
			NPCS1	PIO_PC27C	–
			SPCK	PIO_PC29C	–
	FLEXCOM2	1	MOSI	PIO_PA7A	Y
			MISO	PIO_PA8A	–
			SPCK	PIO_PB1B	–
			NPCS0	PIO_PB2B	–
		2	MOSI	PIO_PA7A	Y
			MISO	PIO_PA8A	–
			SPCK	PIO_PB2B	–
			NPCS1	PIO_PB0B	–
	FLEXCOM3	1	MOSI	PIO_PC22B	Y
			MISO	PIO_PC23B	–
			NPCS0	PIO_PC25B	–
			SPCK	PIO_PC26B	–
		2	MOSI	PIO_PC22B	Y
			MISO	PIO_PC23B	–
			NPCS1	PIO_PC24B	–
			SPCK	PIO_PC26B	–

Table 17.15. PIO Driven During Boot Program Execution (continued)

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up enabled
SPI Flash	FLEXCOM4	1	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS0	PIO_PA14A	–
		2	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS1	PIO_PA0C	–
		3	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS1	PIO_PA7B	–
		4	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS2	PIO_PA1B	–
		5	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS2	PIO_PA8C	–
		6	MISO	PIO_PA11A	–
			MOSI	PIO_PA12A	Y
			SPCK	PIO_PA13A	–
			NPCS3	PIO_PB3B	–
SPI Flash	FLEXCOM5	1	NPCS0	PIO_PA8B	–
			MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	Y
			SPCK	PIO_PA23B	–
		2	NPCS1	PIO_PA0B	–
			MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	Y
			SPCK	PIO_PA23B	–
		3	MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	Y
			SPCK	PIO_PA23B	–
			NPCS1	PIO_PA7C	–
		4	MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	Y
			SPCK	PIO_PA23B	–
			NPCS2	PIO_PA31B	–
		5	MISO	PIO_PA21B	–
			MOSI	PIO_PA22B	Y
			SPCK	PIO_PA23B	–
			NPCS3	PIO_PA30B	–
QSPI NOR Flash	QSPI0	1	QSCK	PIO_PB19A	–
			QCS	PIO_PB20A	–
			QIO0	PIO_PB21A	X
			QIO1	PIO_PB22A	X
			QIO2	PIO_PB23A	X
			QIO3	PIO_PB24A	X

Table 17.15. PIO Driven During Boot Program Execution (continued)

NVM Bootloader	Peripheral	IO Set	Signal	PIO Line	Pull-up enabled
ROM code console and SAM-BA Monitor	DBGU	1	DTXD	PIO_PA27A	–
			DRXD	PIO_PA26A	X
	FLEXCOM0	1	DTXD	PIO_PA30A	–
			DRXD	PIO_PA31A	X
	FLEXCOM1	1	DTXD	PIO_PA28A	–
			DRXD	PIO_PA29A	X
	FLEXCOM2	1	DTXD	PIO_PA13A	–
			DRXD	PIO_PA14A	X
	FLEXCOM3	1	DTXD	PIO_PC22B	–
			DRXD	PIO_PC23B	X
	FLEXCOM4	1	DTXD	PIO_PA10A	–
			DRXD	PIO_PA9A	X
	FLEXCOM5	1	DTXD	PIO_PA16B	–
			DRXD	PIO_PA15B	X
	FLEXCOM6	1	DTXD	PIO_PA14A	–
			DRXD	PIO_PA15A	X
	FLEXCOM7	1	DTXD	PIO_PC0C	–
			DRXD	PIO_PC1C	X
	FLEXCOM8	1	DTXD	PIO_PB4B	–
			DRXD	PIO_PB5B	X
	FLEXCOM9	1	DTXD	PIO_PC8C	–
			DRXD	PIO_PC9C	X
	FLEXCOM10	1	DTXD	PIO_PC16C	–
			DRXD	PIO_PC17C	X
	FLEXCOM11	1	DTXD	PIO_PB15C	–
			DRXD	PIO_PB16C	X
	FLEXCOM12	1	DTXD	PIO_PB17C	–
			DRXD	PIO_PB18C	X

Memories

18. Overview

18.1. Embedded Memories

18.1.1. Internal SRAM

SAM9X7 Series devices embed 68 Kbytes of high-speed SRAM. SRAM0 (64 KB) is always accessible at address 0x0030 0000. After remap, SRAM0 is also available at address 0x0000 0000. A 4-Kbyte SRAM memory, SRAM1, is used for OTP emulation and is always accessible at address 0x0040 0000.

18.1.2. Internal ROM

The ROM contains a bootloader program mapped at address 0 after reset and the BCH (Bose, Chaudhuri and Hocquenghem) code table mapped at address 0x0010_0000 for NAND Flash ECC correction.

18.1.3. Boot Strategies

Refer to Boot Strategies.

18.2. External Memory

SAM9X7 Series devices offer connections to a wide range of external memories or to parallel peripherals.

18.2.1. External Bus Interface

The External Bus Interface (EBI) is an interface that features:

- Three external memory controllers:
  - a 16-bit Static Memory Controller (SMC)
  - a NAND Flash Controller (NFC)
  - a Multi-Port DDR-SDRAM Controller (MPDDRC) supporting 16-bit DDR3(L)/DDR2
- 8- or 16-bit data bus
- Up to 23-bit address bus
- Up to three chip selects with configurable assignment:
  - Static Memory Controller on NCS0, NCS1, NCS2
  - MPDDRC or Static Memory Controller on NCS1
  - Optional NAND Flash support on NCS2

The drive levels are configured in the EBI I/O Drive Configuration register (SFR_CCFG_EBICSA.EBI_DRIVE) in Special Function Registers (SFR). At reset, the selected drive is low. The user must make sure to program the correct drive. Refer to Electrical Characteristics.

18.2.2. Supported Memories on MPDDRC Interface

The MPDDRC supports the following memories:

- DDR-SDRAM with four or eight internal banks (DDR2-SDRAM or DDR3(L)-SDRAM)
- DDR-SDRAM with 16-bit data path for system-oriented word access
- 2K, 4K, 8K, 16K row address memory parts

18.2.3. Supported Memories on Static Memories and NAND Flash Interfaces

The SMC supports:

- Asynchronous SRAM-like memories and parallel peripherals
- 8- or 16-bit data bus

The NFC supports:

- 8-bit NAND Flash (SLC)
- 24-bit Programmable Multi-bit Error Correcting Code (PMECC) with dedicated databus NANDDAT0-NANDDAT7

18.3. Product Dependencies

18.3.1. Clocks

Memory clocks are not controlled by the PMC.

18.3.2. Interrupts

Refer to the table Peripheral Identifiers.

18.3.3. Reset

SMC and NFC are connected to the processor and peripherals reset line.

18.3.4. I/Os

DDR-SDRAM I/Os are high-speed specific I/Os powered by VDDIOM.

18.4. Special Functions in SFR

18.4.1. DDRC Multiport Enabling

The SFR_REMAP_MP_DDR configuration register includes 14 bits, one for each host. The register has the same function as bit DDR_MP_EN in SFR_CCFG_EBICSA, and can be used instead of DDR_MP_EN for more granularity. If DDR_MP_EN is not enabled, each bit in SFR_REMAP_MP_DDR can be used to perform an individual DDR multiport remapping (bit 0 for Host_0, bit 1 for Host_1, etc.).

This register makes port 0 reachable. It is not used when DDR_MP_EN bit is used.

Refer to Bus Matrix (MATRIX) for a description of hosts and clients.

18.4.2. FLEXRAM Clocking

To save power consumption, the SFR_FLEXRAMS_CLKG_DIS register can be used to enable clock gating on both FLEXRAM ports.

18.4.3. Memories Light Sleep Mode

To save power consumption in low-power modes, Light Sleep mode can be enabled in register SFR_LS.

19. External Bus Interface (EBI)

19.1. Description

The External Bus Interface (EBI) is designed to ensure successful data transfers between external devices and the embedded memory controller of the SAM9X7 Series device.

The Static Memory, MPDDR and ECC Controllers are all featured external memory controllers on the EBI. These external memory controllers are capable of handling several types of external memory and peripheral devices, such as SRAM, PROM, EPROM, EEPROM, Flash and DDR3(L)/DDR2 SDRAM.

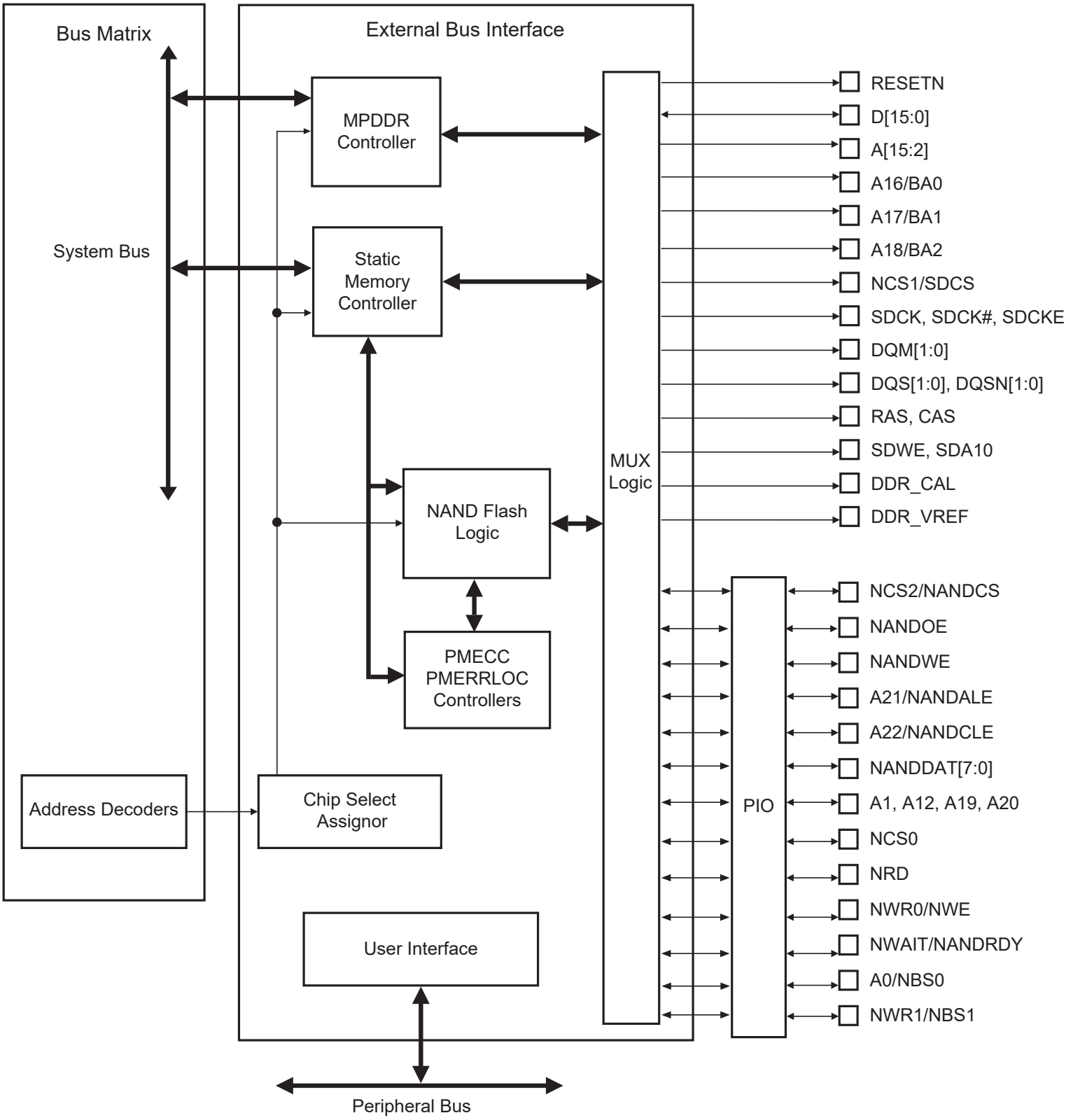
The EBI also supports the NAND Flash protocols via an integrated circuitry that greatly reduces the requirements for external components. Furthermore, the EBI handles data transfers with up to three external devices, each assigned to three address spaces defined by the embedded memory controller. Data transfers are performed through a 16-bit data bus, an address bus of up to 23 bits, up to three chip select lines (NCS[2:0]) and several control pins that are generally multiplexed between the different external memory controllers.

19.2. Embedded Characteristics

- Integrates Three External Memory Controllers:
  - Static memory controller
  - MPDDR controller
  - 8-bit NAND Flash ECC controller
- Up to 23-bit Address Bus (up to 64 Mbytes linear per chip select)
- Up to Three Chip Selects with Configurable Assignment:
  - Static memory controller on NCS0, NCS1, NCS2
  - MPDDR/SDRAM controller on NCS1 (SDCS)
  - NAND Flash support on NCS2 (NANDCS)

19.3. EBI Block Diagram

Figure 19.1. External Bus Interface Organization



19.4. I/O Lines Description

Table 19.1. EBI I/O Lines Description

Name	Function	Type	Active Level
EBI			
EBI_D[15:0]	Data Bus	I/O	–
EBI_A[22:0]	Address Bus	Output	–
EBI_NWAIT	External Wait Signal	Input	Low
SMC			
EBI_NCS[2:0]	Chip Select Lines	Output	Low
EBI_NWR[1:0]	Write Signals	Output	Low
EBI_NRD	Read Signal	Output	Low
EBI_NWE	Write Enable	Output	Low
EBI_NBS[1:0]	Byte Mask Signals	Output	Low
EBI for NAND Flash Support			
EBI_NANDCS	NAND Flash Chip Select Line	Output	Low
EBI_NANDOE	NAND Flash Output Enable	Output	Low
EBI_NANDWE	NAND Flash Write Enable	Output	Low
MPDDR Controllers			
EBI_SDCK, EBI_SDCK#	MPDDR Differential Clock	Output	–
EBI_SDCK	MPDDR Clock	Output	–
EBI_SDCKE	MPDDR Clock Enable	Output	High
EBI_DDRCS	MPDDR Chip Select Line	Output	Low
EBI_BA[2:0]	Bank Select	Output	–
EBI_SDWE	MPDDR Write Enable	Output	Low
EBI_RAS-EBI_CAS	Row and Column Signal	Output	Low
EBI_SDA10	SDRAM Address 10 Line	Output	–

The connection of some signals through the MUX logic is not direct and depends on the memory controller currently in use.

The following table details the connections between the two memory controllers and the EBI pins.

Table 19.2. EBI Pins and Memory Controllers I/O Lines Connections

EBIx Pins	MPDDR I/O Lines	SMC I/O Lines
EBI_NWR1/NBS1	NBS1	NWR1
EBI_A0/NBS0	Not Supported	A0
EBI_A1	Not Supported	A1
EBI_A[11:2]	A[9:0]	A[11:2]
EBI_SDA10	A10	Not Supported
EBI_A12	Not Supported	A12
EBI_A[15:13]	A[13:11]	A[15:13]
EBI_A[22:16]	Not Supported	A[22:16]
EBI_D[15:0]	D[15:0]	D[15:0]

19.5. Application Examples

19.5.1. Hardware Interface

The following table details the connections to be applied between the EBI pins and the external devices for each memory controller.

Table 19.3. EBI Pins and External Static Device Connections

Signals: EBI_	Pins of the Interfaced Device		
	8-bit Static Device	2 x 8-bit Static Devices	16-bit Static Device
Controller	SMC		
D[7:0]	D[7:0]	D[7:0]	D[7:0]
D[15:8]	–	D[15:8]	D[15:8]
A0/NBS0[1]	A0	–	NLB
A1[1]	A1	A0	A0
A[22:2][1]	A[20:2]	A[19:1]	A[19:1]
NCS0[1]	CS	CS	CS
NCS1/DDRCS	CS	CS	CS
NCS2/NANDCS[1]	CS	CS	CS
NRD[1]	OE	OE	OE
NWR0/NWE[1]	WE	WE[2]	WE
NWR1/NBS1[1]	–	WE[2]	NUB

Notes:

1. A0/NBS0, A1, A12, A19 and A20, NCS0, NCS2/NANDCS, NRD, NWR0/NWE, NWR1/NBS1 are multiplexed on PD[13:4].
2. NWR1 enables upper byte writes. NWR0 enables lower byte writes.

Table 19.4. EBI Pins and External Device Connections

Signals: EBI_	Power supply	Pins of the Interfaced Device	
		DDR3(L)/DDR2	NAND Flash
Controller		MPDDRC	NFC
D[7:0]	VDDIOM	D[7:0]	NFD[7:0][1]
D[15:8]	VDDIOM	D[15:8]	–
NANDDAT[7:0]	VDDNF	–	NFD[7:0][1]
A0/NBS0	VDDIOM	–	–
A1	VDDNF	–	–
DQM[1:0]	VDDIOM	DQM[1:0]	–
DQS[1:0], DQSN[1:0]	VDDIOM	DQS[1:0], DQSN[1:0]	–
A[10:2]	VDDIOM	A[8:0]	–
A11	VDDIOM	A9	–
SDA10	VDDIOM	A10	–
A12	VDDNF	–	–
A[14:13]	VDDIOM	A[12:11]	–
A15	VDDIOM	A13	–
A16/BA0	VDDIOM	BA0	–
A17/BA1	VDDIOM	BA1	–
A18/BA2	VDDIOM	BA2	–
A19	VDDNF	–	–
A20	VDDNF	–	–
A21/NANDALE	VDDNF	–	ALE
A22/NANDCLE	VDDNF	–	CLE
NCS0	VDDNF	–	–
NCS1/SDCS	VDDIOM	DDRCS	–

Table 19.4. EBI Pins and External Device Connections (continued)

Signals: EBI_	Power supply	Pins of the Interfaced Device	
		DDR3(L)/DDR2	NAND Flash
Controller		MPDDRC	NFC
NCS2/NANDCS	VDDNF	–	CE
NANDOE	VDDNF	–	OE
NANDWE	VDDNF	–	WE
NRD	VDDNF	–	–
NWR0/NWE	VDDNF	–	–
NWR1/NBS1	VDDNF	–	–
SDCK	VDDIOM	CK	–
SDCK#	VDDIOM	CK#	–
SDCKE	VDDIOM	CKE	–
RAS	VDDIOM	RAS	–
CAS	VDDIOM	CAS	–
SDWE	VDDIOM	WE	–
NWAIT/NANDRDY	VDDNF	–	NANDRDY

Note:

1. The NFD0_ON_D16 switch is used to select NAND Flash path on D[7:0] or NANDDAT[7:0] depending on memory power supplies. This switch is located in the SFR_CCFG_EBICSA register in the Special Function Register.

19.5.2. Product Dependencies

19.5.2.1. I/O Lines

The pins used for interfacing the External Bus Interface may be multiplexed with the PIO lines. The programmer must first program the PIO controller to assign the External Bus Interface pins to their peripheral function. If I/O lines of the External Bus Interface are not used by the application, they can be used for other purposes by the PIO Controller.

19.5.3. Functional Description

The EBI transfers data between the internal AHB Bus (handled by the Bus Matrix) and the external devices (memories, FPGA, etc.). It controls the waveforms and the parameters of the external address, data and control buses and is composed of the following elements:

- Static Memory Controller (SMC)
- MPDDR Controller (MPDDRC)
- Programmable Multibit ECC Controller (PMECC)
- A chip select assignment feature that assigns an AHB address space to the external devices
- A multiplex controller circuit that shares the pins between the different Memory Controllers
- Programmable NAND Flash support logic

19.5.3.1. Bus Multiplexing

The EBI offers a complete set of control signals that share the 16-bit data lines, the address lines of up to 23 bits and the control signals through a multiplex logic operating in function of the memory area requests.

Multiplexing is specifically organized in order to guarantee the maintenance of the address and output control lines at a stable state while no external access is being performed. Multiplexing is also designed to respect the data float times defined in the Memory Controllers.

19.5.3.2. Pull-Up and Pull-Down Control

The SFR_CCFG_EBICSA register in the Special Function Register User Interface enable on-chip pull-up and pull-down resistors on data bus lines not multiplexed with the PIO Controller lines. The pull-down resistors are enabled after reset. The bits, EBIx_DBPUC and EBI_DBPDC, control the pull-up and pull-down resistors on the D[15:0] lines.

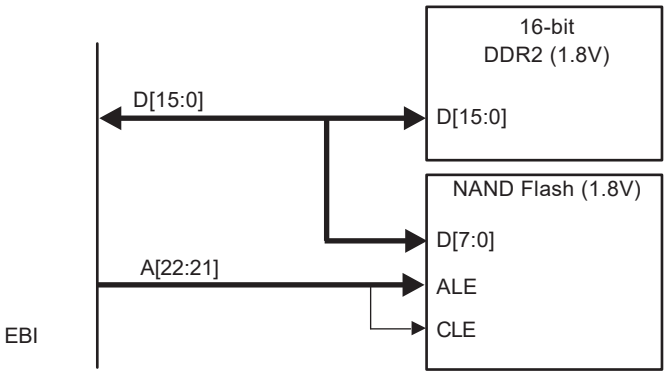
19.5.3.3. Power Supplies

The product embeds a dual power supply for the EBI: VDDNF for NAND Flash signals, and VDDIOM for other signals. This makes it possible to use a 1.8V or 3.3V NAND Flash independently of the SDRAM power supply.

The switch NFD0_ON_D16 is used to select the NAND Flash path on D[7:0] or NANDDAT[7:0] depending on memory power supplies. This switch is located in the SFR_CCFG_EBICSA register (refer to Special Function Registers (SFR)). At reset NFD0_ON_D16 = 0 and the NAND Flash bus is connected to D[7:0].

The following figure illustrates an example of the NAND Flash and the external RAM (DDR2) in the same power supply range (NFD0_ON_D16 = default).

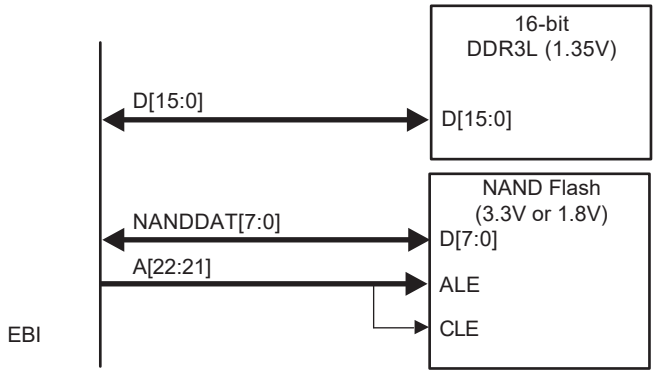
Figure 19.2. NAND Flash and External RAM in Same Power Supply Range (NFD0_ON_D16 = default)



The following figure illustrates an example of the NAND Flash and the external RAM (DDR3L) NOT in the same power supply range (NFD0_ON_D16 = 1).

This can be used if the SMC connects to the NAND Flash only. Using this function with another device on the SMC will lead to an unpredictable behavior of that device. In that case, the default value must be selected.

Figure 19.3. NAND Flash and External RAM Not in Same Power Supply Range (NFD0_ON_D16 = 1)



19.5.3.4. Static Memory Controller

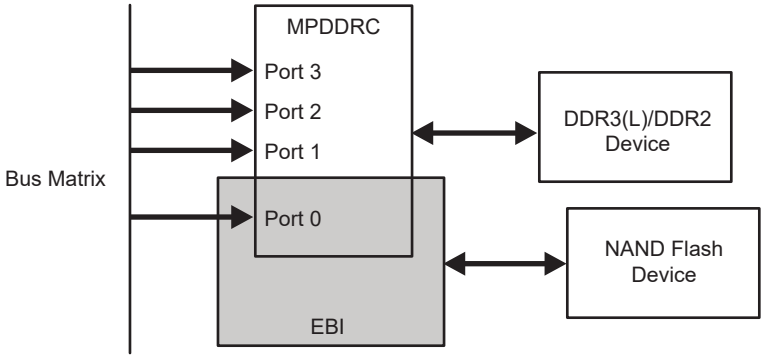
For information, refer to Static Memory Controller (SMC)

19.5.3.5. Multi-Port DDR and SDRAM Controllers

The product embeds a multi-port DDR Controller. This allows to use three additional ports on the MPDDRC to lessen the EBI load from a part of SDRAM accesses. This increases the bandwidth when DDR3(L)/DDR2 and NAND Flash devices are used.

It is controlled by the DDR_MP_EN bit in EBI Chip Select Assignment Register.

Figure 19.4. Multi-Port Enabled MPDDRC (DDR_MP_EN = 1)

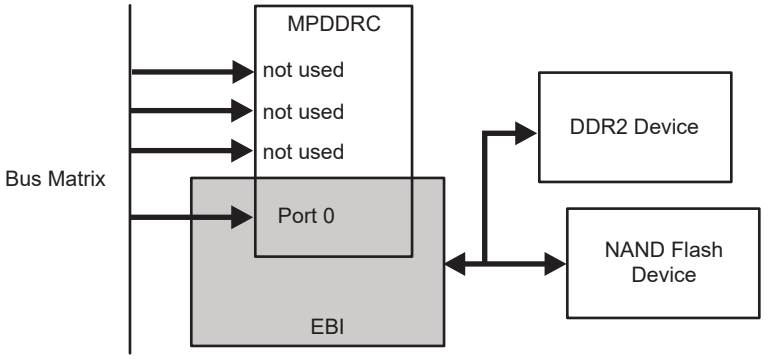


When:

- a NAND Flash memory is connected to NANDDAT[7:0] and
- a DDR2-SDRAM is connected to D[15:0],

the bits SFR_CCFG_EBICSA.DDR_MP_EN and SFR_CCFG_EBICSA.NFD0_ON_D16 must both be set before performing the SDRAM initialization.

Figure 19.5. Multi-Port Disabled MPDDRC (DDR_MP_EN = 0)



19.5.3.6. Programmable Multibit ECC Controller

For information on the PMECC Controller, refer to Programmable Multibit Error Correction Code Controller (PMECC) and Programmable Multibit ECC Error Location Controller (PMERRLOC). Also refer to NAND Flash Boot: PMECC Error Detection and Correction.

19.5.3.7. NAND Flash Support

External Bus Interfaces integrate circuitry that interfaces to NAND Flash devices.

External Bus Interface

The NAND Flash logic is driven by the Static Memory Controller on the NCS2 address space. Programming the EBI_CS2A field in the SFR_CCFG_EBICSA Register in the SFR User Interface to the

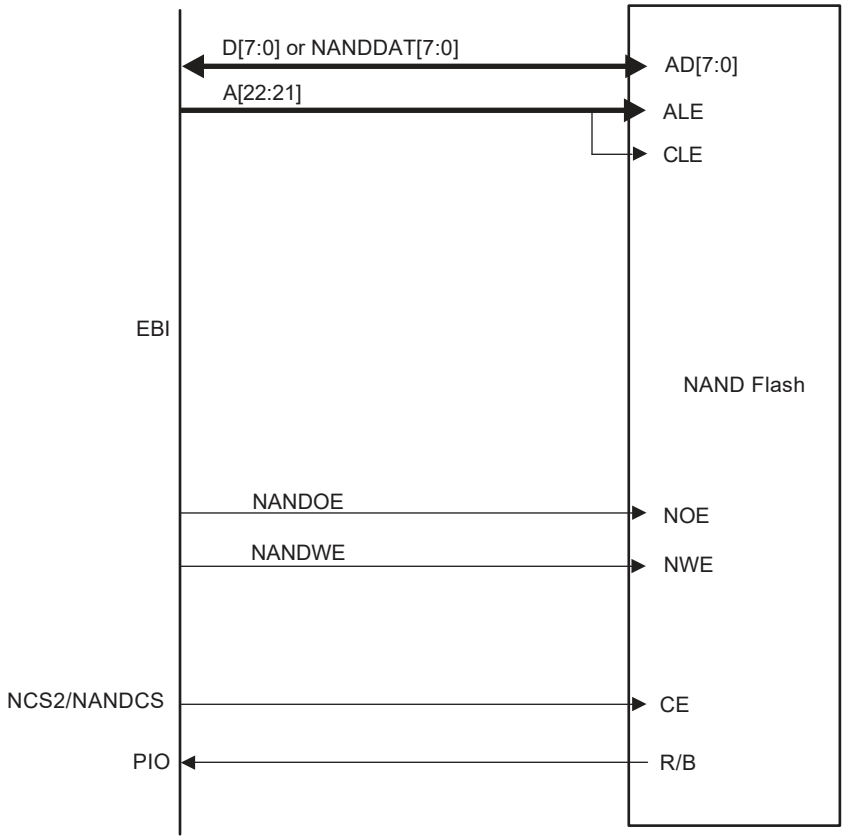
appropriate value enables the NAND Flash logic. For details on this register, refer to Special Function Registers (SFR). Access to an external NAND Flash device is then made by accessing the address space reserved to NCS2 (i.e., between 0x3000 0000 and 0x3FFF FFFF).

The NAND Flash Logic drives the read and write command signals of the SMC on the NANDOE and NANDWE signals when the NCS2 signal is active. NANDOE and NANDWE are invalidated as soon as the transfer address fails to lie in the NCS2 address space. See the figure below for more information. For details on these waveforms, refer to Static Memory Controller (SMC).

NAND Flash Signals

The address latch enable and command latch enable signals on the NAND Flash device are driven by address bits A22 and A21 of the EBI address bus. The command, address or data words on the data bus of the NAND Flash device are distinguished by using their addresses within the NCSx address space. The chip enable (CE) signal of the device and the ready/busy (R/B) signals are connected to PIO lines. The CE signal then remains asserted even when NCSx is not selected, preventing the device from returning to Standby mode.

Figure 19.6. NAND Flash Application Example



Note: The CE signal of the NAND Flash must be connected to PIOD4 (NCS2/NANDCS) if the user's system boots out of NAND Flash.

19.5.4. Implementation Examples

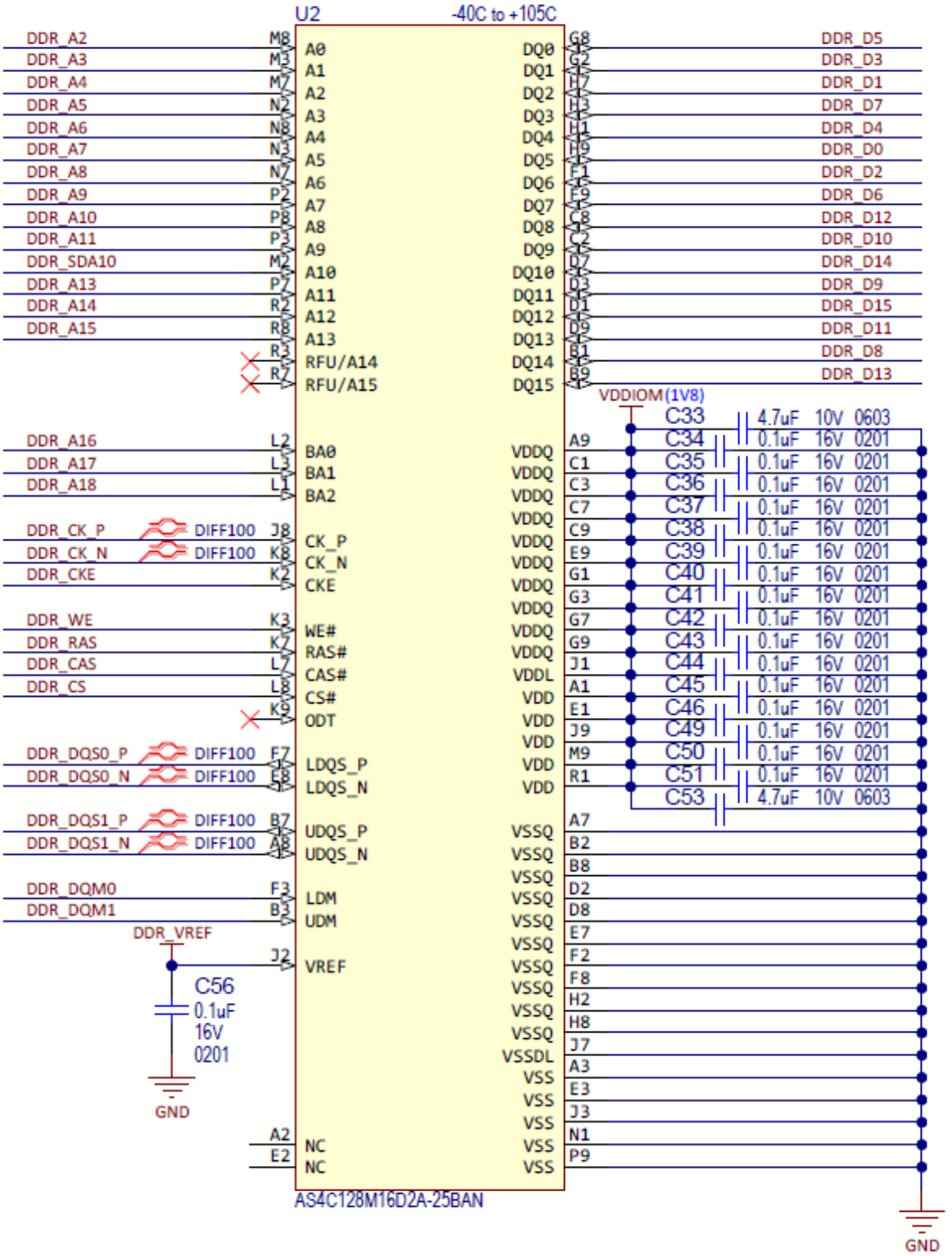
The following hardware configurations are given for illustration only. The user should refer to the memory manufacturer web site to check current device availability.

19.5.4.1.1x16-bit DDR2 on EBI NCS1

19.5.4.1.1.Hardware Configuration

This configuration shows DDR2-SDRAM with a 1.8V power supply. The same configuration can apply to DDR3L-SDRAM with a 1.35V power supply.

Figure 19.7. Configuration Example: 16-bit DDR2 on EBI NCS1



19.5.4.1.2.Software Configuration

- Assign EBI_CS1 to the MPDDRC controller by setting the EBI_CS1A bit in the SFR_CCFG_EBICSA register.
- Initialize the MPDDR Controller depending on the DDR2 device and system bus frequency.

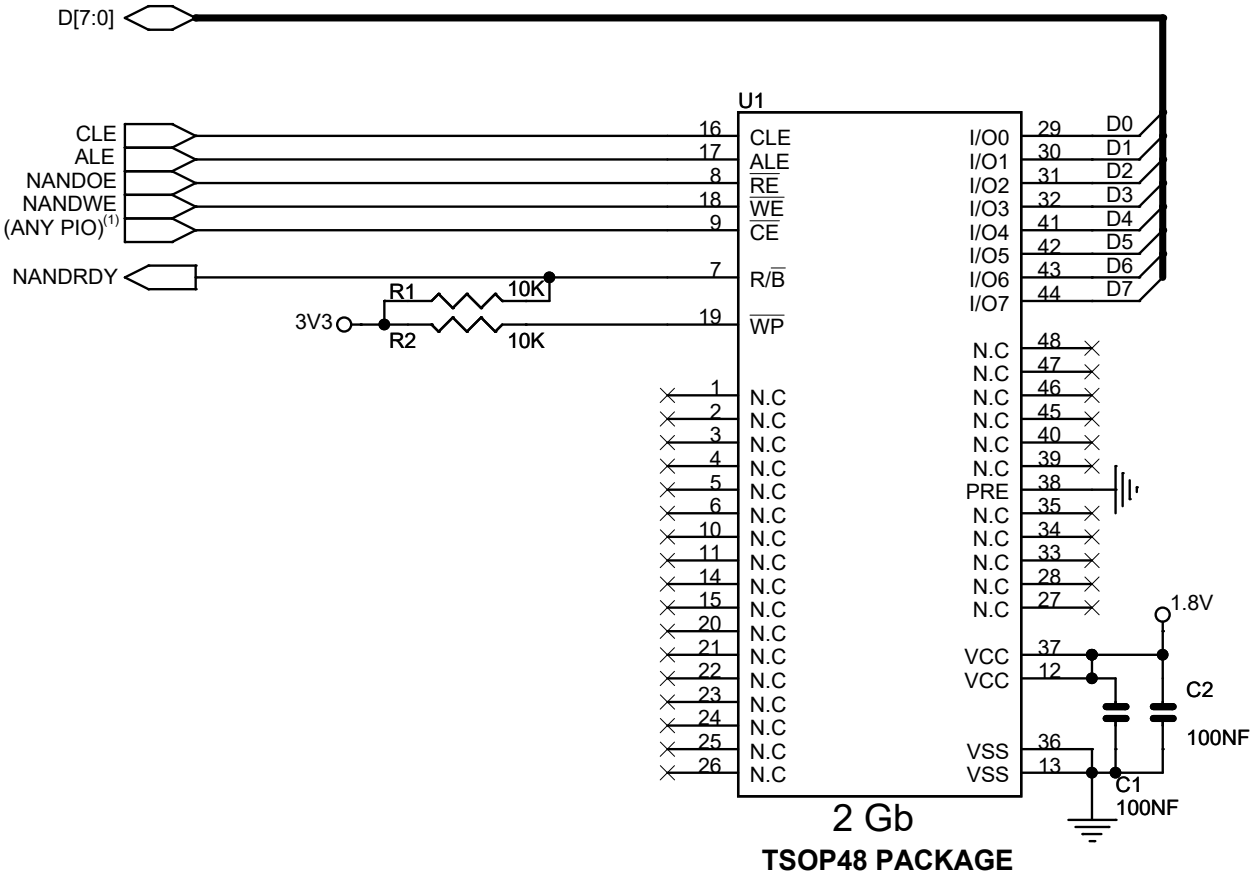
The DDR2 initialization sequence is described in the subsection “DDR2 Device Initialization” of DDR-SDRAM Controller (MPDDRC).

19.5.4.2.Additional 1.8V 8-bit NAND Flash on NCS2 with NFD0_ON_D16 = 0, with DDR2-SDRAM

19.5.4.2.1.Hardware Configuration

In this configuration NAND Flash and DDR2-SDRAM are on the same data bus, with the same power supply. As DDR2-SDRAM is used, NAND Flash must be 1.8V powered. So, VDDIOM and VDDNF must be set to 1.8V.

Figure 19.8. Configuration Example: Additional 1.8V 8-bit NAND Flash on NCS2 with NFD0_ON_D16 = 0, with DDR2-SDRAM



(1) The CE must be connected to NCS2 PIOD4 if the NAND Flash is used by the ROM code.

Note: The CE signal of the NAND Flash must be connected to PIOD4 (NCS2/NANDCS) if the user's system boots out of NAND Flash.

19.5.4.2.2.Software Configuration

The following configuration has to be performed:

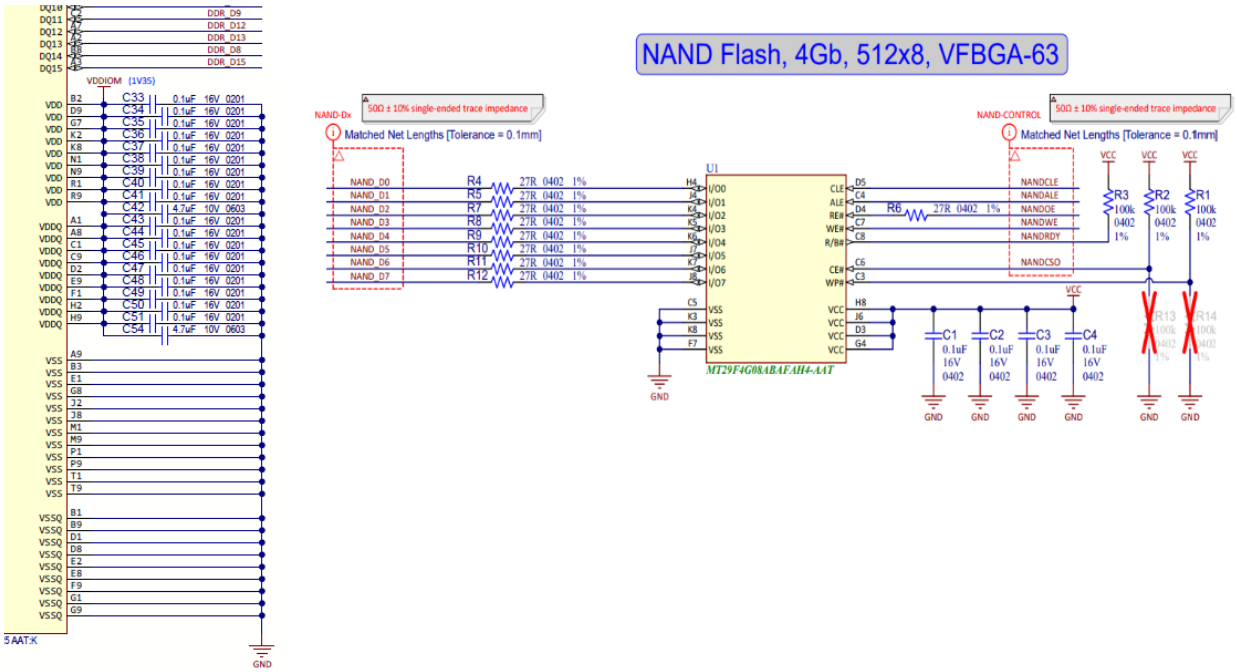
- Set NFD0_ON_D16 = 0 in the SFR_CCFG_EBICSA register.
- Assign the EBI CS2 to the NAND Flash by setting the EBI_CS2A bit in the SFR_CCFG_EBICSA register.
- Reserve A21/A22 for ALE/CLE functions. Address and Command Latches are controlled respectively by setting to 1 the address bits A21 and A22 during accesses.
- Configure PD14 as an input to manage the Ready/Busy signal.
- Configure Static Memory Controller CS2 Setup, Pulse, Cycle and Mode according to NAND Flash timings, data bus width and system bus frequency.

19.5.4.3.16-bit DDR2/3L on NCS1 and 8-bit NAND Flash on NCS2 with NFD0_ON_D16 = 1 (1.8V or 3.3V)

19.5.4.3.1.Hardware Configuration

In this configuration, NAND Flash and DDR2-SDRAM are not on the same data bus and use separate power supplies. DDR2-SDRAM is shown, but the same configuration applies to DDR3L-SDRAM with a 1.35V VDDIOM power supply. NAND Flash is separately powered by VDDNF and can be 1.8V or 3.3V powered.

Figure 19.9. Configuration Example: 16-bit DDR2/3L on NCS1 and 8-bit NAND Flash on NCS2 with NFD0_ON_D16 = 1 (1.8V or 3.3V)



19.5.4.3.2.Software Configuration

The following configuration must be performed:

1. Set NFD0_ON_D16 = 1 in the SFR_CCFG_EBICSA register.
2. Assign the EBI CS2 to the NAND Flash by setting the EBI_CS2A bit in the SFR_CCFG_EBICSA register.
3. Reserve A21 / A22 for ALE / CLE functions. Address and Command Latches are controlled respectively by setting to 1 the address bits A21 and A22 during accesses.
4. Configure PD14 as an input to manage the Ready/Busy signal.

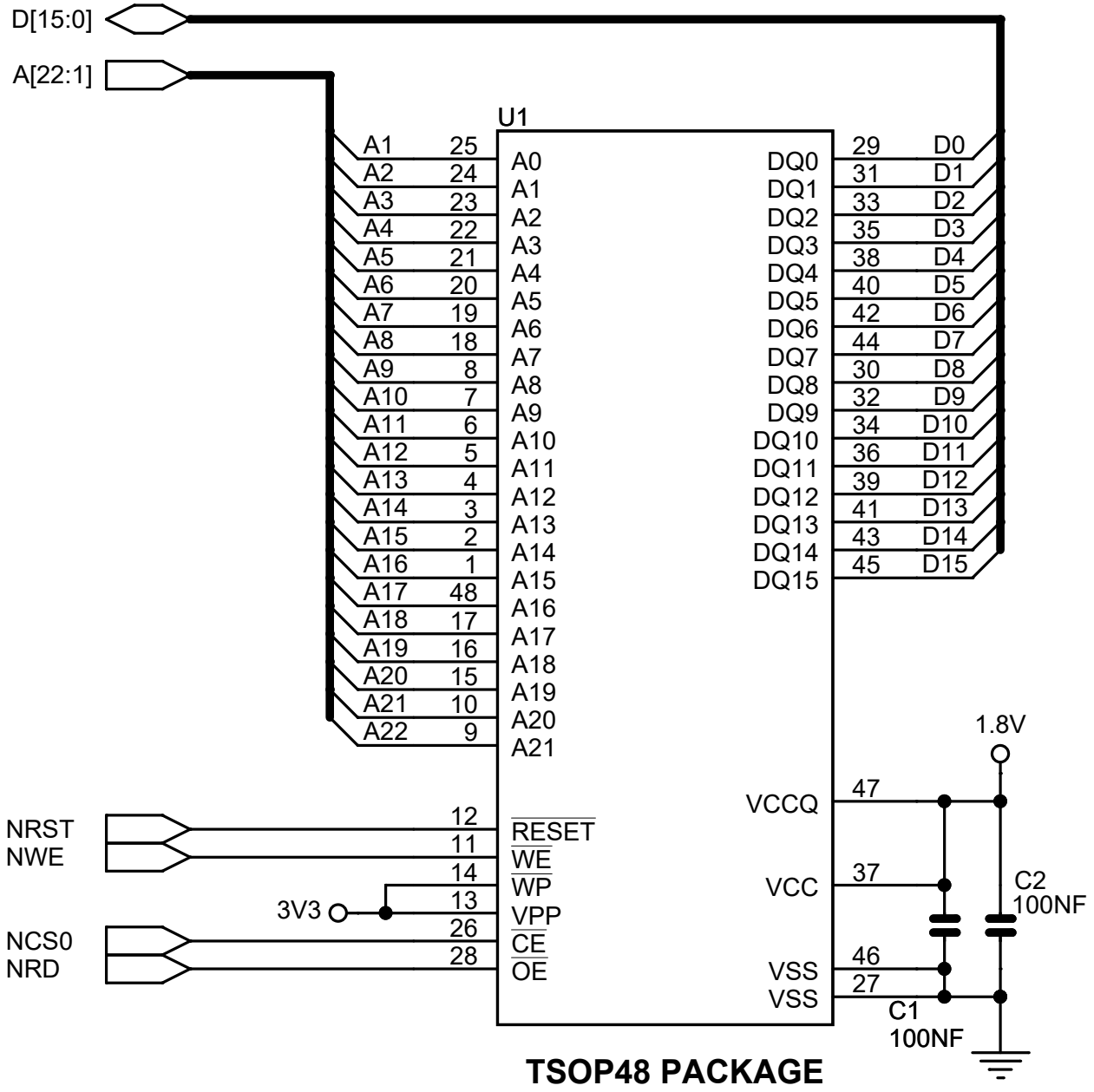
5. Configure Static Memory Controller CS2 Setup, Pulse, Cycle and Mode according to NAND Flash timings, data bus width and system bus frequency.

19.5.4.4.NOR Flash on NCS0

19.5.4.4.1.Hardware Configuration

In this configuration, NOR Flash and DDR2-SDRAM are on the same data bus and use the same power supply. As DDR2-SDRAM is used, NOR Flash must be 1.8V powered. As some control signals are powered by the VDDNF domain, this configuration forces VDDNF to be powered at 1.8V like VDDIOM.

Figure 19.10. Configuration Example: NOR Flash on NCS0



19.5.4.4.2.Software Configuration

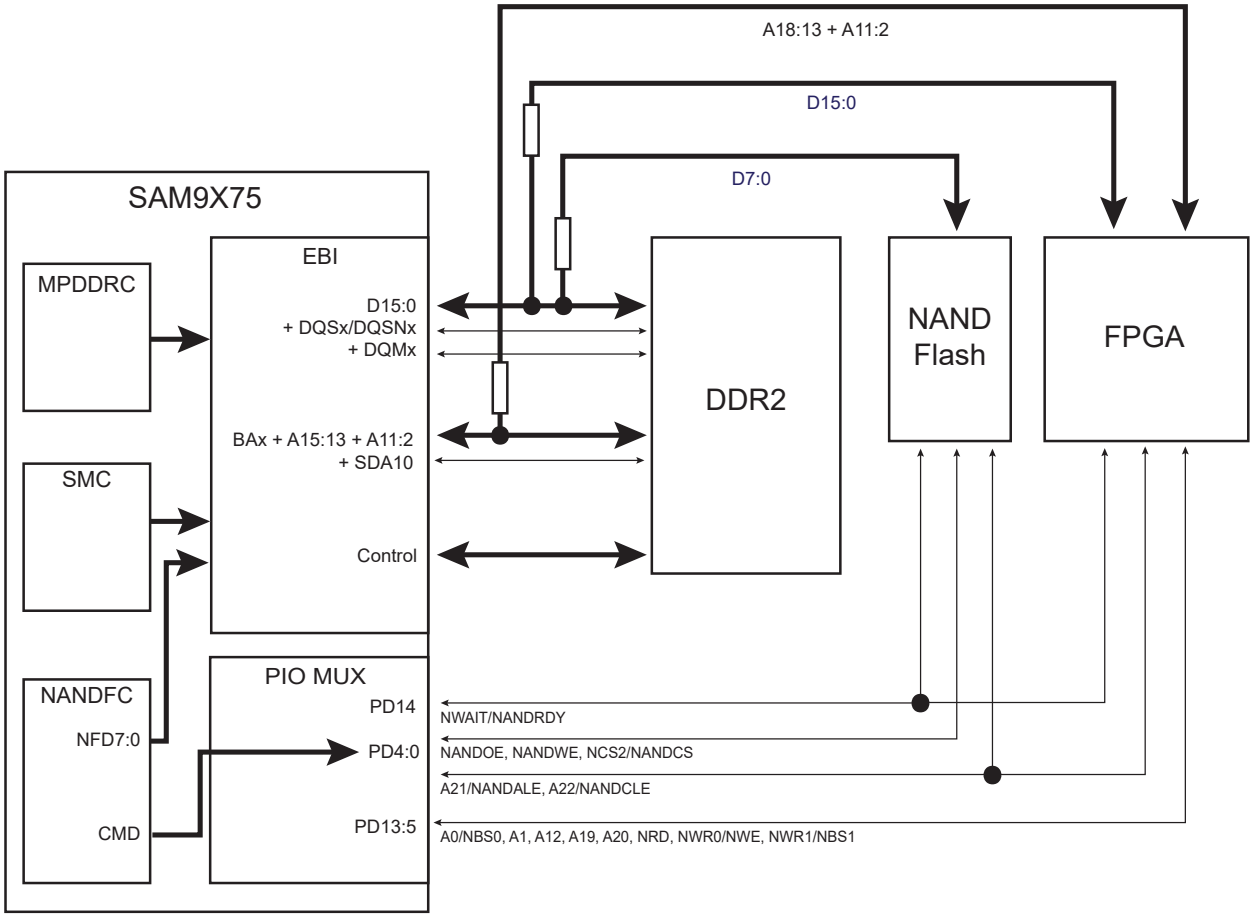
The default configuration for the Static Memory Controller, Byte Select mode, 16-bit data bus, Read/Write controlled by Chip Select, allows boot on 16-bit non-volatile memory at slow clock.

For another configuration, configure the Static Memory Controller CS0 Setup, Pulse, Cycle and Mode depending on Flash timings and system bus frequency.

19.5.4.5. Device (FPGA, Static Memory, etc.) on NCS0, DDR on NCS1 and NAND Flash on NCS2

In this configuration, the device on NCS0, DDR2-SDRAM and NAND Flash are on the same data bus and use the same power supply. As DDR2-SDRAM is used, all components attached to the EBI must be 1.8V powered. NFD0_ON_D16 must be set to 0 in this configuration (NFD0_ON_D16 = 1 is a forbidden configuration with three devices on the EBI). As a few control signals are powered by the VDDNF domain, this configuration forces VDDNF to be powered at 1.8V like VDDIOM. Note this configuration is not supported when using DDR3L-SDRAM.

Figure 19.11. Configuration Example: FPGA on NCS0, DDR on NCS1 and NAND Flash on NCS2



Note: For signal integrity purposes, it is good practice to give priority on the PCB layout to the Processor - DDR2 interface and to isolate this interface from the FPGA and the NAND Flash with serial resistors in the 50 to 200 Ohms range. This configuration implies speed limitation on both the DDR2 interface depending on the layout performance and the device connected to the bus. A signal integrity board simulation must be performed to assess the expected speed operation.

20. Static Memory Controller (SMC)

20.1. Description

The Static Memory Controller (SMC) generates the signals that control the access to the external memory devices or peripheral devices. It has 3 Chip Selects and a 23-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 parametrizable.

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 supports asynchronous burst read in Page mode access for page sizes up to 32 bytes.

20.2. Embedded Characteristics

- 3 Chip Selects Available
- 8-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 Wait Request
- Automatic Switch to Slow Clock Mode
- Asynchronous Read in Page Mode Supported: Page Size Ranges from 4 to 32 Bytes
- Functional Safety - Protection, Monitors and Reports:
  - Register Write Protection
  - Abnormal Software Access and Sequencer Integrity Error Reports

20.3. I/O Lines Description

Table 20.1. I/O Lines Description

Name	Description	Type	Active Level
NCS[2:0]	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[22:2]	Address Bus	Output	–
D[15:0]	Data Bus	I/O	–
NWAIT	External Wait Signal	Input	Low

20.4. Interrupt Source

The SMC has an interrupt line connected to the interrupt line of the external bus interface. The external bus interface (SMC) interrupt line is connected to the interrupt controller. The external bus interface interrupt line can be triggered by SMC.

20.5. Multiplexed Signals

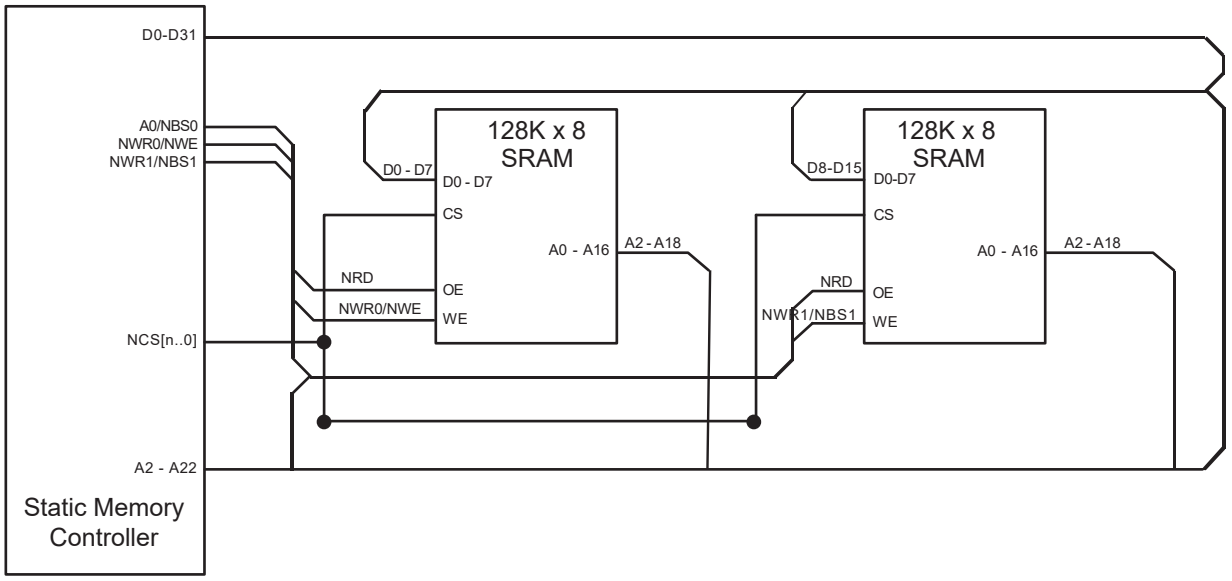
Table 20.2. Static Memory Controller (SMC) Multiplexed Signals

Multiplexed Signals			Related Function
NWR0	NWE	–	Byte-write or byte-select access, see Byte Write or Byte Select Access
A0	NBS0	–	8-bit or 16-bit data bus, see Data Bus Width
NWR1	NBS1	–	Byte-write or byte-select access, see Byte Write or Byte Select Access

20.6. Application Example

20.6.1. Hardware Interface

Figure 20.1. SMC Connections to Static Memory Devices



20.7. Product Dependencies

20.7.1. I/O Lines

The pins used for interfacing the Static Memory Controller may be 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.

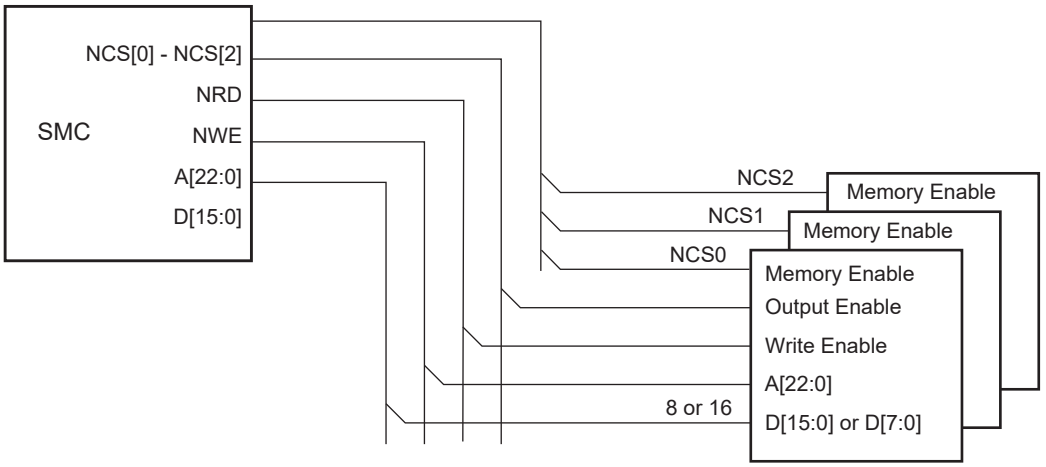
20.8. External Memory Mapping

The SMC provides up to 23 address lines, A[22:0]. This allows each chip select line to address up to 8 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 8 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[22:0] is only significant for an 8-bit memory, A[22:1] is used for a 16-bit memory.

Figure 20.2. Memory Connections for External Devices



20.9. Connection to External Devices

20.9.1. 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 field DBW in the Mode register (SMC_MODE) for the corresponding chip select.

- Figure 20.3 illustrates how to connect a 512K x 8-bit memory on NCS2.
- Figure 20.4 illustrates how to connect a 512K x 16-bit memory on NCS2.

20.9.2. 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 access. This is controlled by the BAT field of the SMC_MODE register for the corresponding chip select.

Figure 20.3. Memory Connection for an 8-bit Data Bus

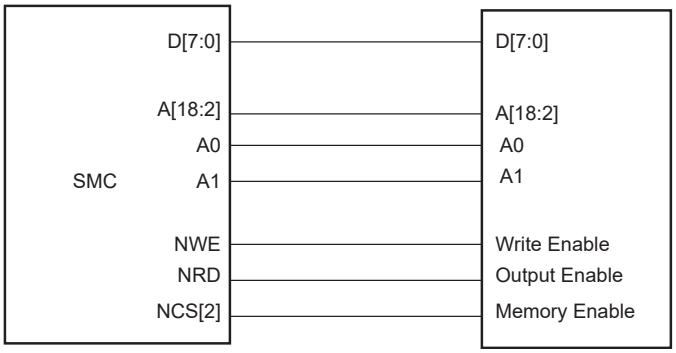
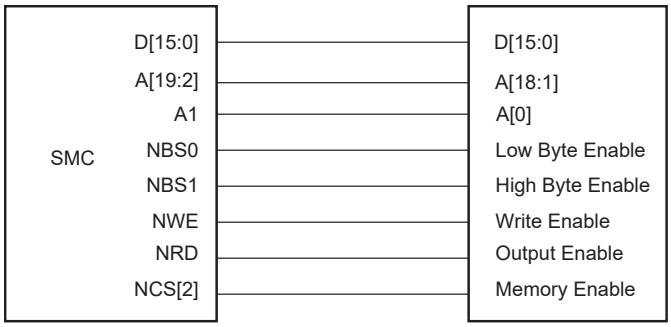


Figure 20.4. Memory Connection for a 16-bit Data Bus



20.9.2.1. Byte Write Access

Byte Write Access supports one byte 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.

Byte Write Access is used to connect 2 x 8-bit devices as a 16-bit memory.

The Byte Write option is illustrated in Figure 20.5.

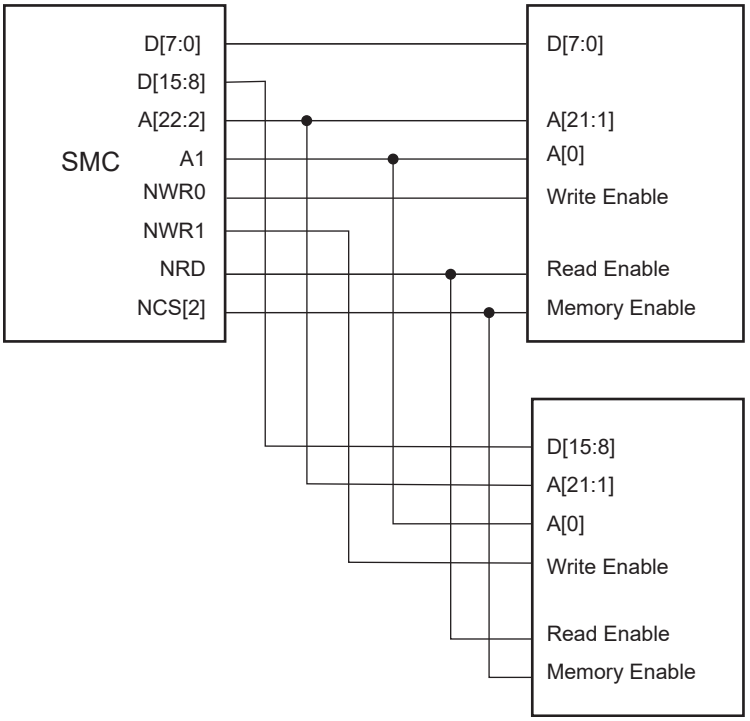
20.9.2.2. Byte Select Access

In this mode, read/write operations can be enabled/disabled at a byte level. One byte-select line per byte of the data bus is provided. One NRD signal 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.

Byte Select Access is used to connect one 16-bit device.

Figure 20.5. Connection of 2 x 8-bit Devices on a 16-bit Bus: Byte Write Option



20.9.2.3.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.

Table 20.3. SMC Multiplexed Signal Translation

Signal Name	16-bit Bus		8-bit Bus
Device Type	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	–

20.10. Standard Read and Write Protocols

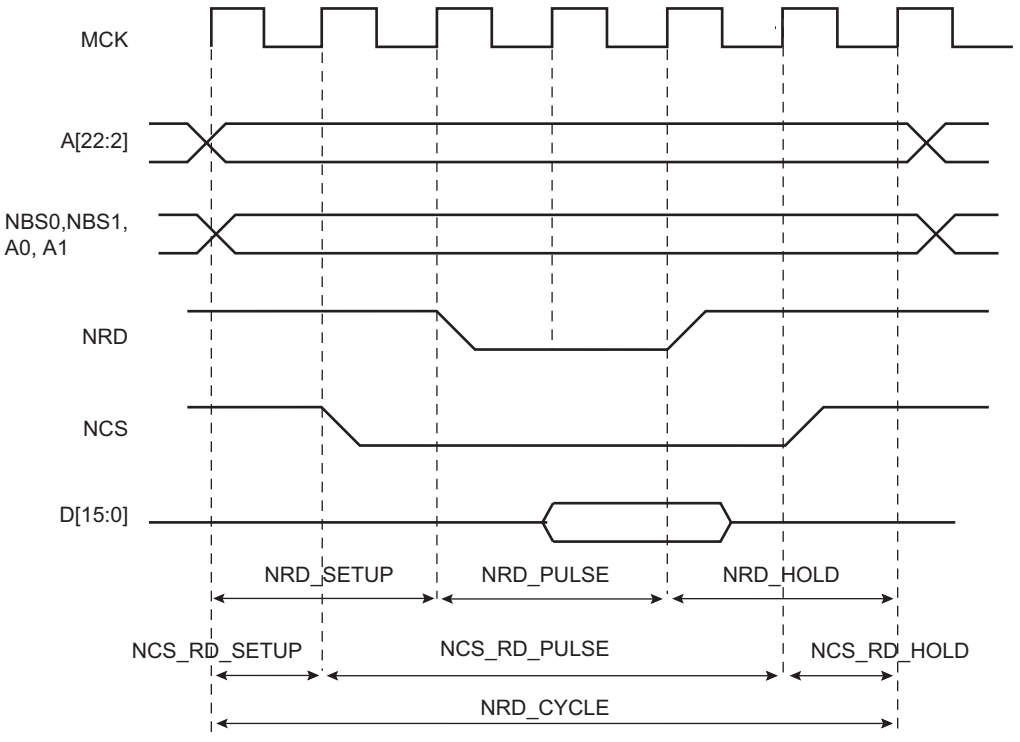
In the following sections, the byte access type is not considered. Byte select lines (NBS0 to NBS1) always have the same timing as the A address bus. NWE represents either the NWE signal in byte select access type or one of the byte write lines (NWR0 to NWR1) in byte write access type. NWR0 to NWR1 have the same timings and protocol as NWE. In the same way, NCS represents one of the NCS[0..2] chip select lines.

20.10.1. Read Waveforms

The following figure shows the read cycle. The read cycle starts with the address setting on the memory address bus:

- {A[22:2], A1, A0} for 8-bit devices
- {A[22:2], A1} for 16-bit devices

Figure 20.6. Standard Read Cycle



20.10.1.1.NRD Waveform

The NRD signal is characterized by a setup timing, a pulse width and a hold timing:

- NRD_SETUP—NRD setup time is defined as the setup of address before the NRD falling edge.
- NRD_PULSE—NRD pulse length is the time between NRD falling edge and NRD rising edge.
- NRD_HOLD—NRD hold time is defined as the hold time of address after the NRD rising edge.

20.10.1.2.NCS Waveform

Similar to the NRD signal, the NCS signal can be divided into a setup time, pulse length and hold time:

- NCS_RD_SETUP—NCS setup time is defined as the setup time of address before the NCS falling edge.
- NCS_RD_PULSE—NCS pulse length is the time between NCS falling edge and NCS rising edge;
- NCS_RD_HOLD—NCS hold time is defined as the hold time of address after the NCS rising edge.

20.10.1.3.Read Cycle

The NRD_CYCLE time is defined as the total duration of the read cycle, that is, from the time where address is set on the address bus to the point where 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 the 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

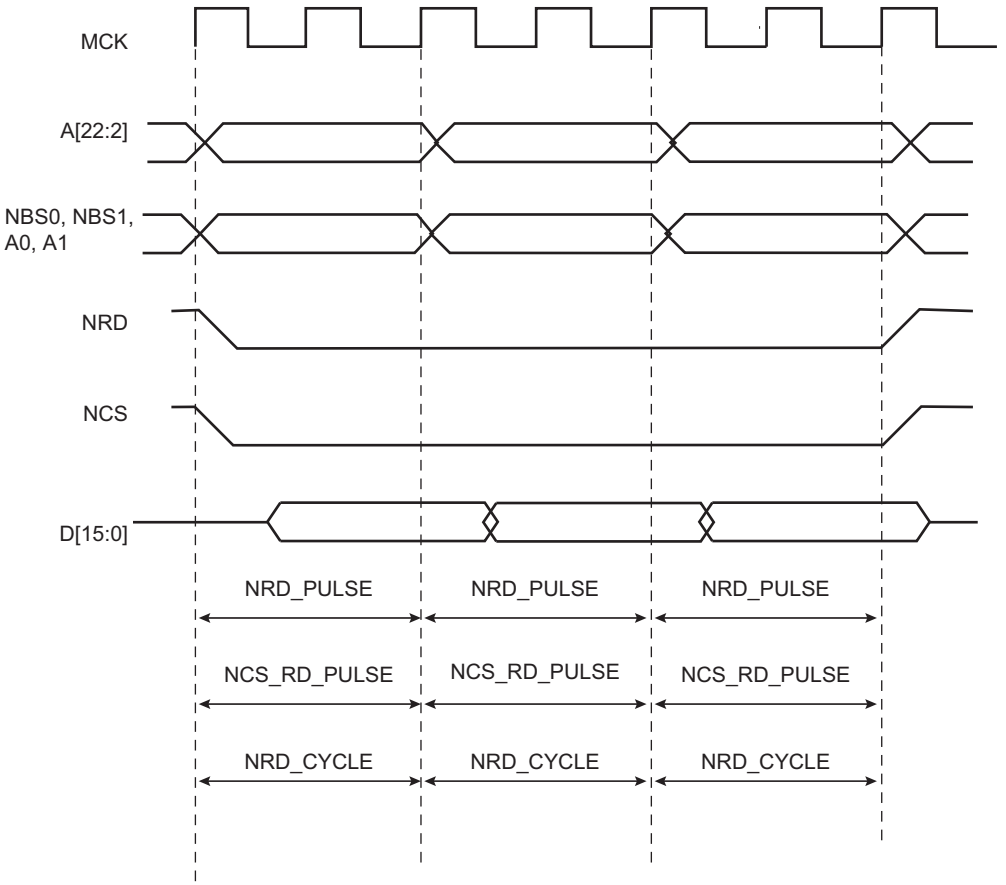
NRD_CYCLE, NCS_RD_SETUP, and NCS_RD_PULSE implicitly define the NCS_RD_HOLD value as:

- NCS_RD_HOLD = NRD_CYCLE - NCS_RD_SETUP - NCS_RD_PULSE

20.10.1.4.Null Delay Setup and Hold

If null setup and hold parameters are programmed for NRD and/or NCS, NRD and NCS remain active continuously in case of consecutive read cycles in the same memory. See the following figure.

Figure 20.7. No Setup, No Hold On NRD and NCS Read Signals



20.10.1.5.Null Pulse

Programming a null pulse is not permitted. Pulse must be at least set to 1. A null value leads to unpredictable behavior.

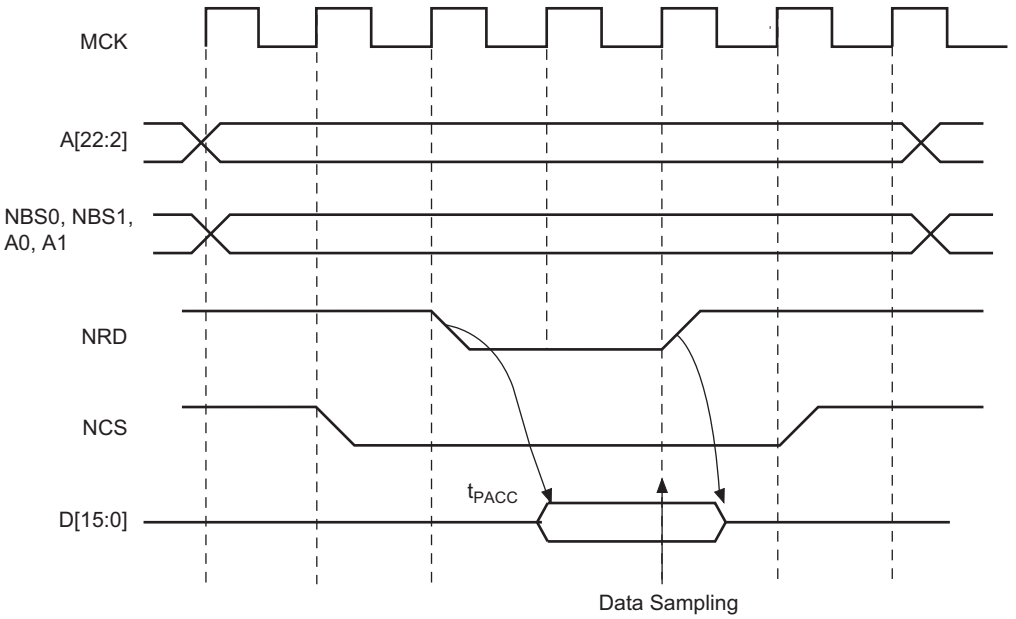
20.10.2. Read Mode

As NCS and NRD waveforms are defined independently of one other, 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 SMC_MODE register of the corresponding chip select indicates which signal of NRD and NCS controls the read operation.

20.10.2.1.Read is Controlled by NRD (READ_MODE = 1)

The following figure 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, regardless of the NCS programmed waveform.

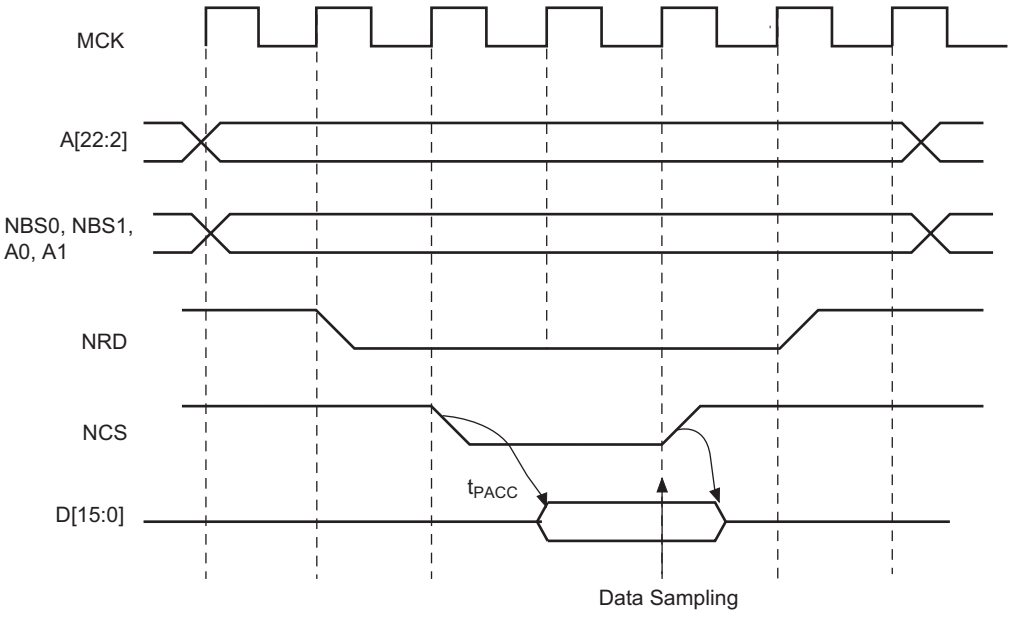
Figure 20.8. READ_MODE = 1 (Data sampled by SMC before rising edge of NRD)



20.10.2.2.Read is Controlled by NCS (READ_MODE = 0)

The following figure shows the typical read cycle of an LCD module. 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, READ_MODE must be set 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, regardless of the NRD programmed waveform.

Figure 20.9. READ_MODE = 0 (Data sampled by SMC before rising edge of NCS)



20.10.3. Write Waveforms

The write protocol (depicted in Figure 20.10) is similar to the read protocol. The write cycle starts with the address setting on the memory address bus.

20.10.3.1.NWE Waveforms

The NWE signal is characterized by a setup timing, a pulse width and a hold timing.

- NWE_SETUP—NWE setup time is defined as the setup of address and data before the NWE falling edge.
- NWE_PULSE—NWE pulse length is the time between NWE falling edge and NWE rising edge.
- NWE_HOLD—NWE hold time is defined as the hold time of address and data after the NWE rising edge.

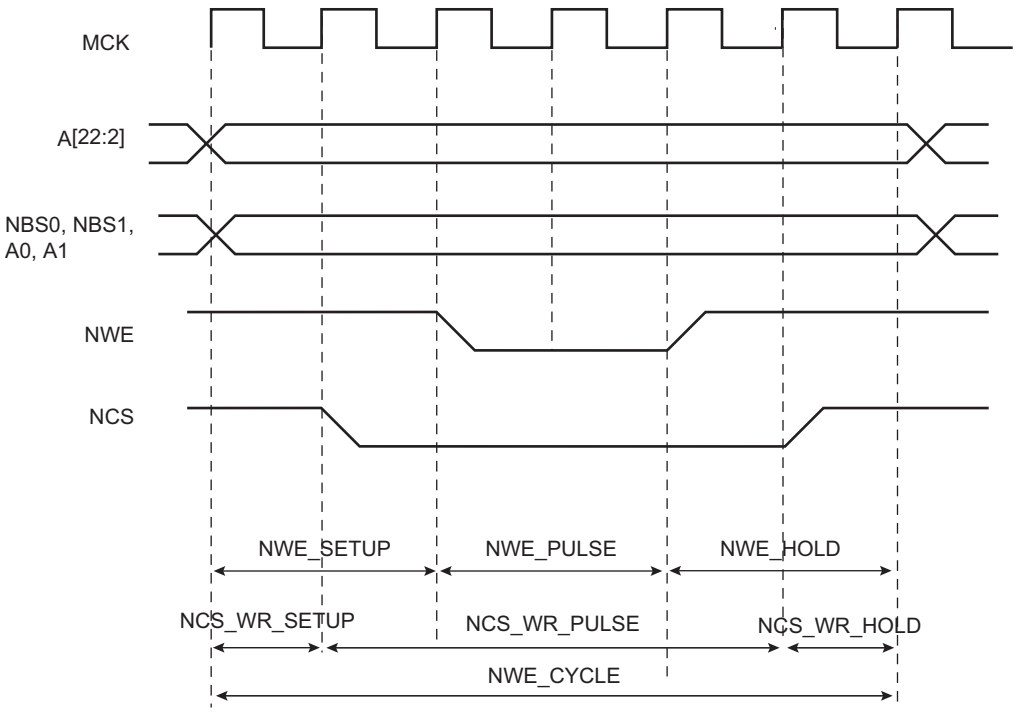
The NWE waveforms apply to all byte-write lines in Byte Write Access mode: NWR0 to NWR1.

20.10.3.2.NCS Waveforms

The NCS signal waveforms in write operations are not the same than those applied in read operations, but are defined separately:

- NCS_WR_SETUP—NCS setup time is defined as the setup time of address before the NCS falling edge.
- NCS_WR_PULSE—NCS pulse length is the time between NCS falling edge and NCS rising edge.
- NCS_WR_HOLD—NCS hold time is defined as the hold time of address after the NCS rising edge.

Figure 20.10. Write Cycle



20.10.3.3.Write Cycle

The write_cycle time is defined as the total duration of the write cycle, that is, from the time where address is set on the address bus to the point where address may change. The total write cycle time is defined as:

- 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 the 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

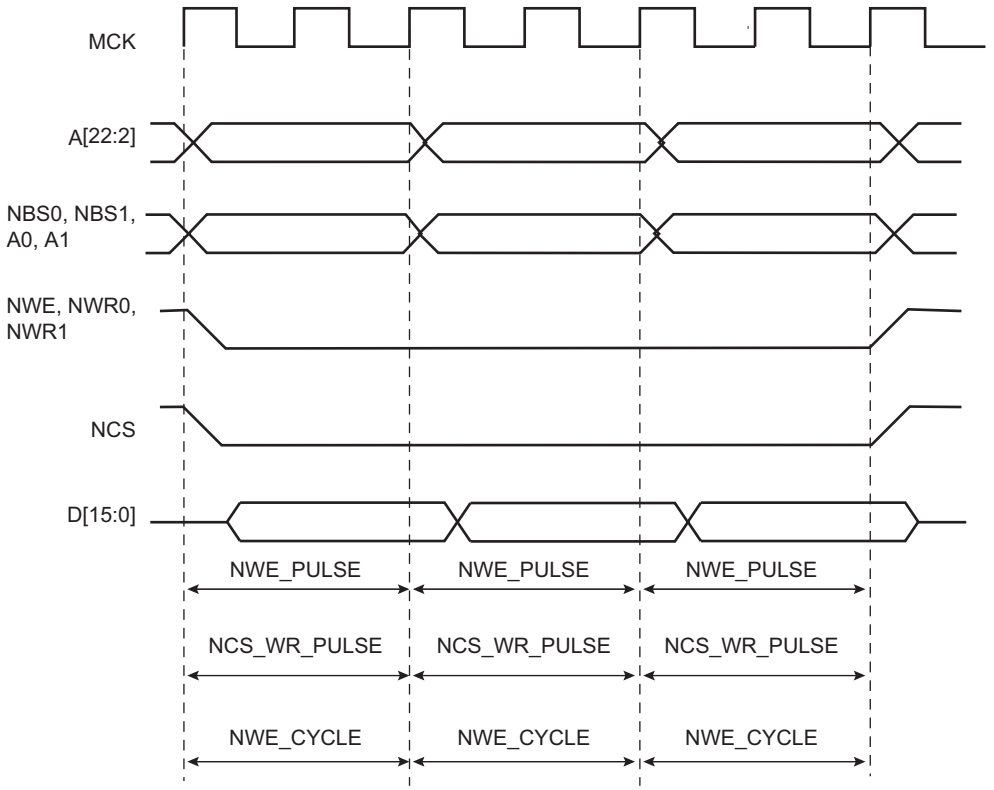
NWE_CYCLE, NCS_WR_SETUP, and NCS_WR_PULSE implicitly define the NCS_WR_HOLD value as:

- NCS_WR_HOLD = NWE_CYCLE - NCS_WR_SETUP - NCS_WR_PULSE

20.10.3.4.Null Delay Setup and Hold

If null setup parameters are programmed for NWE and/or NCS, NWE and/or NCS remain active continuously in case of consecutive write cycles in the same memory. See the following figure. However, for devices that perform write operations on the rising edge of NWE or NCS, such as SRAM, either a setup or a hold must be programmed.

Figure 20.11. Null Setup and Hold Values of NCS and NWE in Write Cycle



20.10.3.5.Null Pulse

Programming a null pulse is not permitted. Pulse must be at least set to 1. A null value leads to unpredictable behavior.

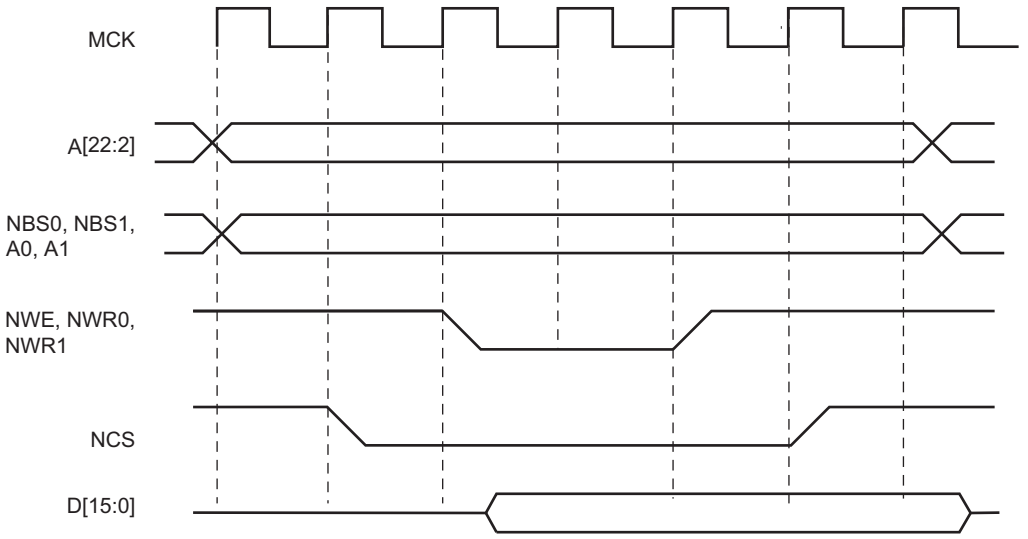
20.10.4. Write Mode

The WRITE_MODE parameter in the SMC_MODE register of the corresponding chip select indicates which signal controls the write operation.

20.10.4.1.Write is Controlled by NWE (WRITE_MODE = 1)

The following figure 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.

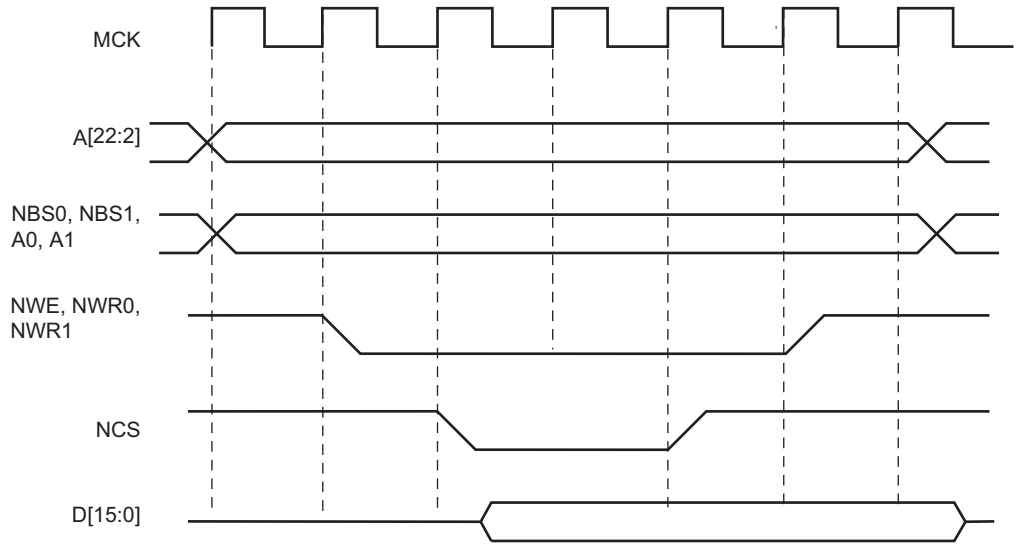
Figure 20.12. WRITE_MODE = 1 (Write Operation Controlled by NWE)



20.10.4.2.Write is Controlled by NCS (WRITE_MODE = 0)

The following figure shows the waveforms of a write operation with WRITE_MODE set 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 20.13. WRITE_MODE = 0 (Write Operation Controlled by NCS)



20.10.5. Coding Timing Parameters

All timing parameters are defined for one chip select and are grouped together in one SMC_REGISTER according to their type.

The SMC_SETUP register groups the definition of all setup parameters:

- NRD_SETUP, NCS_RD_SETUP, NWE_SETUP, NCS_WR_SETUP

The SMC_PULSE register groups the definition of all pulse parameters:

- NRD_PULSE, NCS_RD_PULSE, NWE_PULSE, NCS_WR_PULSE

The SMC_CYCLE register groups the definition of all cycle parameters:

- NRD_CYCLE, NWE_CYCLE

The following table shows how the timing parameters are coded and their permitted range.

Table 20.4. 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 ≤ 31	0 ≤ 128 + 31
pulse [6:0]	7	256 x pulse[6] + pulse[5:0]	0 ≤ 63	0 ≤ 256 + 63
cycle [8:0]	9	256 x cycle[8:7] + cycle[6:0]	0 ≤ 127	0 ≤ 256 + 127 0 ≤ 512 +1 27 0 ≤ 768 + 127

20.10.6. Reset Values of Timing Parameters

For the default values of timing parameters at reset, see SMC_SETUPx, SMC_PULSEx and SMC_CYCLEx.

20.10.7. 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 unpredictable behavior of the SMC.

- For read operations:
  Null but positive setup and hold of address and NRD and/or NCS can not be guaranteed at the memory interface because of the propagation delay of theses signals through external logic and pads. If 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.
- For write operations:
  If a null hold value is programmed on NWE, the SMC can guarantee 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.
- For read and write operations:
  A null value for pulse parameters is forbidden and may lead to 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.

20.11. Automatic Wait States

Under certain circumstances, the SMC automatically inserts idle cycles between accesses to avoid bus contention or operation conflict.

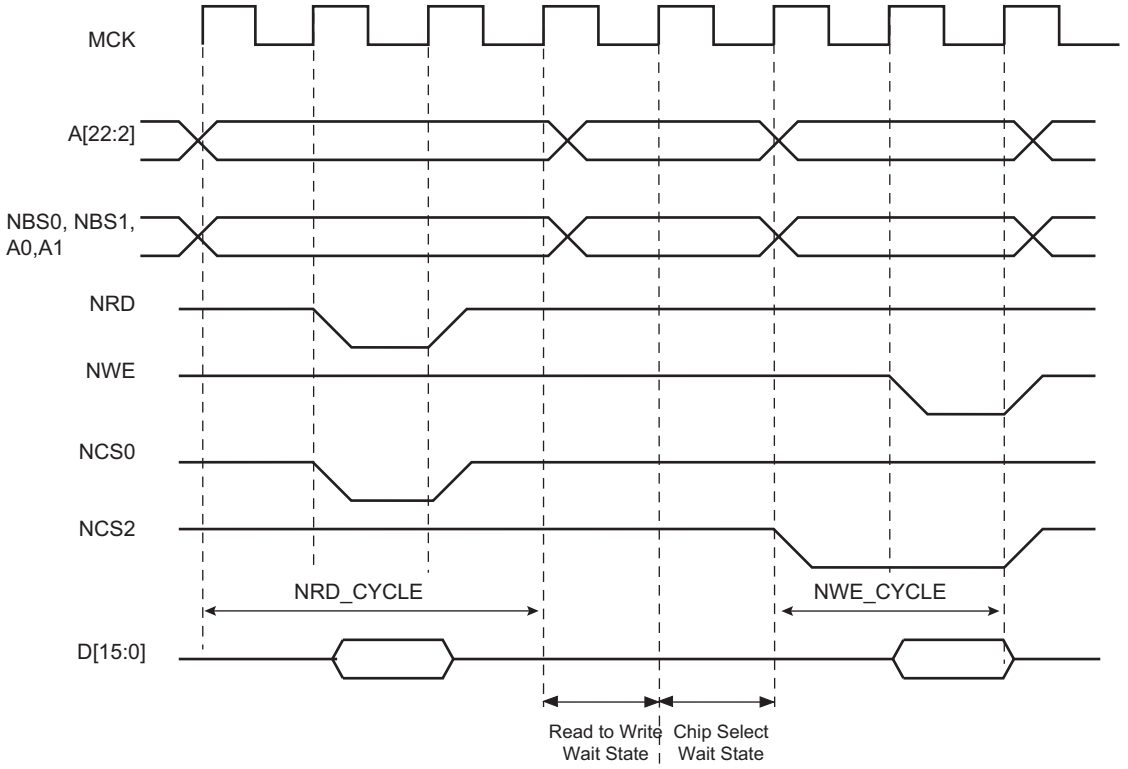
20.11.1. Chip Select Wait States

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 de-activation 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..2], NRD lines are all set to 1.

The following figure illustrates a Chip Select Wait state between accesses on Chip Select 0 and Chip Select 2.

Figure 20.14. Chip Select Wait State between a Read Access on NCS0 and a Write Access on NCS2



20.11.2. 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:

- The write controlling signal has no hold time and the read controlling signal has no setup time (Figure 20.15).
- In NCS Write Controlled mode (WRITE_MODE = 0), there is no hold timing on the NCS signal and the NCS_RD_SETUP parameter is set to 0, regardless of the read mode (Figure 20.16). The write operation must end with a 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 20.17.

Figure 20.15. Early Read Wait State: Write with No Hold Followed by Read with No Setup

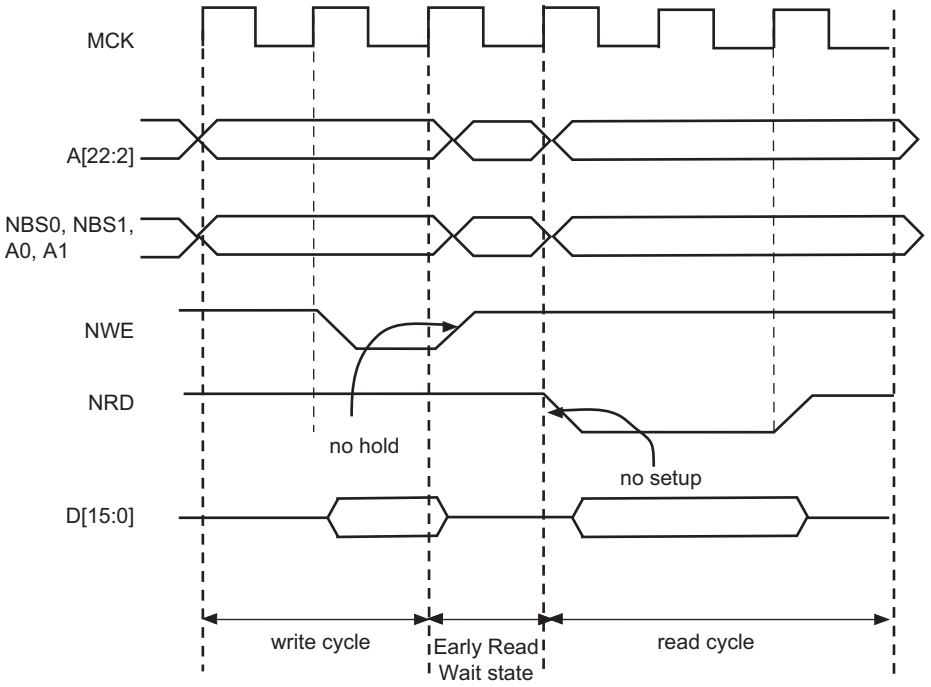


Figure 20.16. Early Read Wait State: NCS Controlled Write with No Hold Followed by a Read with No NCS Setup

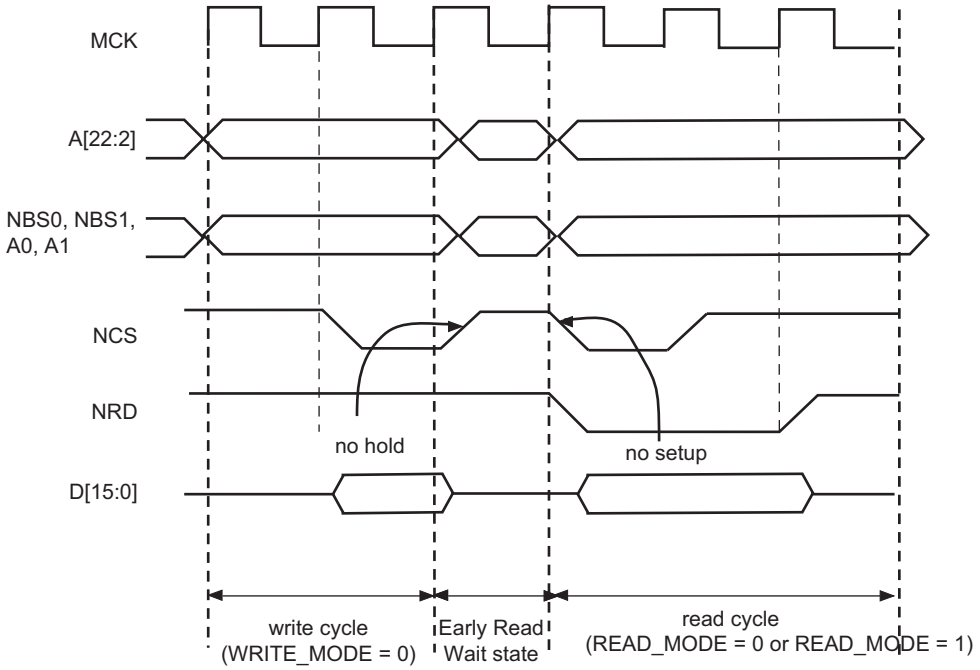
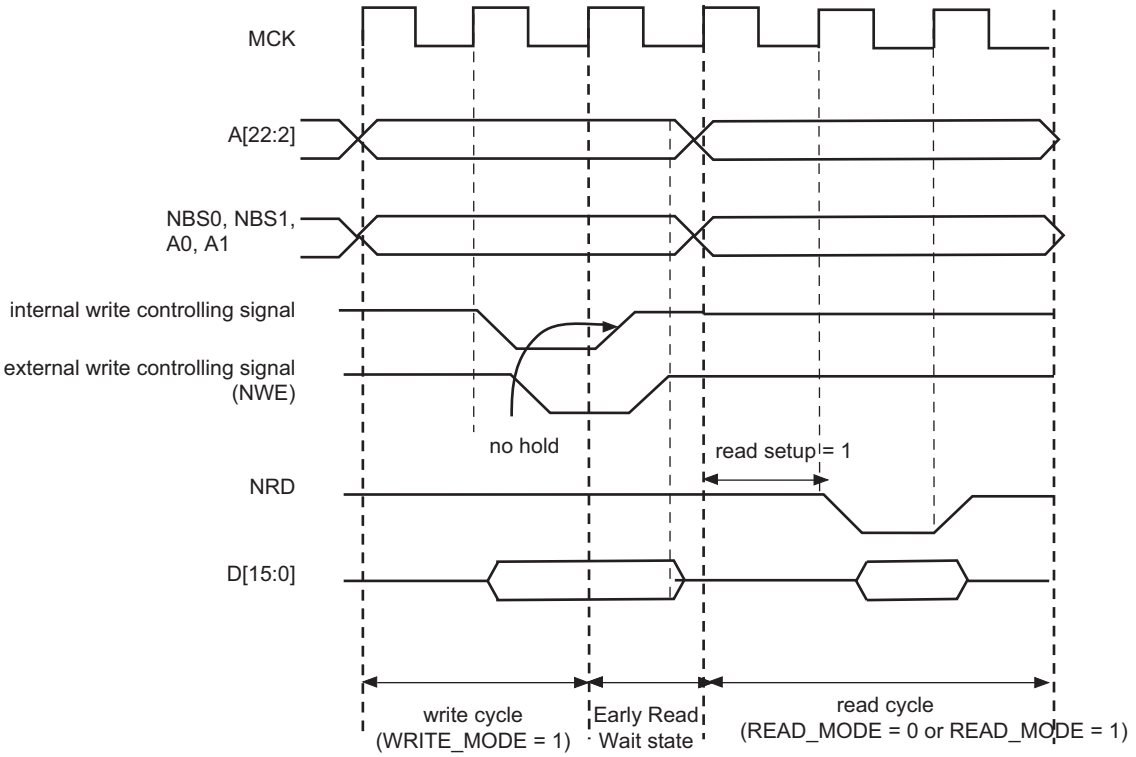


Figure 20.17. Early Read Wait State: NWE-controlled Write with No Hold Followed by a Read with one Set-up Cycle



20.11.3. 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 re-programming 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.

20.11.3.1. User Procedure

To insert a Reload Configuration Wait State, the SMC detects a write access to any SMC_MODE register of the user interface. If the user only modifies timing registers (SMC_SETUP, SMC_PULSE, SMC_CYCLE registers) in the user interface, he must validate the modification by writing the SMC_MODE, even if no change was made on the mode parameters.

The user must not change the configuration parameters of an SMC Chip Select (Setup, Pulse, Cycle, Mode) if accesses are performed on this CS during the modification. Any change of the Chip Select parameters, while fetching the code from a memory connected on this CS, may lead to unpredictable behavior. The instructions used to modify the parameters of an SMC Chip Select can be executed from the internal RAM or from a memory connected to another CS.

20.11.3.2. Slow Clock Mode Transition

A Reload Configuration Wait state is also inserted when the Slow Clock mode is entered or exited, after the end of the current transfer (see Slow Clock Mode).

20.11.4. 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 20.14.

20.12. Data Float Wait States

Some memory devices are slow to release the external bus. For such devices, it is necessary to add wait states (data float wait states) after a read access:

- before starting a read access to a different external memory
- before starting a write access to the same device or to a different external one.

The Data Float Output Time (t_{DF}) for each external memory device is programmed in the TDF_CYCLES field of the SMC_MODE register for the corresponding chip select. The value of TDF_CYCLES indicates the number of data float wait cycles (between 0 and 15) before the external device releases the bus, and represents the time allowed for the data output to go to high impedance after the memory is disabled.

Data float wait states do not delay internal memory accesses. Hence, a single access to an external memory with long t_{DF} will not slow down the execution of a program from internal memory.

The data float wait states management depends on the READ_MODE and the TDF_MODE fields of the SMC_MODE register for the corresponding chip select.

20.12.1. READ_MODE

Setting READ_MODE to 1 indicates to the SMC that the NRD signal is responsible for turning off the tri-state buffers of the external memory device. The Data Float Period then begins after the rising edge of the NRD signal and lasts TDF_CYCLES MCK cycles.

When the read operation is controlled by the NCS signal (READ_MODE = 0), the TDF field gives the number of MCK cycles during which the data bus remains busy after the rising edge of NCS.

Figure 20.18 illustrates the Data Float Period in NRD-controlled mode (READ_MODE = 1), assuming a data float period of two cycles (TDF_CYCLES = 2). Figure 20.19 shows the read operation when controlled by NCS (READ_MODE = 0) and the TDF_CYCLES parameter equals 3.

Figure 20.18. TDF Period in NRD Controlled Read Access (TDF = 2)

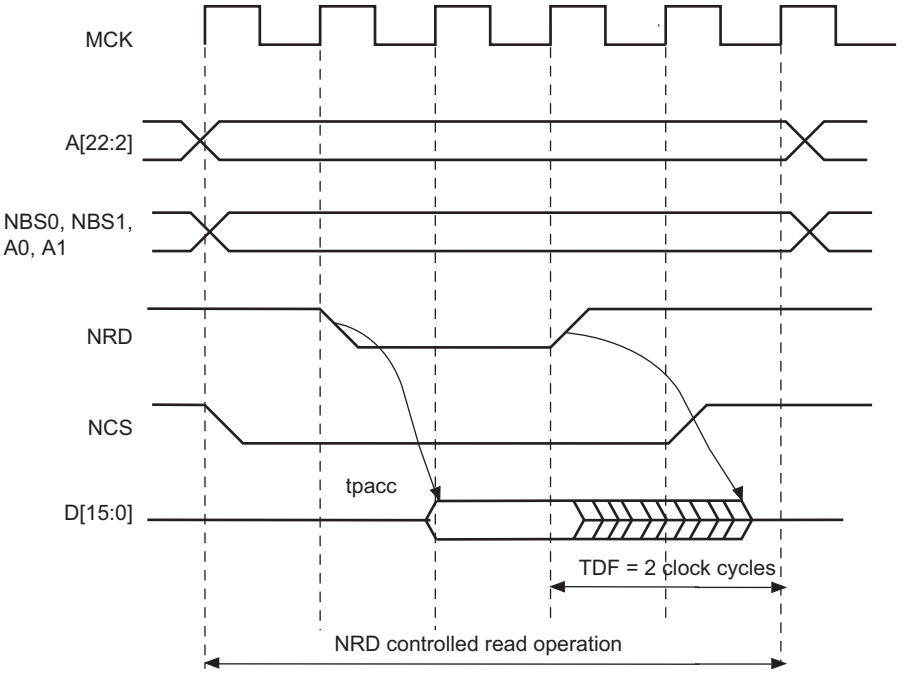
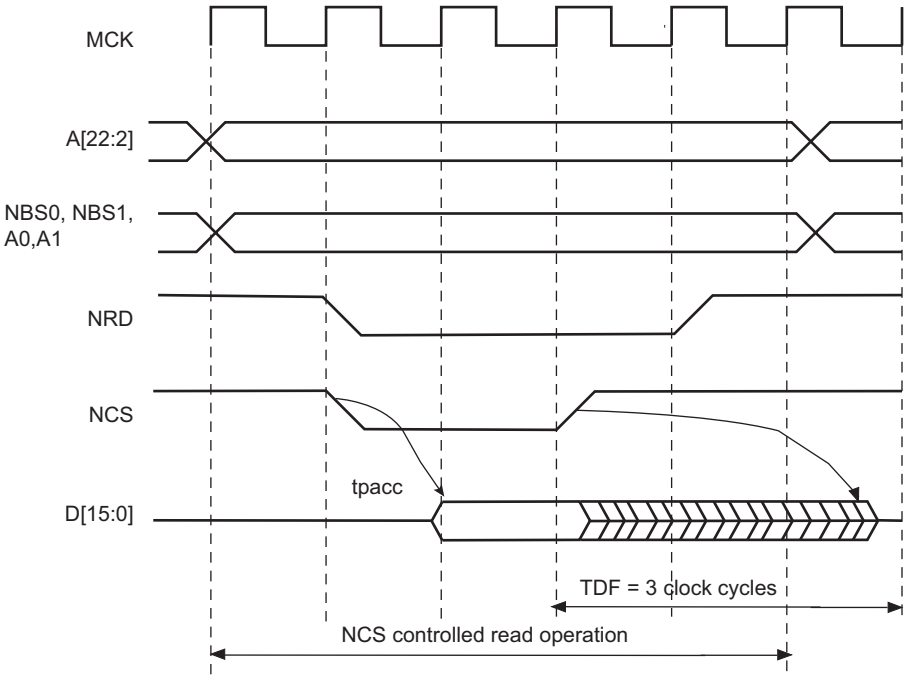


Figure 20.19. TDF Period in NCS Controlled Read Operation (TDF = 3)



20.12.2. TDF Optimization Enabled (TDF_MODE = 1)

When SMC_MODE.TDF_MODE is set to 1 (TDF optimization enabled), the SMC takes advantage of the setup period of the next access to optimize the number of Wait state cycles to insert.

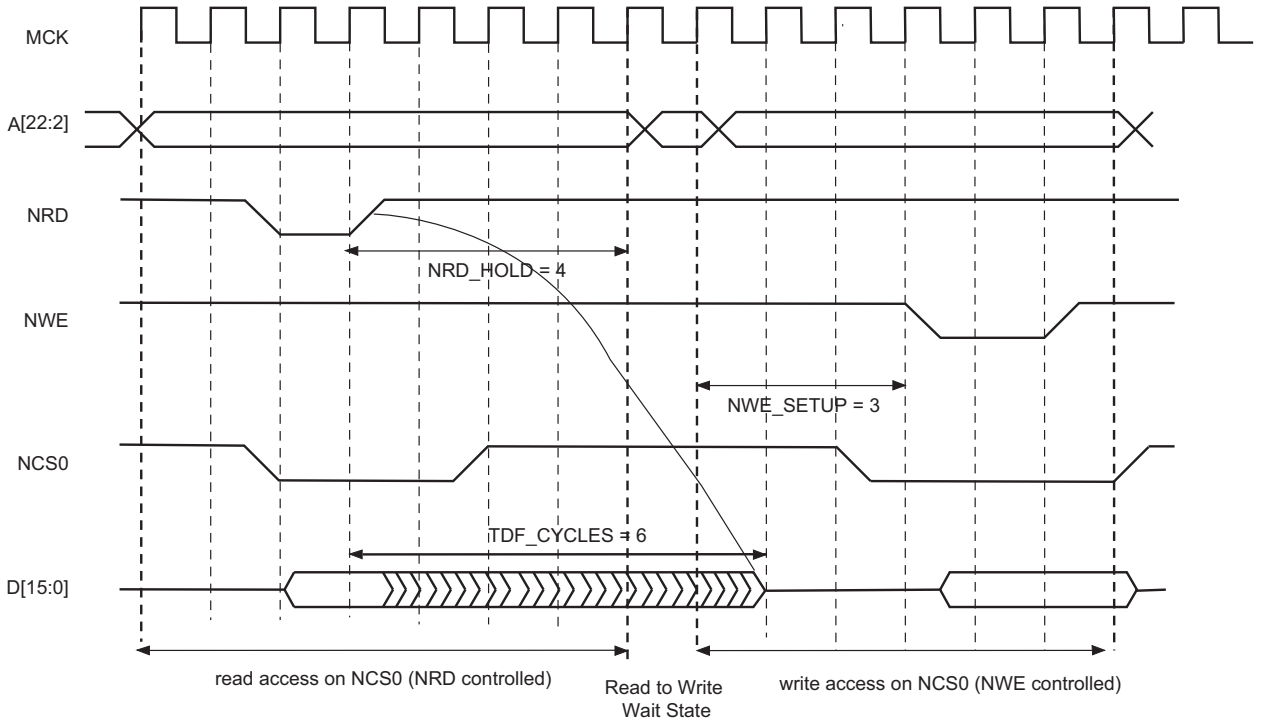
The following figure shows a read access controlled by NRD, followed by a write access controlled by NWE, on Chip Select 0. Chip Select 0 has been programmed with:

NRD_HOLD = 4; READ_MODE = 1 (NRD controlled)

NWE_SETUP = 3; WRITE_MODE = 1 (NWE controlled)

TDF_CYCLES = 6; TDF_MODE = 1 (optimization enabled).

Figure 20.20. TDF Optimization: No TDF Wait states are inserted if the TDF period is over when the next access begins



20.12.3. TDF Optimization Disabled (TDF_MODE = 0)

When optimization is disabled, TDF Wait states are inserted at the end of the read transfer, so that the data float period is ended when the second access begins. If the hold period of the read1 controlling signal overlaps the data float period, no additional TDF Wait states will be inserted.

Figure 20.21, Figure 20.22 and Figure 20.23 illustrate the following cases, with no TDF optimization:

- Read access followed by a read access on another chip select
- Read access followed by a write access on another chip select
- Read access followed by a write access on the same chip select

Figure 20.21. TDF Optimization Disabled (TDF Mode = 0): TDF Wait states between 2 read accesses on different chip selects

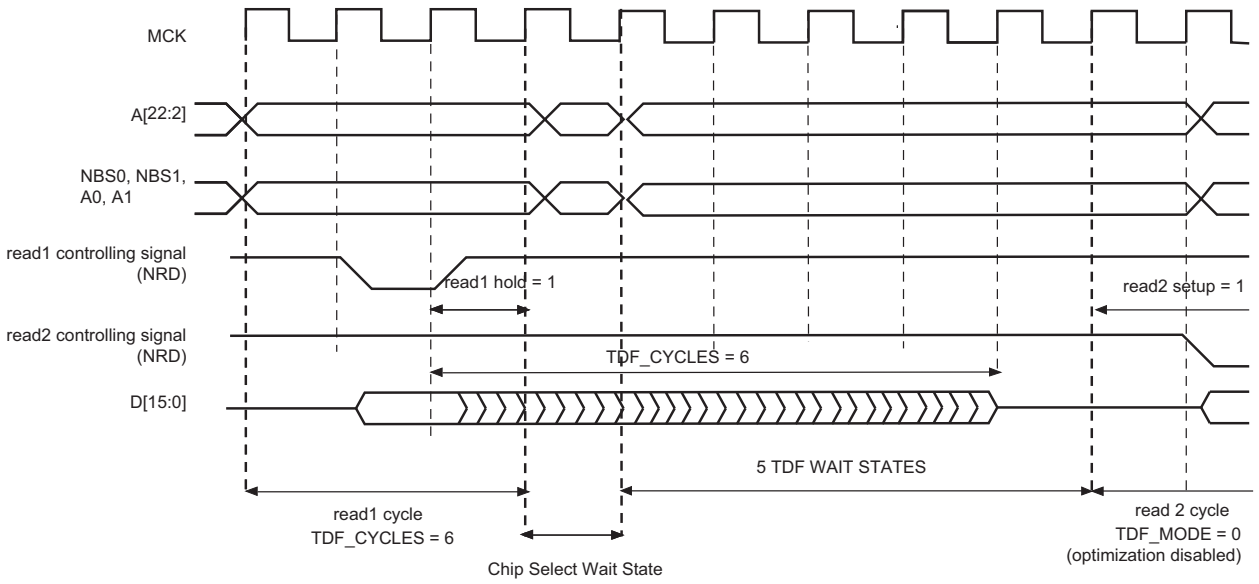


Figure 20.22. TDF Mode = 0: TDF Wait states between read and write accesses on different chip selects

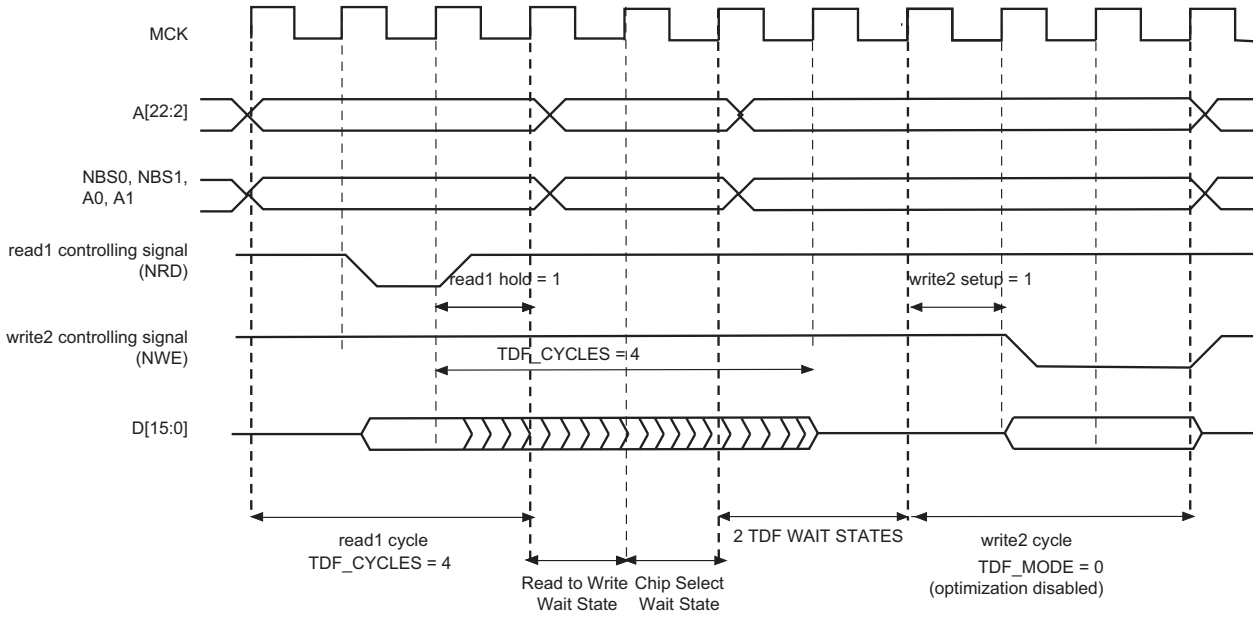
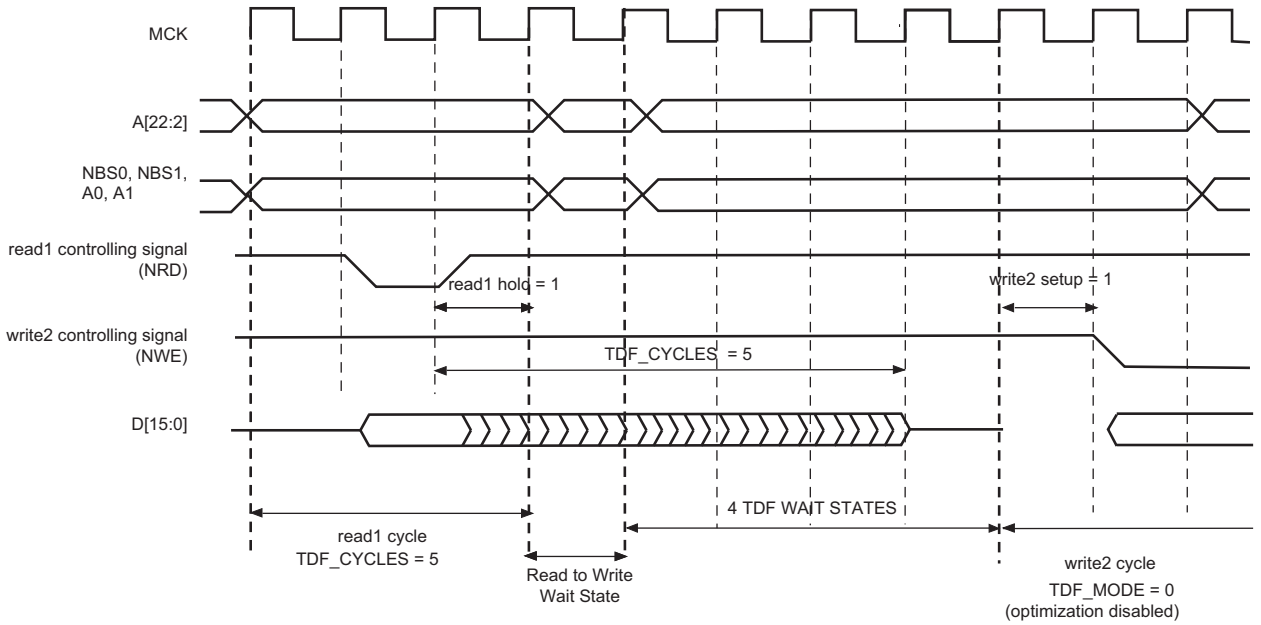


Figure 20.23. TDF Mode = 0: TDF Wait states between read and write accesses on the same chip select



20.13. 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 SMC_MODE register on the corresponding chip select must be set to either to '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 as regards the read or write controlling signal, depending on the read and write modes of the corresponding chip select.

20.13.1. Restriction

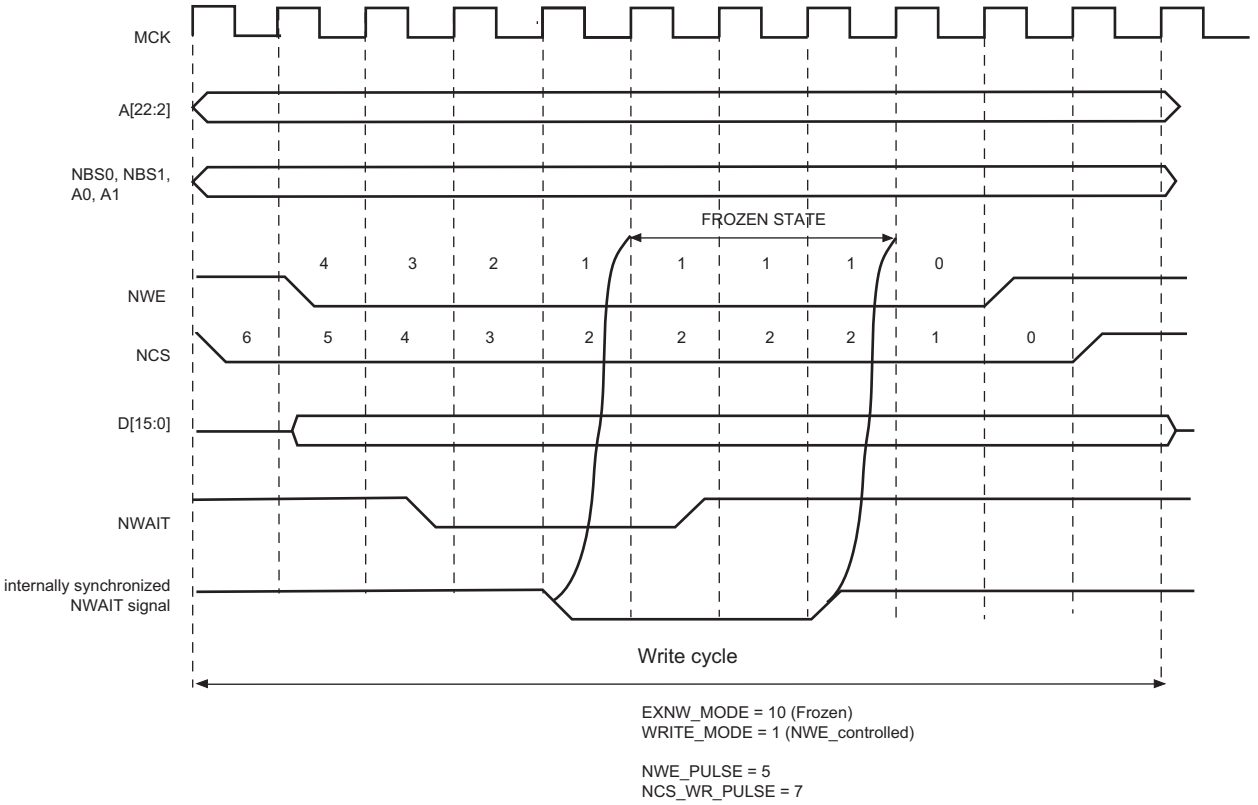
When one of the EXNW_MODE is enabled, at least one hold cycle must be programmed for the read/write controlling signal. For that reason, the NWAIT signal cannot be used in Page mode (see Asynchronous Page Mode) nor in Slow Clock mode (see Slow Clock Mode).

The NWAIT signal is assumed to be a response of the external device to the read/write request of the SMC. Then NWAIT is examined by the SMC only 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 SMC behavior.

20.13.2. Frozen Mode

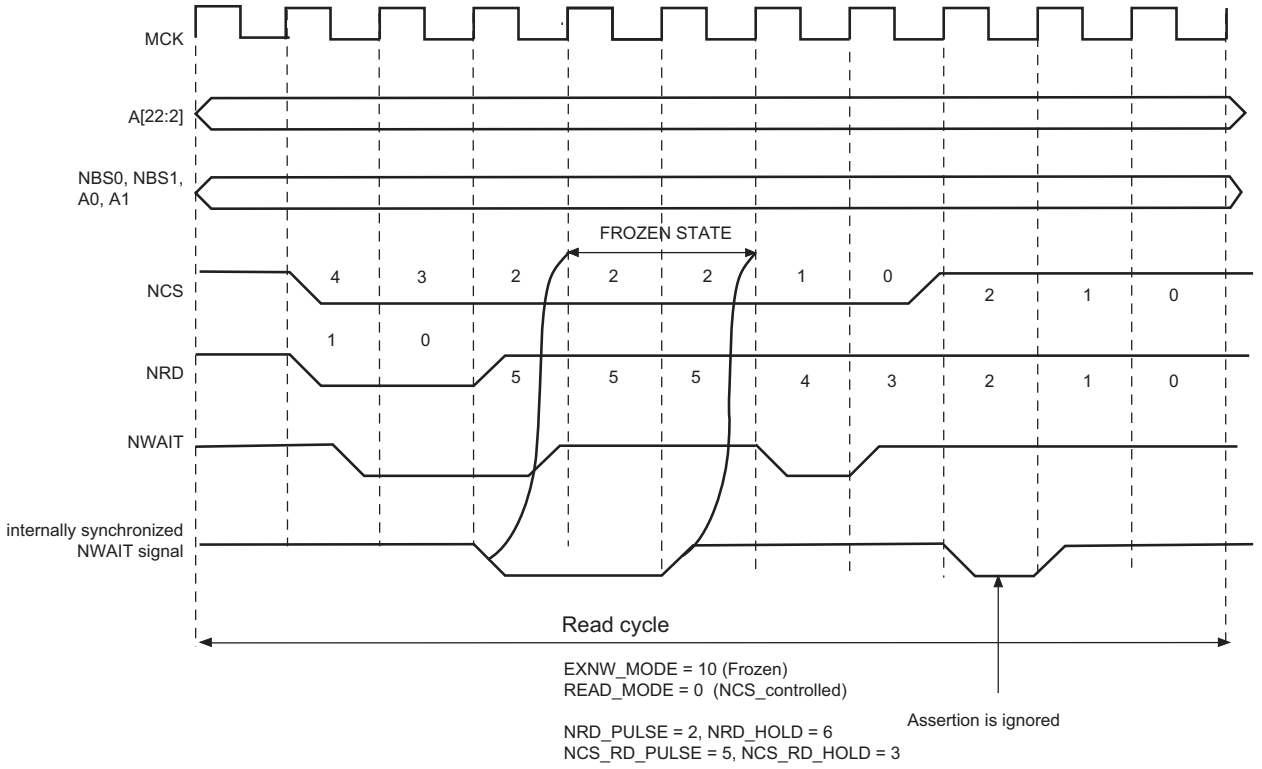
When the external device asserts the NWAIT signal (active low), and after internal synchronization of this signal, the SMC state is frozen, i.e., SMC internal counters are frozen, and all control signals remain unchanged. When the resynchronized NWAIT signal is deasserted, the SMC completes the access, resuming the access from the point where it was stopped. See the following figure. This mode must be selected when the external device uses the NWAIT signal to delay the access and to freeze the SMC.

Figure 20.24. Write Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



The assertion of the NWAIT signal outside the expected period is ignored as illustrated in the figure below.

Figure 20.25. Read Access with NWAIT Assertion in Frozen Mode (EXNW_MODE = 10)



20.13.3. 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 Figure 20.26 and Figure 20.27. 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 Figure 20.27.

Figure 20.26. NWAIT Assertion in Write Access: Ready Mode (EXNW_MODE = 11)

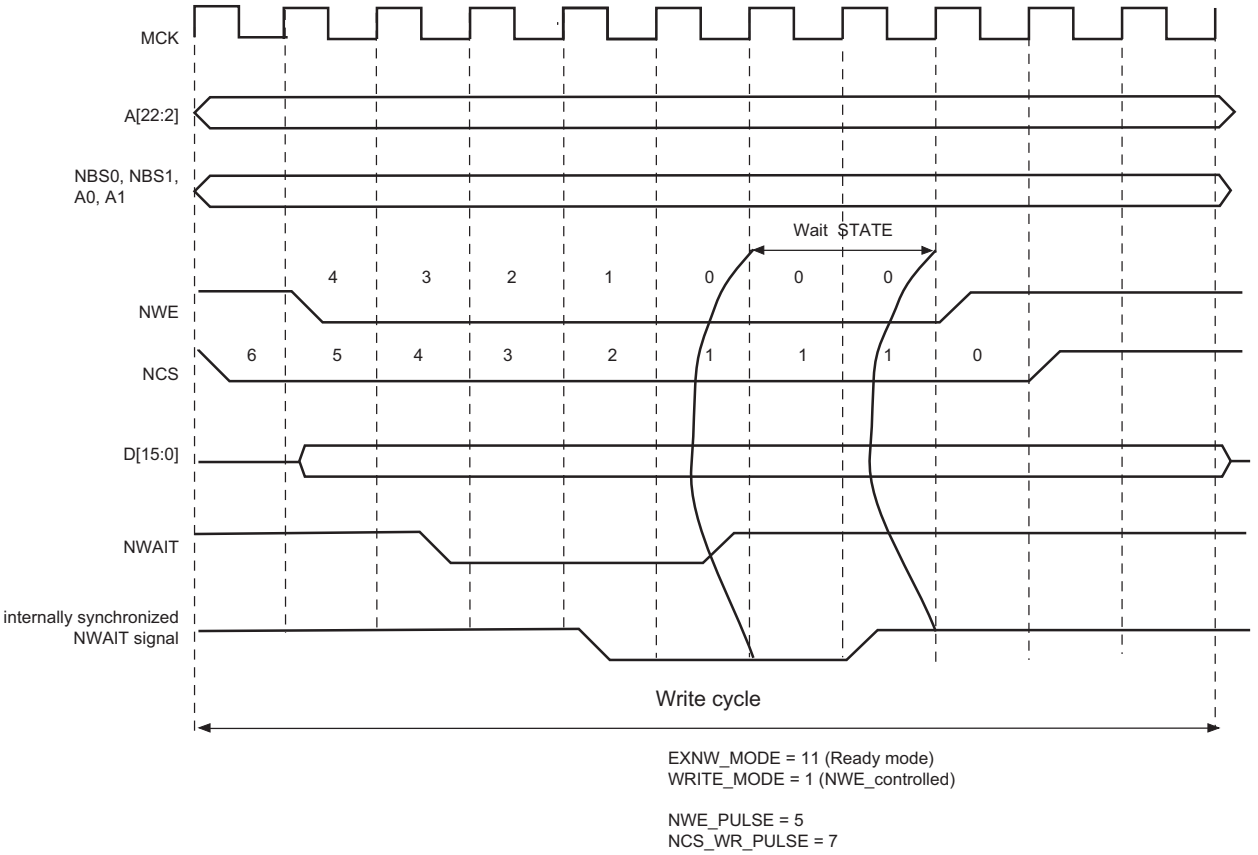
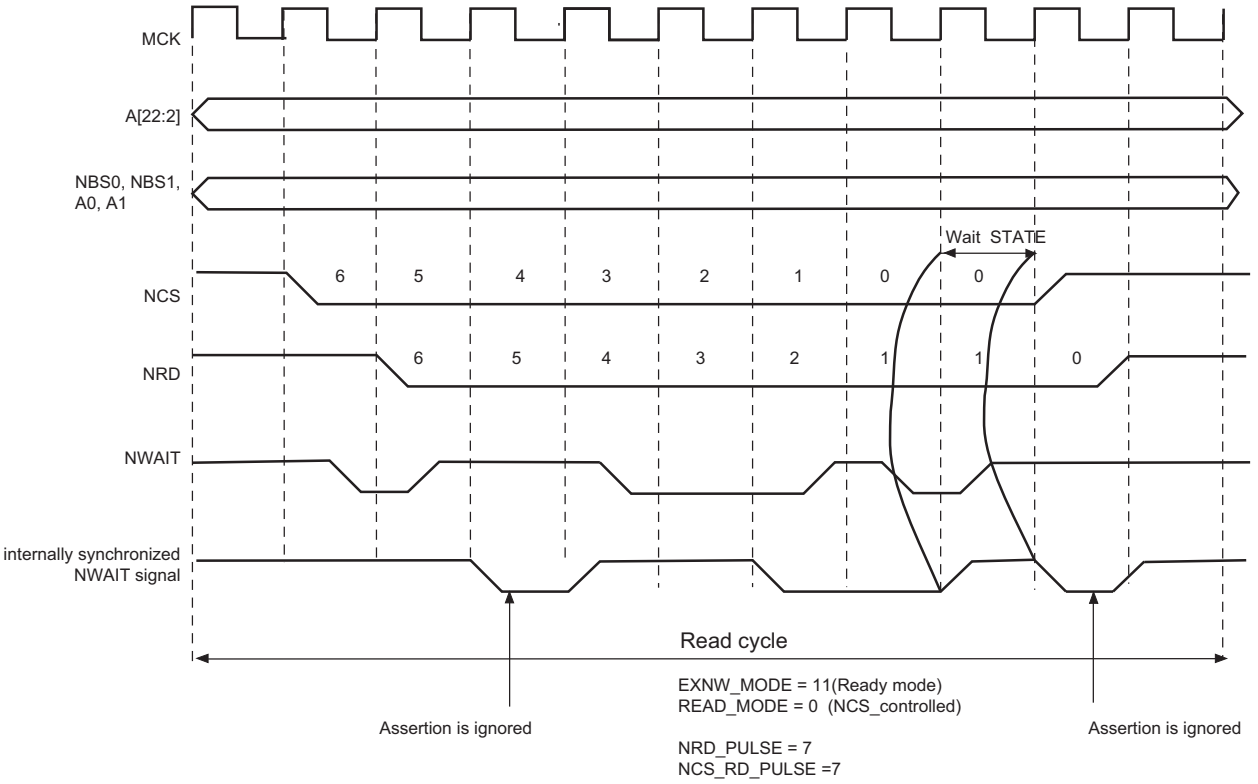


Figure 20.27. NWAIT Assertion in Read Access: Ready Mode (EXNW_MODE = 11)



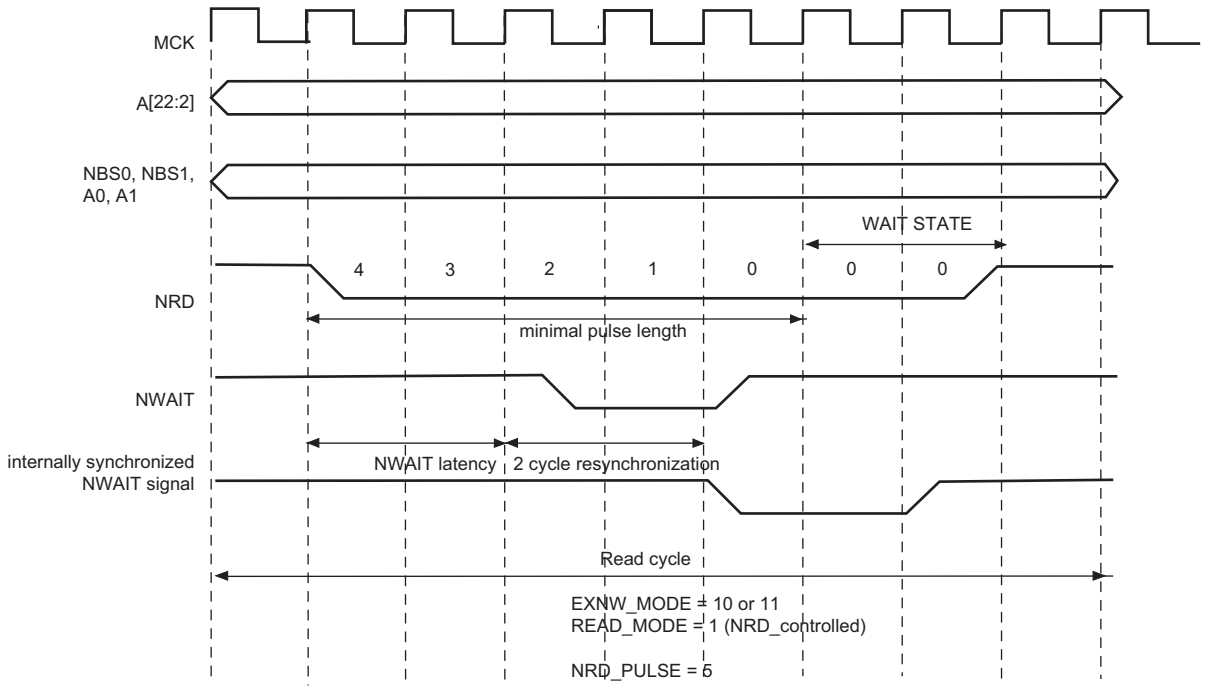
20.13.4. NWAIT Latency and Read/Write Timings

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 resynchronization cycles + 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 following figure.

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 20.28. NWAIT Latency



20.14. 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 configured to a very slow clock rate (typically, 32 kHz). In this mode, the user-programmed waveforms are ignored and the Slow Clock mode waveforms are applied. This mode is provided to avoid reprogramming the User Interface with appropriate waveforms at very slow clock rates. When activated, Slow Clock mode is active on all chip selects.

20.14.1. 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 following table indicates the value of read and write parameters in Slow Clock mode.

Figure 20.29. Read/Write Cycles in Slow Clock Mode

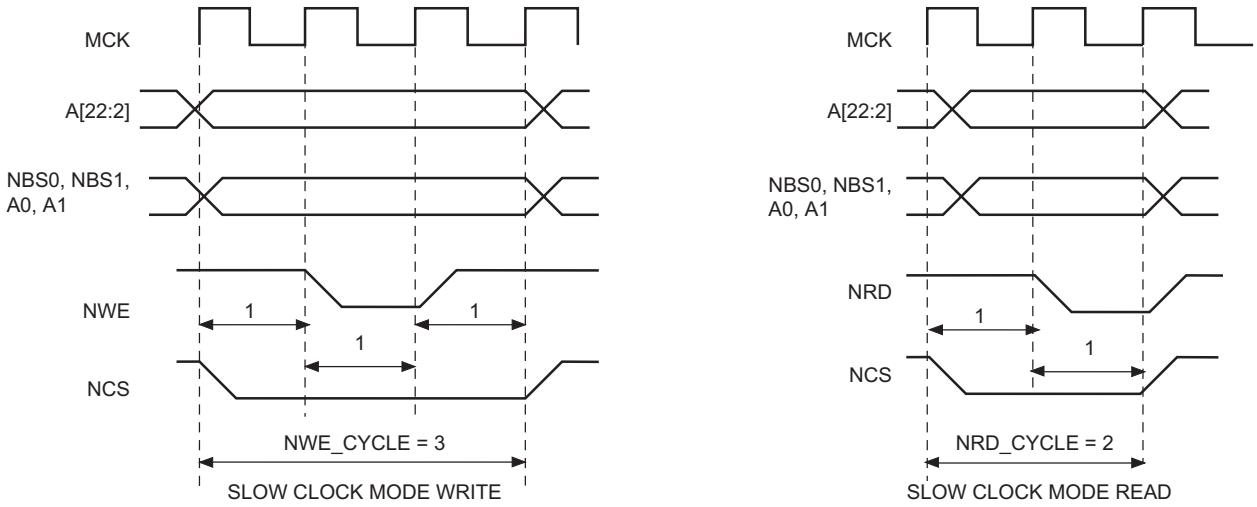


Table 20.5. Read and Write Timing Parameters in Slow Clock Mode

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

20.14.2. 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 Figure 20.30. The external device may not be fast enough to support such timings.

Figure 20.31 illustrates the recommended procedure to properly switch from one mode to the other.

Figure 20.30. Clock Rate Transition Occurs while the SMC is Performing a Write Operation

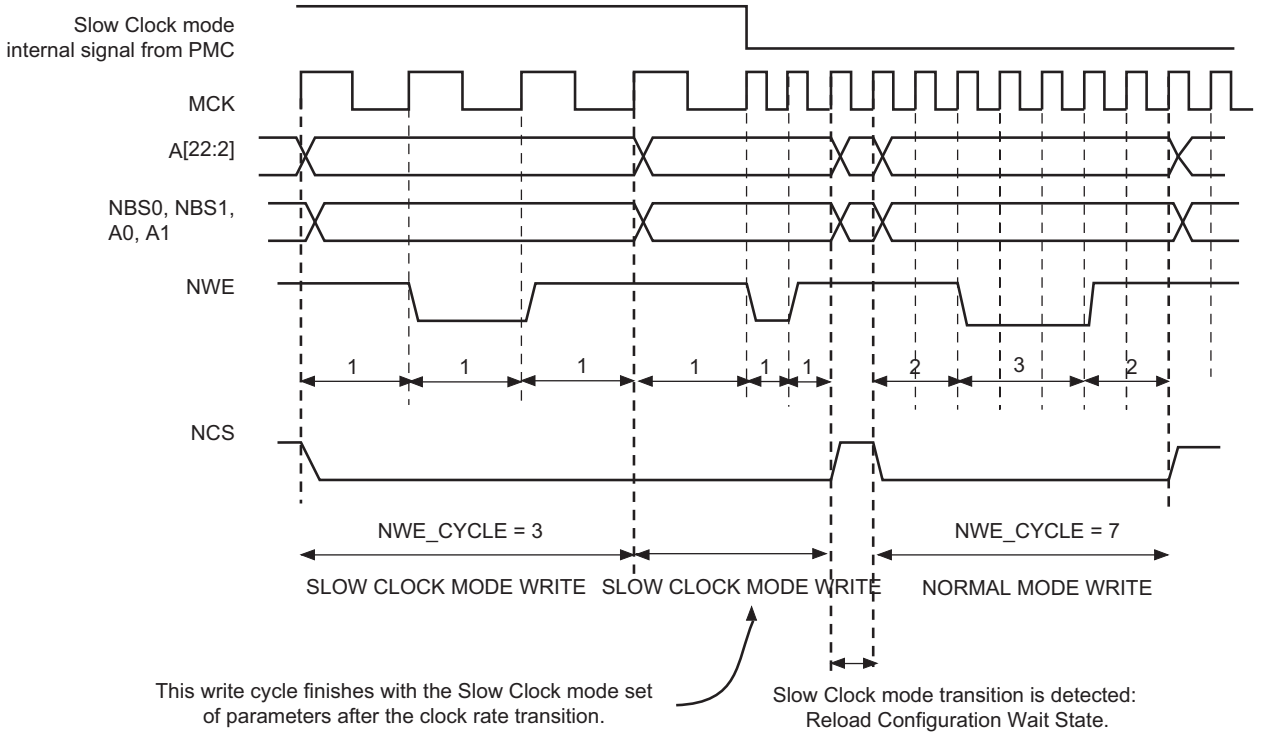
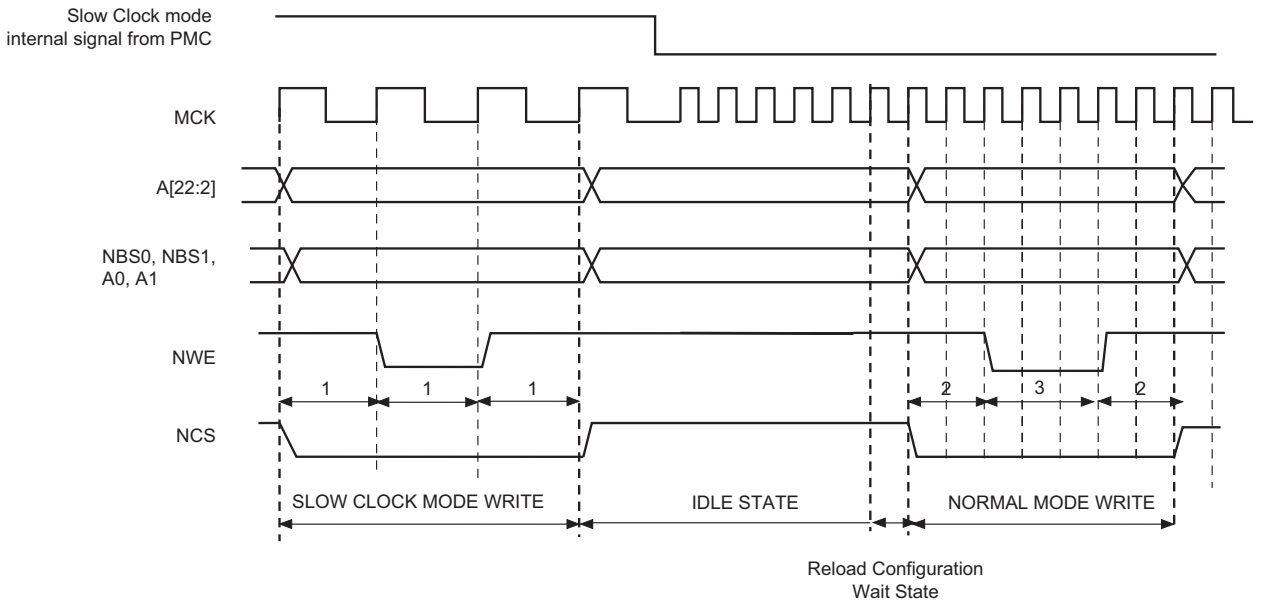


Figure 20.31. Recommended Procedure to Switch from Slow Clock Mode to Normal Mode or from Normal Mode to Slow Clock Mode



20.15. Asynchronous Page Mode

The SMC supports asynchronous burst reads in Page mode, providing that the Page mode is enabled in the SMC_MODE register (PMEN field). The page size must be configured in the SMC_MODE register (PS field) to 4, 8, 16 or 32 bytes.

The page defines a set of consecutive bytes into memory. A 4-byte page (resp. 8-, 16-, 32-byte page) is always aligned to 4-byte boundaries (resp. 8-, 16-, 32-byte boundaries) of memory. The MSB of

data address defines the address of the page in memory, the LSB of address define the address of the data in the page as detailed in the table below.

With Page mode memory devices, the first access to one page (t_{pa}) takes longer than the subsequent accesses to the page (t_{sa}) as shown in Figure 20.32. When in Page mode, the SMC enables the user to define different read timings for the first access within one page, and next accesses within the page.

Table 20.6. Page Address and Data Address Within a Page

Page Size	Page Address (1)	Data Address in the Page (2)
4 bytes	A[22:2]	A[1:0]
8 bytes	A[22:3]	A[2:0]
16 bytes	A[22:4]	A[3:0]
32 bytes	A[22:5]	A[4:0]

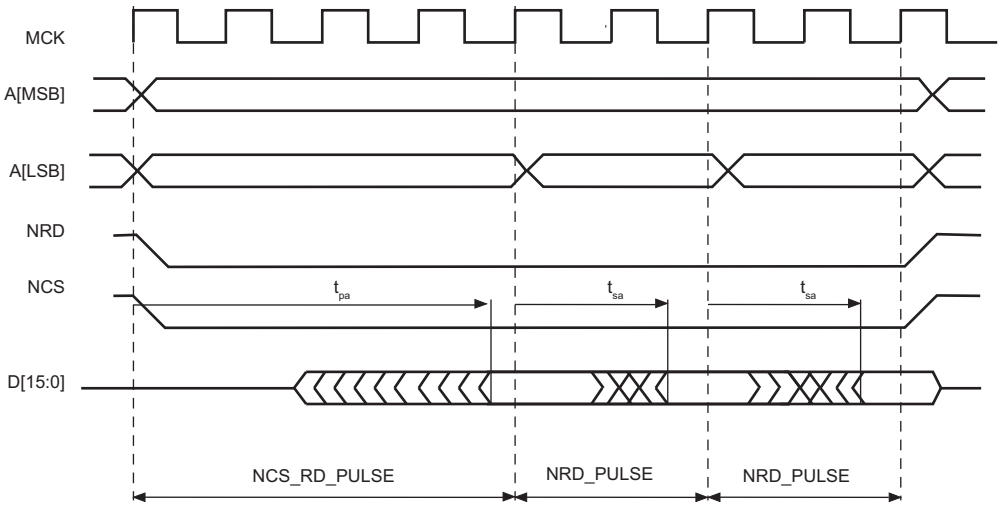
Notes:

1. 'A' denotes the address bus of the memory device.
2. For 16-bit devices, bit 0 of the address is ignored.

20.15.1. Protocol and Timings in Page Mode

The following figure shows the NRD and NCS timings in Page mode access. For the address MSB and LSB, see Table 20.6.

Figure 20.32. Page Mode Read Protocol



The NRD and NCS signals are held low during all read transfers, whatever the programmed values of the setup and hold timings in the User Interface may be. Moreover, the NRD and NCS timings are identical. The pulse length of the first access to the page is defined with the NCS_RD_PULSE field of the SMC_PULSE register. The pulse length of subsequent accesses within the page are defined using the NRD_PULSE parameter.

Programming of the read timings in Page mode is described in the following table.

Table 20.7. Programming of Read Timings in Page Mode

Parameter	Value	Definition
READ_MODE	'x'	No impact
NCS_RD_SETUP	'x'	No impact
NCS_RD_PULSE	t_{pa}	Access time of first access to the page

Table 20.7. Programming of Read Timings in Page Mode (continued)

Parameter	Value	Definition
NRD_SETUP	'x'	No impact
NRD_PULSE	t_{sa}	Access time of subsequent accesses in the page
NRD_CYCLE	'x'	No impact

The SMC does not check the coherency of timings. It will always apply the NCS_RD_PULSE timings as page access timing (t_{pa}) and the NRD_PULSE for accesses to the page (t_{sa}), even if the programmed value for t_{pa} is shorter than the programmed value for t_{sa}.

20.15.2. Byte Access Type in Page Mode

The byte access type (BAT) configuration remains active in Page mode. For 16-bit Page mode devices that require byte selection signals, write a 0 to the BAT bit in the SMC Mode register (SMC_MODE) to select the byte select access type.

20.15.3. Page Mode Restriction

The Page mode is not compatible with the use of the NWAIT signal. Using the Page mode and the NWAIT signal may lead to unpredictable behavior.

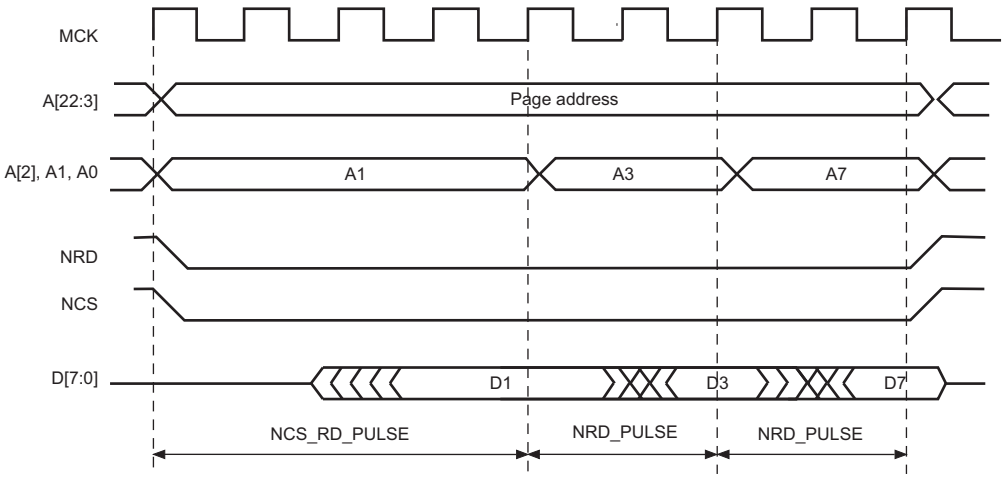
20.15.4. Sequential and Non-Sequential Accesses

If the chip select and the MSB of addresses as defined in Table 20.6 are identical, then the current access lies in the same page as the previous one, and no page break occurs.

Using this information, all data within the same page, sequential or not sequential, are accessed with a minimum access time (t_{sa}). The following figure illustrates access to an 8-bit memory device in Page mode, with 8-byte pages. Access to D1 causes a page access with a long access time (t_{pa}). Accesses to D3 and D7, though they are not sequential accesses, only require a short access time (t_{sa}).

If the address MSBs differ, the SMC performs the access of a new page. In the same way, if the chip select is different from the previous access, a page break occurs. If two sequential accesses are made to the Page mode memory, but separated by an other internal or external peripheral access, a page break occurs on the second access because the chip select of the device was deasserted between both accesses.

Figure 20.33. Access to Non-sequential Data within the Same Page



20.16. Register Write Protection

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

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status Register (SMC_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 SMC_WPSR.

The following registers can be write-protected:

- SMC Setup Register
- SMC Pulse Register
- SMC Cycle Register
- SMC Mode Register
- SMC Off-Chip Memory Scrambling Register
- SMC Safety Report Interrupt Enable Register

20.17. Security and Safety Analysis and Reports

Several types of checks are performed when the SMC is reading or writing an external memory.

The internal sequencer of the SMC is monitored for integrity and if an abnormal state is detected, the flag SMC_WPSR.SEQE is set. This flag is not set under normal operating conditions.

The software accesses to the SMC are monitored and if an incorrect access is performed, the flag SMC_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the SMC_WPSR.SWETYP field (see SMC Write Protection Status Register for details). The flags SEQE, SWE and WPVS are automatically cleared when SMC_WPSR is read.

If one of these flags is set, an interrupt can be triggered if the SMC_SRIER.SRIE bit is ‘1’.

20.18. Scrambling/Unscrambling Function

The external data bus can be scrambled to make more difficult the recovery of intellectual property data located in off-chip memories by means of data analysis 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/unscrambling function can be enabled or disabled by configuring the CSxSE bits in the SMC Off-Chip Memory Scrambling Register (SMC_OCMS).

When multiple chip selects are handled, the scrambling function per chip select is configurable using the CSxSE bits in the SMC_OCMS register.

The scrambling method depends on two user-configurable key registers, SMC_KEY1 and SMC_KEY2 plus a random value depending on device processing characteristics. These key registers cannot be read. They can be written once after a system reset.

The scrambling user key or the seed for key generation must be securely stored in a reliable non-volatile 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.

20.19. Clearing Scrambling Keys on a Tamper Event

On a tamper detection event on WKUP pins, it is possible to perform an immediate clear of the scrambling keys (SMC_KEY1 and SMC_KEY2) if SMC_OCMS.TAMPCLR is set.

20.20. Register Summary

The SMC is programmed using the registers listed below. For each chip select, a set of four registers is used to program the parameters of the external device connected on it. The number of SMC_SETUP, SMC_PULSE, SMC_CYCLE and SMC_MODE registers depends on the number of chip selects. Sixteen bytes (0x10) are required per chip select.

Note: The user must confirm the SMC configuration by writing any one of the SMC_MODE registers.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SMC_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]					
0x04	SMC_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]						
0x08	SMC_CYCLE0	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x0C	SMC_MODE0	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x10	SMC_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]					
0x14	SMC_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]						
0x18	SMC_CYCLE1	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							
0x1C	SMC_MODE1	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x20	SMC_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]					
0x24	SMC_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]						
0x28	SMC_CYCLE2	31:24								NRD_CYCLE[8]
		23:16	NRD_CYCLE[7:0]							
		15:8								NWE_CYCLE[8]
		7:0	NWE_CYCLE[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x2C	SMC_MODE2	31:24			PS[1:0]					PMEN
		23:16				TDF_MODE	TDF_CYCLES[3:0]			
		15:8			DBW[1:0]					BAT
		7:0			EXNW_MODE[1:0]				WRITE_MODE	READ_MODE
0x30 ... 0x7F	Reserved									
0x80	SMC_OCMS	31:24								
		23:16								
		15:8						CS2xSE	CS1xSE	CS0xSE
		7:0				TAMPCLR				SMSE
0x84	SMC_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x88	SMC_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							
0x8C ... 0x8F	Reserved									
0x90	SMC_SRIER	31:24								
		23:16								
		15:8								
		7:0								SRIE
0x94 ... 0xE3	Reserved									
0xE4	SMC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	SMC_WPSR	31:24							SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE		WPVS

20.20.1. SMC Setup Register

Name: SMC_SETUPx
Offset: 0x00 + x*0x10 [x=0..2]
Reset: 0x01010101
Property: Read/Write

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

Note: The number of SMC_SETUP registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
			NCS_RD_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	23	22	21	20	19	18	17	16
			NRD_SETUP[5:0]					
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
			NCS_WR_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
			NWE_SETUP[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/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 in clock cycles 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

20.20.2. SMC Pulse Register

Name: SMC_PULSEx
Offset: 0x04 + x*0x10 [x=0..2]
Reset: 0x01010101
Property: Read/Write

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

Note: The number of SMC_PULSE registers depends on the chip select number.

Bit	31	30	29	28	27	26	25	24
		NCS_RD_PULSE[6:0]						
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
		NRD_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
		NCS_WR_PULSE[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

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

Bits 30:24 – NCS_RD_PULSE[6:0] NCS Pulse Length in READ Access
In standard read access, the NCS signal pulse length is defined as:
NCS pulse length = (256 * NCS_RD_PULSE[6] + NCS_RD_PULSE[5:0]) clock cycles
The NCS pulse length must be at least 1 clock cycle.
In Page mode read access, the NCS_RD_PULSE parameter defines the duration of the first access to one page.

Bits 22:16 – NRD_PULSE[6:0] NRD Pulse Length
In standard read access, the NRD signal pulse length is defined in clock cycles as:
NRD pulse length = (256 * NRD_PULSE[6] + NRD_PULSE[5:0]) clock cycles
The NRD pulse length must be at least 1 clock cycle.
In Page mode read access, the NRD_PULSE parameter defines the duration of the subsequent accesses in the page.

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 length must be at least 1 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 length must be at least 1 clock cycle.

20.20.3. SMC Cycle Register

Name: SMC_CYCLEx
Offset: 0x08 + x*0x10 [x=0..2]
Reset: 0x00030003
Property: Read/Write

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

Note: The number of SMC_CYCLE registers depends on the chip select number.

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

20.20.4. SMC Mode Register

Name: SMC_MODEx
Offset: 0x0C + x*0x10 [x=0..2]
Reset: 0x10001000
Property: Read/Write

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

The user must confirm the SMC configuration by writing any one of the SMC_MODE registers.

Note: The number of SMC_MODE registers depends on the chip select number.

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

Bit	23	22	21	20	19	18	17	16
				TDF_MODE	TDF_CYCLES[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
			DBW[1:0]					BAT
Access			R/W	R/W				R/W
Reset			0	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			0	0

Bits 29:28 – PS[1:0] Page Size
If Page mode is enabled, this field indicates the size of the page in bytes.

Value	Name	Description
0	BYTE_4	4-byte page
1	BYTE_8	8-byte page
2	BYTE_16	16-byte page
3	BYTE_32	32-byte page

Bit 24 – PMEN Page Mode Enabled

Value	Description
1	Asynchronous burst read in Page mode is applied on the corresponding chip select.
0	Standard read is applied.

Bit 20 – TDF_MODE TDF Optimization

Value	Description
1	TDF optimization enabled—The number of TDF Wait states is optimized using the setup period of the next read/write access.
0	TDF optimization disabled—The number of TDF Wait states is inserted before the next access begins.

Bits 19:16 – TDF_CYCLES[3:0] Data Float Time

This field gives the integer number of clock cycles required by the external device to release the data after the rising edge of the read controlling signal. The SMC always provides one full cycle of bus turnaround after the TDF_CYCLES period. The external bus cannot be used by another chip select during TDF_CYCLES + 1 cycles. From 0 up to 15 TDF_CYCLES can be set.

Bits 13:12 – DBW[1:0] Data Bus Width

Value	Name	Description
0	BIT_8	8-bit bus
1	BIT_16	16-bit bus
2	—	Reserved
3	—	Reserved

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 of the 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 Mode—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	Write operation controlled by NCS signal—If TDF optimization is enabled (TDF_MODE = 1), TDF Wait states will be inserted after the setup of NCS.
1	NWE_CTRL	Write operation controlled by NWE signal—If TDF optimization is enabled (TDF_MODE = 1), TDF Wait states will be inserted after the setup of NWE.

Bit 0 – READ_MODE Selection of the Control Signal for Read Operation

Value	Name	Description
0	NCS_CTRL	Read operation controlled by NCS signal • If TDF cycles are programmed, the external bus is marked busy after the rising edge of NCS. • If TDF optimization is enabled (TDF_MODE = 1), TDF Wait states are inserted after the setup of NCS.

Value	Name	Description
1	NRD_CTRL	Read operation controlled by NRD signal • If TDF cycles are programmed, the external bus is marked busy after the rising edge of NRD. • If TDF optimization is enabled (TDF_MODE = 1), TDF Wait states are inserted after the setup of NRD.

20.20.5. SMC Off-Chip Memory Scrambling Register

Name: SMC_OCMS
Offset: 0x80
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
						CS2xSE	CS1xSE	CS0xSE
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 8, 9, 10 – CSxSE Chip Select (x = 0 to 2) Scrambling Enable

Value	Description
0	Disables scrambling for CSx.
1	Enables scrambling for CSx.

Bit 4 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on SMC scrambling keys.
1	A tamper detection event immediately clears SMC scrambling keys.

Bit 0 – SMSE Static Memory Controller Scrambling Enable

Value	Description
0	Disables scrambling for SMC access.
1	Enables scrambling for SMC access.

20.20.6. SMC Off-Chip Memory Scrambling Key1 Register

Name: SMC_KEY1
Offset: 0x84
Reset: 0x00000000
Property: Write-only

This register is a 'Write-once' register: the first write access after a system reset prevents any further modification of the register value.

This register is erased if a tamper is detected on fast wake-up pins and bit SMC_OCMS.TAMPCLR = 1.

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
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
	KEY1[23:16]							
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
	KEY1[15:8]							
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
	KEY1[7:0]							
Access	W	W	W	W	W	W	W	W
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, KEY1 and KEY2 values determine data scrambling.

20.20.7. SMC Off-Chip Memory Scrambling Key2 Register

Name: SMC_KEY2
Offset: 0x88
Reset: 0x00000000
Property: Write-only

This register is a 'Write-once' register: the first write access after a system reset prevents any further modification of the register value.

This register is erased if a tamper is detected on fast wake-up pins and bit SMC_OCMS.TAMPCLR = 1.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
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
	KEY2[23:16]							
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
	KEY2[15:8]							
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
	KEY2[7:0]							
Access	W	W	W	W	W	W	W	W
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, KEY1 and KEY2 values determine data scrambling.

20.20.8. SMC Safety Report Interrupt Enable Register

Name: SMC_SRIER
Offset: 0x90
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
Access								
Reset								

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

Bit 0 – SRIE Safety Report Interrupt Enable

Value	Description
0	Disables the SMC safety report interrupt from External Bus Interface.
1	Enables the SMC safety report interrupt from External Bus Interface.

20.20.9. SMC Write Protection Mode Register

Name: SMC_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
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).

20.20.10.SMC Write Protection Status Register

Name: SMC_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
	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		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.
1	WRITE_WO	A write access has been performed on a read-only register.
2	UNDEF_RW	Access to an undefined address.

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 SMC_WPSR.
1	A software error has occurred since the last read of SMC_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 internal sequencer error has occurred since the last read of SMC_WPSR.
1	An internal sequencer error has occurred since the last read of SMC_WPSR. This flag can only be set under abnormal operating conditions.

Bit 0 – WPVS Write Protection Violation Status

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

21. Programmable Multibit Error Correction Code Controller (PMECC)

21.1. Description

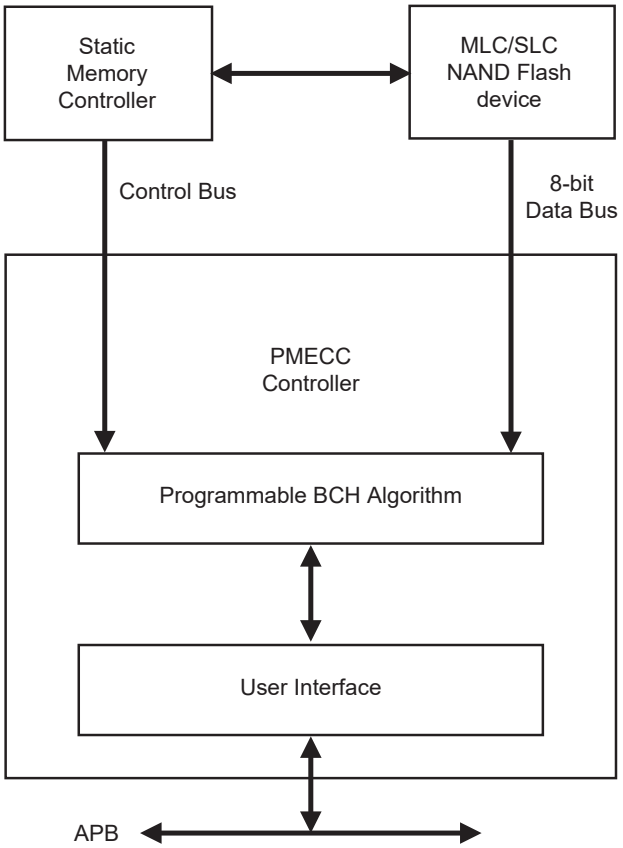
The Programmable Multibit Error Correction Code Controller (PMECC) is a programmable binary BCH (Bose, Chaudhuri and Hocquenghem) encoder/decoder. This controller can be used to generate redundancy information for both Single-Level Cell (SLC) and Multi-level Cell (MLC) NAND Flash devices. It supports redundancy for correction of 2, 4, 8, 12 or 24 bits of error per sector of data.

21.2. Embedded Characteristics

- 8-bit NAND Flash Data Bus Support
- Multibit Error Correcting Code
- Algorithm Based on Binary Shortened Bose, Chaudhuri and Hocquenghem (BCH) Codes
- Programmable Error Correcting Capability: 2, 4, 8, 12 and 24 Bits of Error per Sector
- Programmable Sector Size: 512 Bytes or 1024 Bytes
- Programmable Number of Sectors per Page: 1, 2, 4 or 8 Sectors of Data per Page
- Programmable Spare Area Size
- Supports Spare Area ECC Protection
- Supports 8 Kbytes Page Size Using 1024 Bytes per Sector and 4 Kbytes Page Size Using 512 Bytes per Sector
- Configurable through APB Interface
- Interrupt-Driven Multibit Error Detection

21.3. Block Diagram

Figure 21.1. PMECC Block Diagram



21.4. Functional Description

The NAND Flash 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 Table 21.1). This operation is executed for each sector. At decoding time, the PMECC module generates the remainder of the received codeword by minimal polynomials. When all polynomial remainders for a given sector are set to zero, no error occurred. When the polynomial remainders are other than zero, the codeword is corrupted and further processing is required.

The PMECC generates an interrupt indicating that an error occurred. The processor must read the Interrupt Status register (PMECC_ISR). This register indicates which sector is corrupted.

To find the error location within a sector, the processor must execute the following decoding steps:

1. Syndrome computation
2. Find the error locator polynomials
3. Find the roots of the error locator polynomial

All decoding steps involve finite field computation, for which a library of finite field arithmetic must be available to perform addition, multiplication and inversion. The finite field 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. Assume beta = alpha ^ index, then beta belongs to the field, and gf_log(beta) = gf_log(alpha ^ index) = index. The gf_antilog tables provide exponent inverse of the element, if beta = alpha ^ index, then gf_antilog(index) = beta.

The first step consists of the syndrome computation. The PMECC computes the remainders and software must substitute the power of the primitive element.

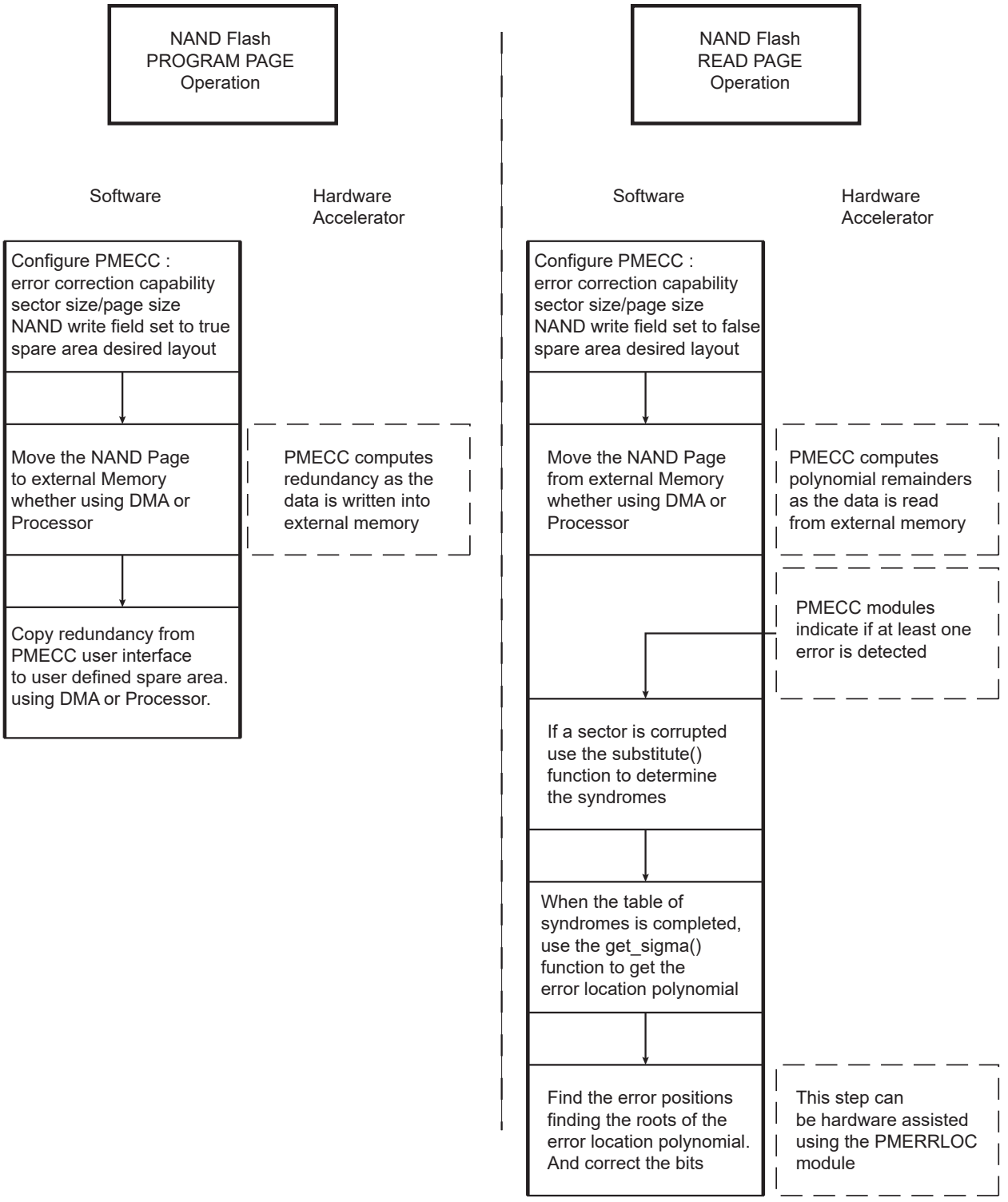
The procedure implementation is described in Remainder Substitution Procedure.

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 described in Find 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, as there is no straightforward method of finding the roots, except by 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 Programmable Multibit Error Correction Code Location (PMERRLOC) module provides this type of hardware acceleration.

Figure 21.2. Software/Hardware Multibit Error Correction Dataflow



21.4.1. MLC/SLC Write Page Operation using PMECC

When an MLC write page operation is performed, the PMECC controller is configured with the NANDWR bit in the Configuration register (PMECC_CFG) set to one. When the NAND spare area contains file system information and redundancy (PMECCx), the spare area is error protected,

then PMECC_CFG.SPAREEN is set to '1'. When the NAND spare area contains only redundancy information, SPAREEN is set to '0'.

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 21.1. Relevant Redundancy Registers

BCH_ERR Field	Sector Size Set to 512 Bytes	Sector Size Set to 1024 Bytes
0	PMECC_ECC0	PMECC_ECC0
1	PMECC_ECC0, PMECC_ECC1	PMECC_ECC0, PMECC_ECC1
2	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3
3	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6
4	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6, PMECC_ECC7, PMECC_ECC8, PMECC_ECC9	PMECC_ECC0, PMECC_ECC1, PMECC_ECC2, PMECC_ECC3, PMECC_ECC4, PMECC_ECC5, PMECC_ECC6, PMECC_ECC7, PMECC_ECC8, PMECC_ECC9, PMECC_ECC10

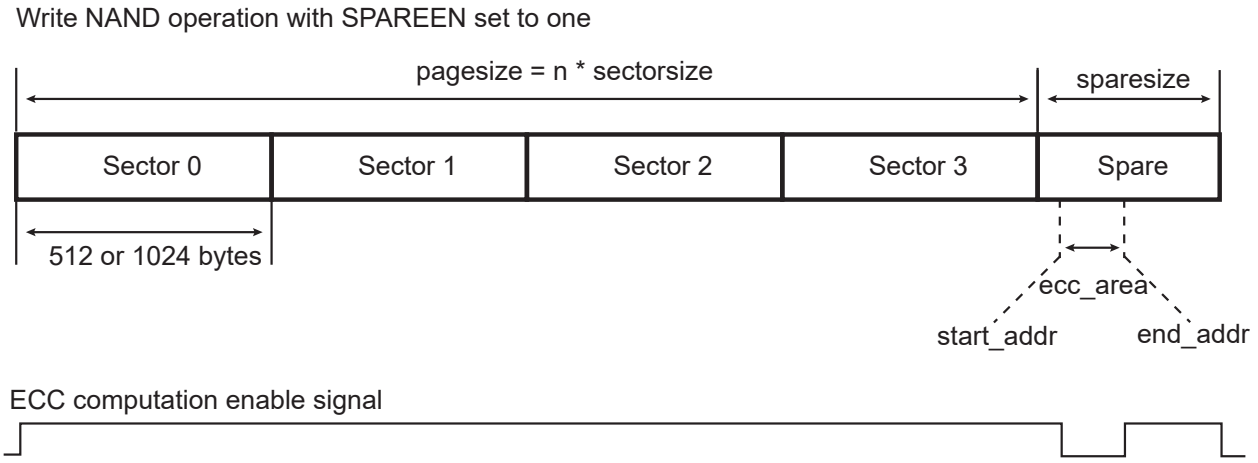
Table 21.2. Number of Relevant ECC Bytes per Sector, Copied from LSbyte to MSbyte

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

21.4.1.1. SLC/MLC Write Operation with Spare Enable Bit Set

When PMECC_CFG.SPAREEN is set to '1', the spare area of the page is encoded with the stream of data of the last sector of the page. This mode is entered by writing one in the DATA bit in the Control register (PMECC_CTRL). When the encoding process is over, the redundancy is written to the spare area in user mode, PMECC_CTRL.USER must be set to '1'.

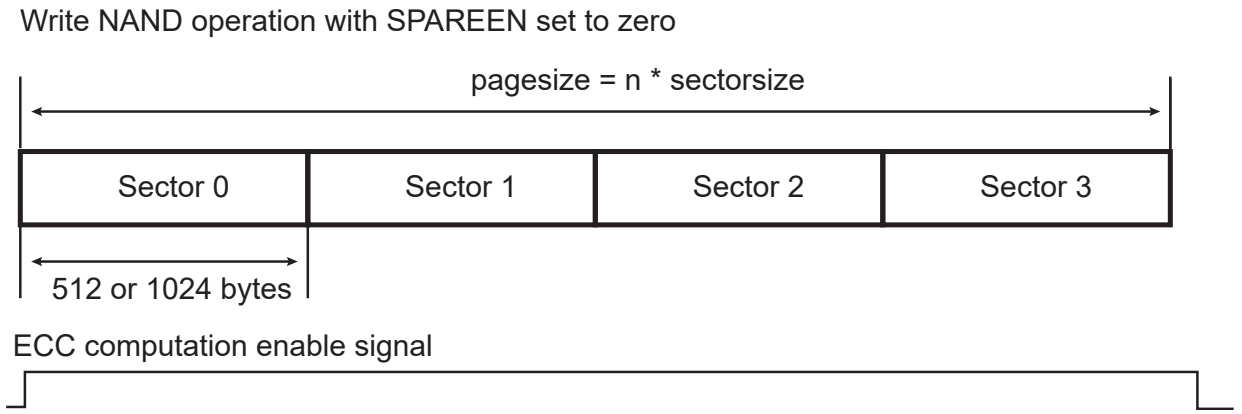
Figure 21.3. NAND Write Operation with Spare Encoding



21.4.1.2. MLC/SLC Write Operation with Spare Area Disabled

When PMECC_CFG.SPAREEN is set to '0', the spare area is not encoded with the stream of data. This mode is entered by writing '1'to PMECC_CTRL.DATA.

Figure 21.4. NAND Write Operation



21.4.2. MLC/SLC Read Page Operation using PMECC

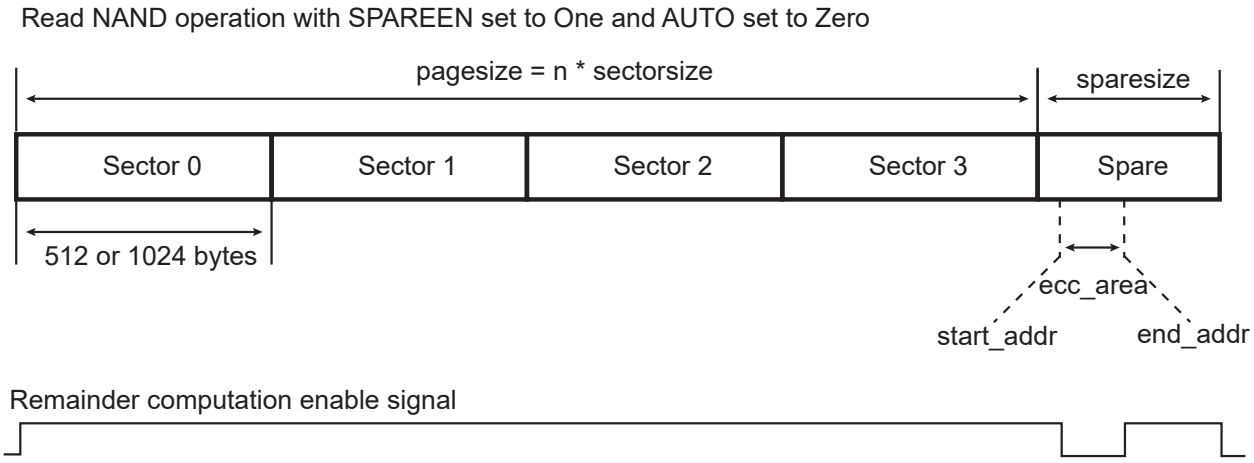
Table 21.3. Relevant Remainders Registers

BCH_ERR Field	Sector Size Set to 512 Bytes	Sector Size Set to 1024 Bytes
0	PMECC_REM0	PMECC_REM0
1	PMECC_REM0, PMECC_REM1	PMECC_REM0, PMECC_REM1
2	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3,	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3
3	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7
4	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7, PMECC_REM8, PMECC_REM9, PMECC_REM10, PMECC_REM11	PMECC_REM0, PMECC_REM1, PMECC_REM2, PMECC_REM3, PMECC_REM4, PMECC_REM5, PMECC_REM6, PMECC_REM7, PMECC_REM8, PMECC_REM9, PMECC_REM10, PMECC_REM11

21.4.2.1. MLC/SLC Read Operation with Spare Decoding

When the spare area is protected, the spare area contains valid data. As the redundancy may be included in the middle of the information stream, the user programs the start address and the end address of the ECC area. The controller will automatically skip the ECC area. This mode is entered by writing a '1' to PMECC_CTRL.DATA. When the page has been fully retrieved from NAND, the ECC area is read using the user mode by writing a '1' to PMECC_CTRL.USER.

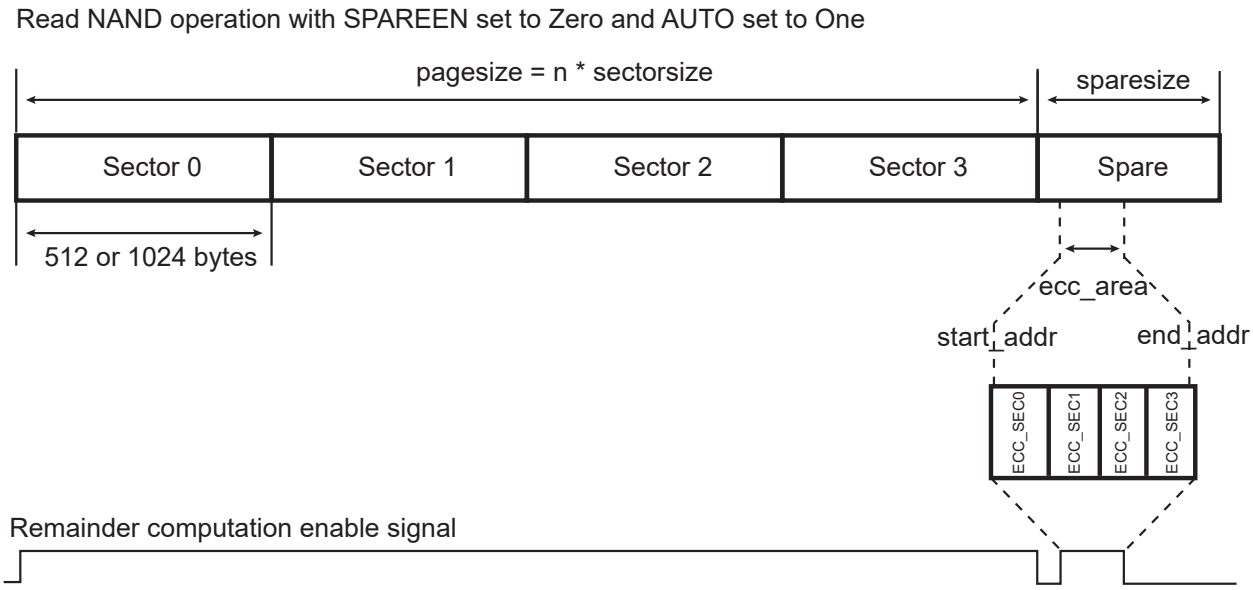
Figure 21.5. Read Operation with Spare Decoding



21.4.2.2. MLC/SLC Read Operation

If the spare area is not protected with the error correcting code, the redundancy area is retrieved directly. This mode is entered by writing a '1' to PMECC_CTRL.DATA. When PMECC_CFG.AUTO is set to '1', the ECC is retrieved automatically, otherwise the ECC must be read using user mode.

Figure 21.6. Read Operation

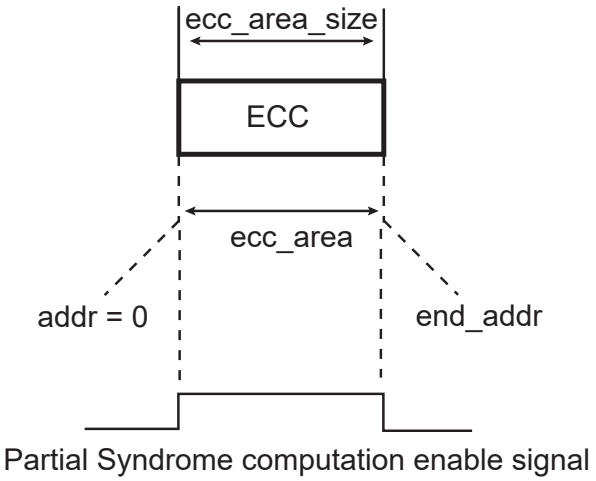


21.4.2.3. MLC/SLC User Read ECC Area

This mode allows a manual retrieve of the ECC.

This mode is entered by writing a '1' to PMECC_CTRL.USER.

Figure 21.7. User Read Mode



21.5. Software Implementation

21.5.1. Remainder Substitution Procedure

The substitute function evaluates the polynomial remainder, with different values of the field primitive elements. The finite field arithmetic addition operation is performed with the Exclusive or. The finite field arithmetic multiplication operation is performed through the gf_log, gf_antilog lookup tables.

The REM2NP1 and REMN2NP3 fields of the PMECC Remainder x registers (PMECC_REMx) contain only odd remainders. Each bit indicates whether the coefficient of the polynomial remainder 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;
    }
```

21.5.2. Find 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;
}
```

21.5.3. Find the Error Position

The output of the get_sigma() procedure is a polynomial stored in the smu[NB_ERROR+1][] table. The error position is the roots of that polynomial. The degree of this polynomial is very important information, as it gives the number of errors. The PMERRLOC module provides a hardware accelerator for this step.

21.6. Register Summary

Notes: The blocks of registers listed below are instanced 8 times in the user interface:

- PMECC_ECCx[x=0..10]
- PMECC_REMx[x=0..11]

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PMECC_CFG	31:24								
		23:16				AUTO				SPAREEN
		15:8				NANDWR			PAGESIZE[1:0]	
		7:0				SECTORSZ		BCH_ERR[2:0]		
0x04	PMECC_SAREA	31:24								
		23:16								
		15:8								SPARESIZE[8]
		7:0	SPARESIZE[7:0]							
0x08	PMECC_SADDR	31:24								
		23:16								
		15:8								STARTADDR[8]
		7:0	STARTADDR[7:0]							
0x0C	PMECC_EADDR	31:24								
		23:16								
		15:8								ENDADDR[8]
		7:0	ENDADDR[7:0]							
0x10	PMECC_CLK	31:24								
		23:16								
		15:8								
		7:0						CLKCTRL[2:0]		
0x14	PMECC_CTRL	31:24								
		23:16								
		15:8								
		7:0			DISABLE	ENABLE		USER	DATA	RST
0x18	PMECC_SR	31:24								
		23:16								
		15:8								
		7:0				ENABLE				BUSY
0x1C	PMECC_IER	31:24								
		23:16								
		15:8								
		7:0								ERRIE
0x20	PMECC_IDR	31:24								
		23:16								
		15:8								
		7:0								ERRID
0x24	PMECC_IMR	31:24								
		23:16								
		15:8								
		7:0								ERRIM
0x28	PMECC_ISR	31:24								
		23:16								
		15:8								
		7:0	ERRIS[7:0]							
0x2C ... 0x3F	Reserved									
0x40	PMECC_ECC0	31:24	ECC[31:24]							
		23:16	ECC[23:16]							
		15:8	ECC[15:8]							
		7:0	ECC[7:0]							
...										

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x68	PMECC_ECC10	31:24	ECC[31:24]							
		23:16	ECC[23:16]							
		15:8	ECC[15:8]							
		7:0	ECC[7:0]							
0x6C ... 0x023F	Reserved									
0x0240	PMECC_REM0	31:24			REM2NP3[13:8]					
		23:16	REM2NP3[7:0]							
		15:8			REM2NP1[13:8]					
		7:0	REM2NP1[7:0]							
...										
0x026C	PMECC_REM11	31:24			REM2NP3[13:8]					
		23:16	REM2NP3[7:0]							
		15:8			REM2NP1[13:8]					
		7:0	REM2NP1[7:0]							

21.6.1. PMECC Configuration Register

Name: PMECC_CFG
Offset: 0x000
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 spare enable is activated.

Value	Description
0	Indicates that the spare area is not protected. In that case the ECC computation takes into account the ECC area located in the spare area. (within the start address and the end address).
1	Indicates that the spare is error protected. In this case, the ECC computation takes into account the whole spare area minus the ECC area in the ECC computation operation.

Bit 16 – SPAREEN Spare Enable

- For NAND write access:
  — 0: The spare area is skipped
  — 1: The spare area is protected with the last sector of data.
- For NAND read access:
  — 0: The spare area is skipped.
  — 1: The spare area contains protected data or only redundancy information.

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)

Value	Name	Description
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 errors 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.
1	The ECC computation is based on a sector of 1024 bytes.

Bits 2:0 – BCH_ERR[2:0] Error Correct 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

21.6.2. PMECC Spare Area Size Register

Name: PMECC_SAREA
Offset: 0x004
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								R/W
Reset								0

Bit	7	6	5	4	3	2	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 8:0 – SPARESIZE[8:0] Spare Area Size
The spare area size is equal to (SPARESIZE+1) bytes.

21.6.3. PMECC Start Address Register

Name: PMECC_SADDR
Offset: 0x008
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 (byte oriented address)
This field indicates the first byte address of the ECC area. Location 0 matches the first byte of the spare area.

21.6.4. PMECC End Address Register

Name: PMECC_EADDR
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								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 (byte oriented address)
This field indicates the last byte address of the ECC area.

21.6.5. PMECC Clock Control Register

Name: PMECC_CLK
Offset: 0x010
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
						CLKCTRL[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – CLKCTRL[2:0] Clock Control Register
The PMECC datapath setup time is set to CLKCTRL+1.
This field indicates the database setup times in number of clock cycles. At 133 MHz, this field must be programmed with 2, indicating that the setup time is 3 clock cycles.

21.6.6. PMECC Control Register

Name: PMECC_CTRL
Offset: 0x014
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 Module Disable
The PMECC module must always be configured after being deactivated.

Bit 4 – ENABLE PMECC Module Enable
The PMECC module must always be configured before being activated.

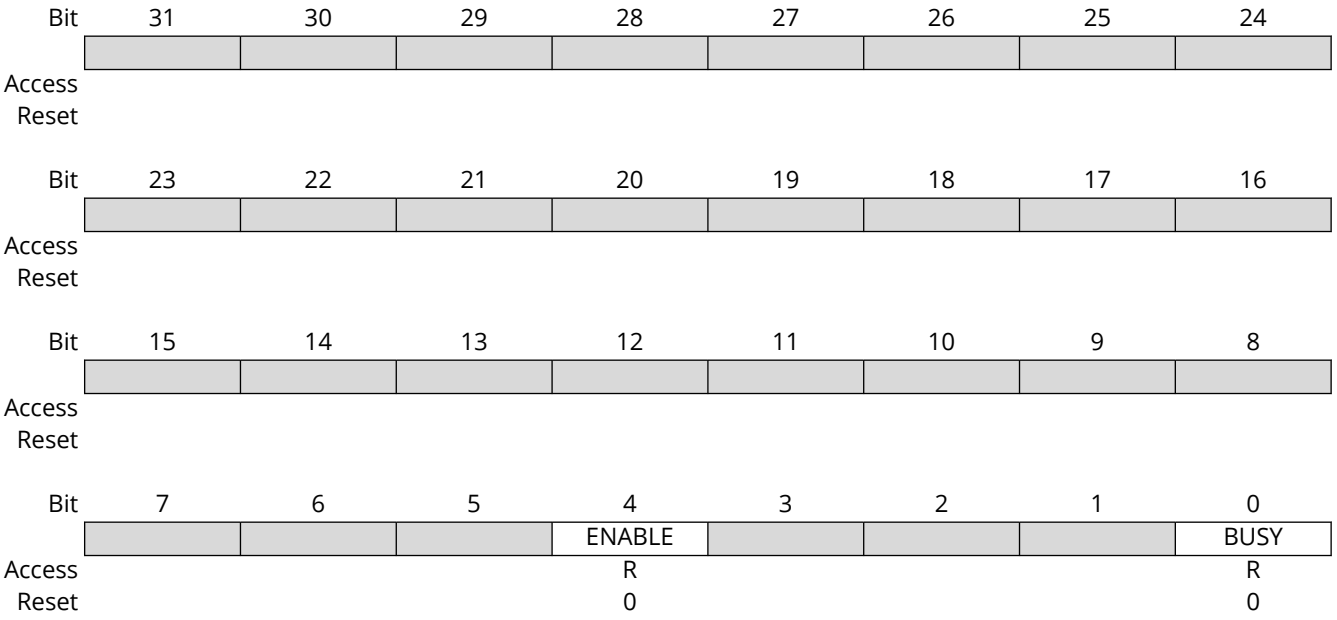
Bit 2 – USER Start a User Mode Phase

Bit 1 – DATA Start a Data Phase

Bit 0 – RST Reset the PMECC Module
When set to one, this bit resets the PMECC Controller; configuration registers remain unaffected.

21.6.7. PMECC Status Register

Name: PMECC_SR
Offset: 0x018
Reset: 0x00000000
Property: Read-only



Bit 4 – ENABLE PMECC Module Status

Value	Description
0	The PMECC module is disabled and can be configured.
1	The PMECC module is enabled and the configuration registers cannot be written.

Bit 0 – BUSY Kernel of the PMECC is Busy

21.6.8. PMECC Interrupt Enable Register

Name: PMECC_IER
Offset: 0x01C
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
								ERRIE
Access								W
Reset								–

Bit 0 – ERRIE Error Interrupt Enable

21.6.9. PMECC Interrupt Disable Register

Name: PMECC_IDR
Offset: 0x020
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
								ERRID
Access								W
Reset								–

Bit 0 – ERRID Error Interrupt Disable

21.6.10. PMECC Interrupt Mask Register

Name: PMECC_IMR
Offset: 0x024
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
								ERRIM
Access								R
Reset								0

Bit 0 – ERRIM Error Interrupt Mask

21.6.11. PMECC Interrupt Status Register

Name: PMECC_ISR
Offset: 0x028
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
When set to one, bit i of the PMECC_ISR indicates that sector i is corrupted.

21.6.12. PMECC ECC x Register

Name: PMECC_ECCx
Offset: 0x40 + x*0x04 [x=0..10]
Reset: 0xFFFFFFFF
Property: Read-only

Note: The block of registers PMECC_ECCx[x=0..10] 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	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	ECC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
	ECC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

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

Bits 31:0 – ECC[31:0] BCH Redundancy
This register contains the remainder of the division of the codeword by the generator polynomial.

21.6.13. PMECC Remainder x Register

Name: PMECC_REMx
Offset: 0x0240 + x*0x04 [x=0..11]
Reset: 0x00000000
Property: Read-only

Note: The block of registers PMECC_REMx[x=0..11] 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.

22. Programmable Multibit ECC Error Location Controller (PMERRLOC)

22.1. Description

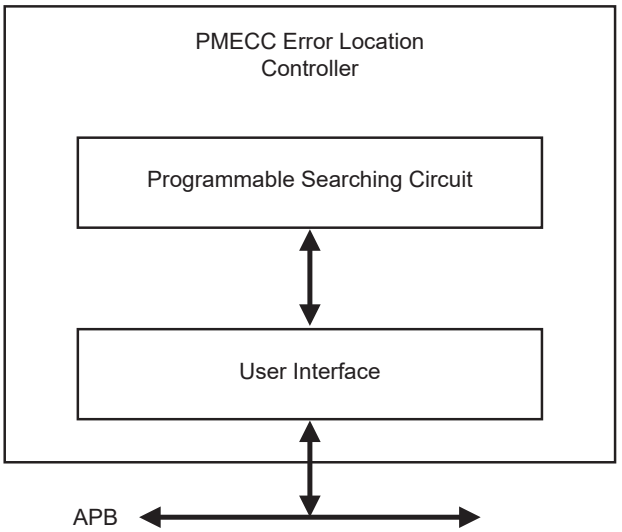
The PMECC Error Location Controller provides hardware acceleration for determining roots of polynomials over two finite fields: GF(2^13) and GF(2^14). It integrates 24 fully programmable coefficients. These coefficients belong to GF(2^13) or GF(2^14). The coefficient programmed in the PMERRLOC_SIGMAx register is the coefficient of degree x in the polynomial.

22.2. Embedded Characteristics

- Provides Hardware Acceleration to Determine 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

22.3. Block Diagram

Figure 22.1. Block Diagram



22.4. Functional Description

The PMERRLOC search operation is started as soon as a write access is detected in the PMERRLOC_ELEN register and can be disabled by writing to the PMERRLOC_ELDIS register. PMERRLOC_ELEN.ENINIT shall be initialized with the number of Galois field elements to test. The set of the roots can be limited to a valid range.

Table 22.1. 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

Table 22.2. ENINIT Field Value for a Sector Size of 1024 Bytes

Error Correcting Capability	ENINIT Value
2	8220
4	8248
8	8304
12	8360
24	8528

When the PMERRLOC is searching for roots, PMERRLOC_ELSR.BUSY remains asserted. An interrupt is asserted at the end of the computation, and PMERRLOC_ELISR.DONE is set. PMERRLOC_ELISR.ERR_CNT indicates the number of errors. The error position can be read in the PMERRLOC_ELx registers.

22.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PMERRLOC_ELCFG	31:24								
		23:16				ERRNUM[4:0]				
		15:8								
		7:0								SECTORSZ
0x04	PMERRLOC_ELPRIM	31:24								
		23:16								
		15:8	PRIMITIV[15:8]							
		7:0	PRIMITIV[7:0]							
0x08	PMERRLOC_ELEN	31:24								
		23:16								
		15:8			ENINIT[13:8]					
		7:0	ENINIT[7:0]							
0x0C	PMERRLOC_ELDIS	31:24								
		23:16								
		15:8								
		7:0								DIS
0x10	PMERRLOC_ELSR	31:24								
		23:16								
		15:8								
		7:0								BUSY
0x14	PMERRLOC_ELIER	31:24								
		23:16								
		15:8								
		7:0								DONE
0x18	PMERRLOC_ELIDR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x1C	PMERRLOC_ELIMR	31:24								
		23:16								
		15:8								
		7:0								DONE
0x20	PMERRLOC_ELISR	31:24								
		23:16								
		15:8				ERR_CNT[4:0]				
		7:0								DONE
0x24 ... 0x27	Reserved									
0x28	PMERRLOC_SIGMA0	31:24								
		23:16								
		15:8			SIGMA0[13:8]					
		7:0	SIGMA0[7:0]							
0x2C	PMERRLOC_SIGMA1	31:24								
		23:16								
		15:8			SIGMA[13:8]					
		7:0	SIGMA[7:0]							
...										
0x88	PMERRLOC_SIGMA24	31:24								
		23:16								
		15:8			SIGMA[13:8]					
		7:0	SIGMA[7:0]							
0x8C	PMERRLOC_EL0	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							
...										

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE8	PMERRLOC_EL23	31:24								
		23:16								
		15:8			ERRLOCN[13:8]					
		7:0	ERRLOCN[7:0]							

22.5.1. PMERRLOC Configuration Register

Name: PMERRLOC_ELCFG
Offset: 0x000
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.

22.5.2. PMERRLOC Primitive Register

Name: PMERRLOC_ELPRIM
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
	PRIMITIV[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
	PRIMITIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PRIMITIV[15:0] Primitive Polynomial

22.5.3. PMERRLOC Enable Register

Name: PMERRLOC_ELEN
Offset: 0x008
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
			ENINIT[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
	ENINIT[7:0]							
Access	R/W	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 – ENINIT[13:0] Initial Number of Bits in the Codeword

22.5.4. PMERRLOC Disable Register

Name: PMERRLOC_ELDIS
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
								DIS
Access								R/W
Reset								0

Bit 0 – DIS Disable Error Location Engine

22.5.5. PMERRLOC Status Register

Name: PMERRLOC_ELSR
Offset: 0x010
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
								BUSY
Access								R/W
Reset								0

Bit 0 – BUSY Error Location Engine Busy

22.5.6. PMERRLOC Interrupt Enable Register

Name: PMERRLOC_ELIER
Offset: 0x014
Reset: 0x00000000
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								0

Bit 0 – DONE Computation Terminated Interrupt Enable

22.5.7. PMERRLOC Interrupt Disable Register

Name: PMERRLOC_ELIDR
Offset: 0x018
Reset: 0x00000000
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								

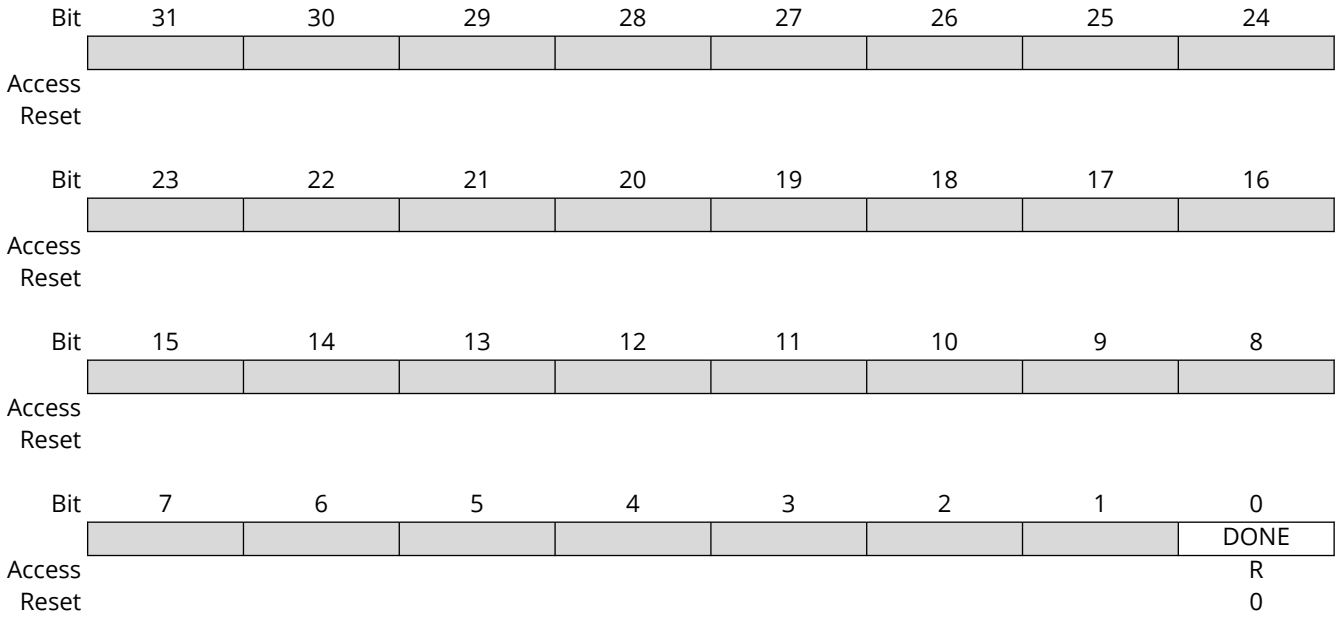
Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								DONE
Access								W
Reset								0

Bit 0 – DONE Computation Terminated Interrupt Disable

22.5.8. PMERRLOC Interrupt Mask Register

Name: PMERRLOC_ELIMR
Offset: 0x01C
Reset: 0x00000000
Property: Read-only



Bit 0 – DONE Computation Terminated Interrupt Mask

22.5.9. PMERRLOC Interrupt Status Register

Name: PMERRLOC_ELISR
Offset: 0x020
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[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:8 – ERR_CNT[4:0] Error Counter Value

Bit 0 – DONE Computation Terminated Interrupt Status

22.5.10. PMERRLOC SIGMA0 Register

Name: PMERRLOC_SIGMA0
Offset: 0x028
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.

22.5.11. PMERRLOC SIGMAx Register

Name: PMERRLOC_SIGMAx
Offset: 0x2C + (x-1)*0x04 [x=1..24]
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								

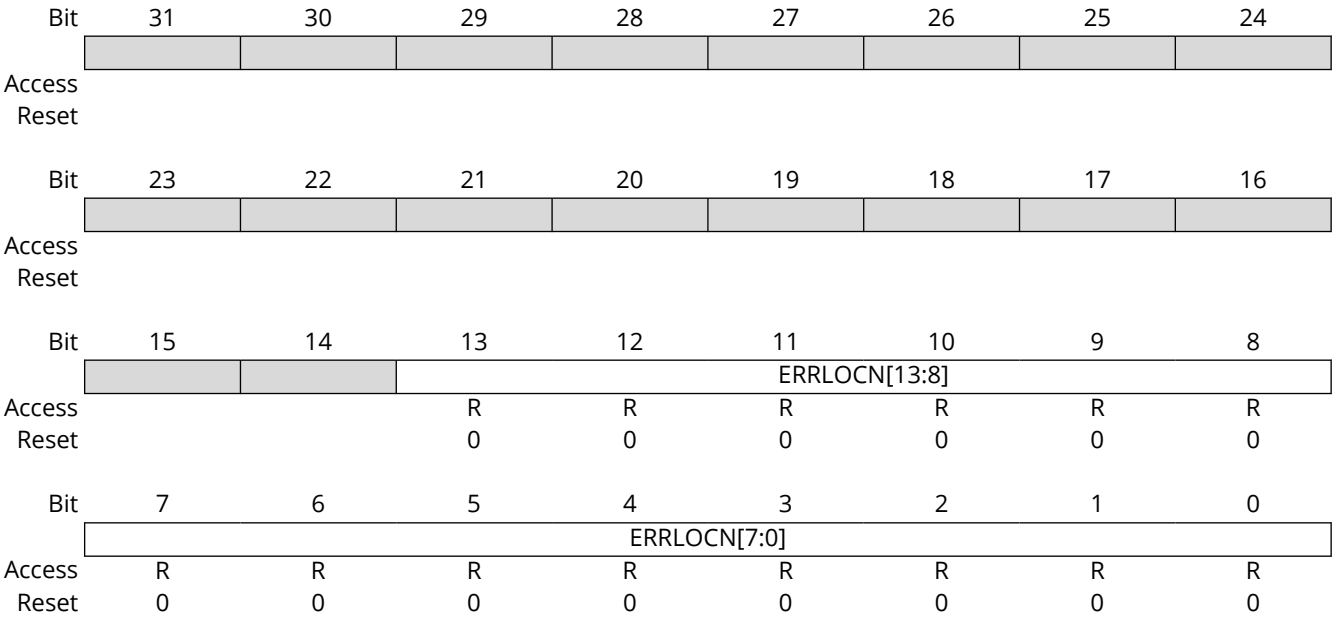
Bit	15	14	13	12	11	10	9	8
			SIGMA[13:8]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	SIGMA[7:0]							
Access	R/W	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 – SIGMA[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.

22.5.12. PMERRLOC Error Location x Register

Name: PMERRLOC_ELx
Offset: 0x8C + x*0x04 [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
			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 0 when the first bit of the main area is corrupted.
If the sector size is set to 512 bytes, ERRLOCN points to 4096 when the last bit of the sector area is corrupted.
If the sector size is set to 1024 bytes, ERRLOCN points to 8192 when the last bit of the sector area is corrupted.
If the sector size is set to 512 bytes, ERRLOCN points to 4097 when the first bit of the spare area is corrupted.
If the sector size is set to 1024 bytes, ERRLOCN points to 8193 when the first bit of the spare area is corrupted.

23. DDR-SDRAM Controller (MPDDRC)

23.1. Description

DDR-SDRAM Controller (MPDDRC) maximizes memory bandwidth and minimizes transaction latency due to the DDR-SDRAM protocol.

The MPDDRC extends the memory capabilities of a chip by providing the interface to the external 16-bit DDR-SDRAM device. The page size supports ranges from 2048 to 16384 rows and from 512 to 4096 columns. It supports word (32-bit), half-word (16-bit), and byte (8-bit) accesses.

The MPDDRC supports a read or write burst length of eight locations. This enables the command and address bus to anticipate the next command, thus reducing latency imposed by the DDR-SDRAM protocol and improving the DDR-SDRAM bandwidth. Moreover, MPDDRC keeps track of the active row in each bank, thus maximizing DDR-SDRAM performance, e.g., the application may be placed in one bank and data in other banks. To optimize performance, avoid accessing different rows in the same bank. The MPDDRC supports a CAS latency of 3, 4, 5 or 6 and optimizes the read access depending on the frequency.

Self-Refresh and Power-Down modes minimize the consumption of the DDR-SDRAM device.

OCD (Off-chip Driver) and ODT (On-die Termination) modes, and Write Leveling, are not supported.

The MPDDRC supports DDR3-SDRAM and DDR3L-SDRAM devices with DLL disabled (DLL Off) and DLL enabled (DLL On).

In DLL Off mode, as per the applicable JEDEC standard, the maximum clock frequency is 125 MHz. However, check with memory suppliers for support for higher speeds. The DLL Off mode sets the CAS Read Latency (CRL) and the CAS Write Latency (CWL) to 6.

In DLL On mode, as per the applicable JEDEC standard, the minimum clock frequency is 303 MHz. However, check with memory suppliers for support for lower speeds. The DLL On mode sets the CAS Read Latency (CRL) to 6 and the CAS Write Latency (CWL) to 5. DDR3-SDRAM supports high-capacity (1 Gbit and greater) and can reduce power consumption with a 1.5V supply (DDR3-SDRAM) or a 1.35V supply (DDR3L-SDRAM).

23.2. Embedded Characteristics

- Supported Memory Devices:
  - DDR2-SDRAM
  - DDR3-SDRAM (DLL Off/On mode)
  - DDR3L-SDRAM (DLL Off/On mode)
- Arbitration Policies: Round-Robin, On Request, Bandwidth, Quality of Service
- Microchip Technology Non-Blocking Bus Interface Improves Quality of Service
- Multiple Outstanding Bus Transactions Increase Bandwidth (up to 30 %)
- 7 System Bus Interfaces; Management of all Accesses Maximizes Memory Bandwidth and Minimizes Transaction Latency
- Bus Transfer: Word, Half Word, Byte Access
- Supported Configurations:
  - 2K, 4K, 8K, 16K row address memory parts
  - DDR-SDRAM with four or eight internal banks (DDR2-SDRAM/DDR3-SDRAM/DDR3L-SDRAM)
  - DDR-SDRAM with 16-bit data path for system-oriented word access
  - One chip select for SDRAM device (256-Mbyte address space)
- Programming Facilities

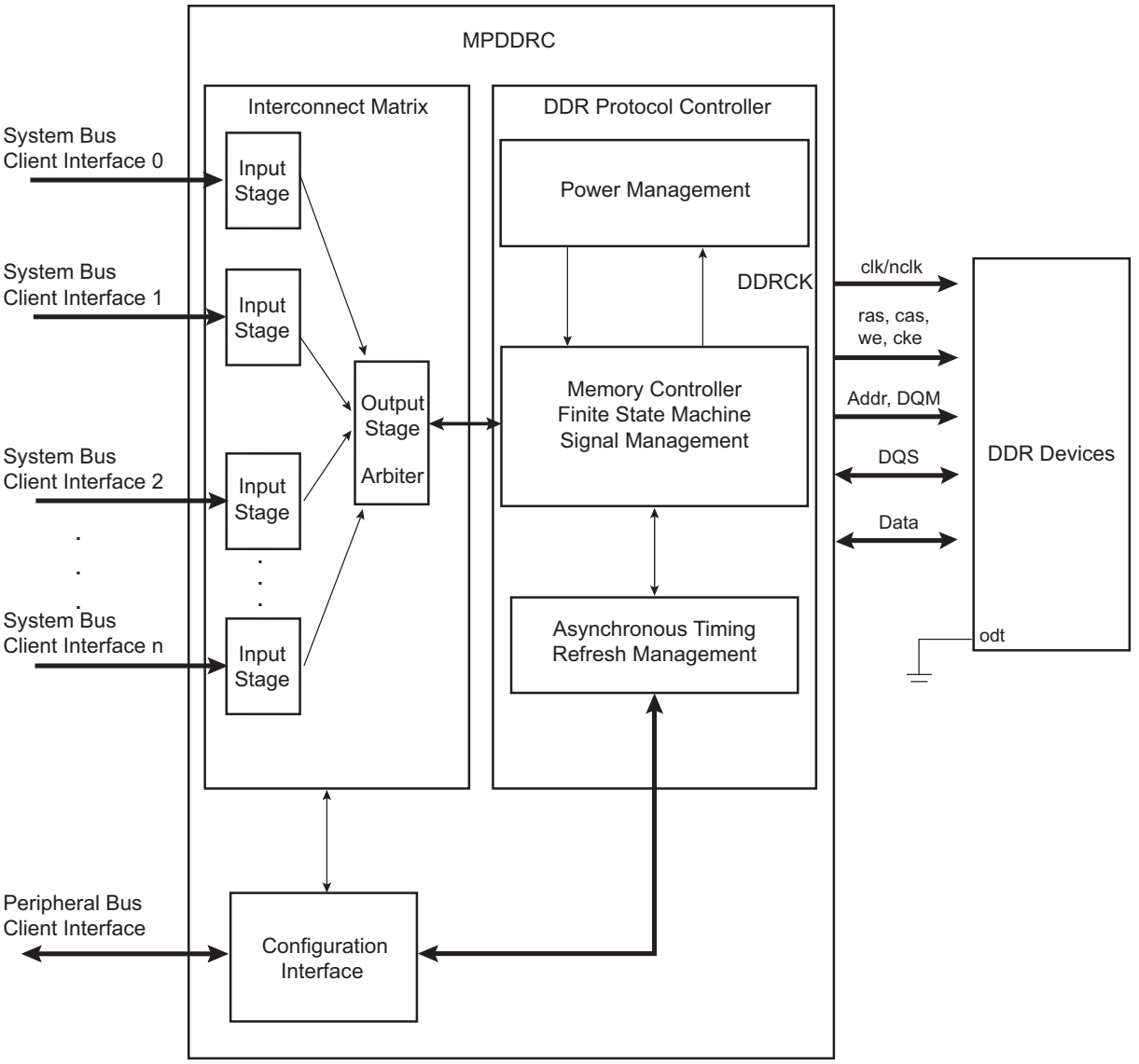
  - Multibank ping-pong access (up to four or eight banks opened at the same time = reduced average latency of transactions)
  - Timing parameters specified by software
  - Automatic refresh operation, refresh rate is programmable
- Energy-Saving Capabilities
  - Self-Refresh, Power-Down and Active Power-Down modes supported
- DDR-SDRAM Power-Up Initialization by Software
- CAS Latency of 3, 4, 5 or 6 Supported
- Reset Function Supported (DDR2-SDRAM)
- Clock Frequency Change in Self-Refresh Mode Supported (/DDR2-SDRAM/DDR3-SDRAM/DDR3L-SDRAM)
- Auto-precharge Command Not Used
- OCD (Off-chip Driver) Mode, ODT (On-die Termination), Write leveling are Not Supported
- Abnormal Software Access and Sequencer Integrity Error Reports
- Dynamic Scrambling with User Key (No Impact on Bandwidth)
- Bus Monitor

23.3. Block Diagram

The MPDDRC is partitioned in two blocks (see figure below):

- An Interconnect Matrix block that manages concurrent accesses on the system bus between 7 system bus hosts and integrates an arbiter
- A DDR Controller that translates system bus requests (read/write) in the DDR-SDRAM protocol

Figure 23.1. Block Diagram



23.4. Product Dependencies, Initialization Sequence

23.4.1. Interrupt Sources

The MPDDRC interrupt line is connected on one of the internal sources of the interrupt controller. Using the MPDDRC interrupt requires the interrupt controller to be programmed first.

23.4.2. Performance Optimization

To improve Quality of Service and increase the bandwidth, enable the MPDDRC Multiport feature (SFR_CCFG_EBICSA.DDR_MP_EN = 1) and set the type of arbitration to Quality of Service policy (MPDDRC_CONF_ARBITER.ARB = 3).

23.4.3. DDR2-SDRAM Initialization

Set the DDR I/O calibration codes prior to initializing the DDR-SDRAM in SFR_CAL1 register. Write 0x0000_0194 for DDR2-SDRAM.

The initialization sequence is generated by software. The DDR2-SDRAM devices are initialized by the following sequence:

1. Program the memory device type in the Memory Device register (MPDDRC_MD).
2. Program the shift sampling value in the Read Data Path register (MPDDRC_RD_DATA_PATH).
3. Program MPDDRC_IO_CALIBR.
4. Program features of the DDR2-SDRAM device in the Configuration register (MPDDRC_CR) (number of columns, rows, banks, CAS latency and output driver impedance control) and in the Timing Parameter 0 register/Timing Parameter 1 register (MPDDRC_TPR0/1) (asynchronous timing: TRC, TRAS, etc.).
5. A NOP command is issued to the DDR2-SDRAM. Program the NOP command in the Mode register (MPDDRC_MR). The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command. The clocks which drive the DDR2-SDRAM device are now enabled.
6. A pause of at least 200 µs must be observed before a signal toggle.
7. A NOP command is issued to the DDR2-SDRAM. Program the NOP command in the MPDDRC_MR. The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command. CKE is now driven high.
8. An All Banks Precharge command is issued to the DDR2-SDRAM. Program All Banks Precharge command in the MPDDRC_MR. The application must configure the MODE field to 2 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
9. An Extended Mode Register Set (EMRS2) cycle is issued to choose between commercial or high temperature operations. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00800000.
   Note: This address is given as an example only. The real address depends on implementation in the product.
10. An Extended Mode Register Set (EMRS3) cycle is issued to set the Extended Mode register to 0. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 1 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00C00000
11. An Extended Mode Register Set (EMRS1) cycle is issued to enable DLL and to program D.I.C. (Output Driver Impedance Control). The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
12. An additional 200 cycles of clock are required for locking DLL.
13. Write a ‘1’ to the DLL bit (enable DLL reset) in the Configuration register (MPDDRC_CR).
14. A Mode Register Set (MRS) cycle is issued to reset DLL. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this

command. The write address must be chosen so that signals BA[1:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.

15. An All Banks Precharge command is issued to the DDR2-SDRAM. Program the All Banks Precharge command in the MPDDRC_MR. The application must configure the MODE field to 2 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
16. Two auto-refresh (CBR) cycles are provided. Program the Auto-refresh command (CBR) in the MPDDRC_MR. The application must configure the MODE field to 4 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM location twice to acknowledge these commands. TRFC must be checked between two auto-refreshes (see MPDDRC_TPR1).
17. Write a '0' to the DLL bit (disable DLL reset) in the MPDDRC_CR.
18. A Mode Register Set (MRS) cycle is issued to program parameters of the DDR2-SDRAM device, in particular CAS latency and to disable DLL reset. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[1:0] are set to 0. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
19. Configure the OCD field (default OCD calibration) to 7 in the MPDDRC_CR.
20. An Extended Mode Register Set (EMRS1) cycle is issued to the default OCD value. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks), the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
21. Configure the OCD field (exit OCD calibration mode) to 0 in the MPDDRC_CR.
22. An Extended Mode Register Set (EMRS1) cycle is issued to enable OCD exit. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR2-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 128-Mbit, DDR2-SDRAM (12 rows, 9 columns, 4 banks) bank address, the DDR2-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x00400000.
23. A Normal Mode command is provided. Program the Normal mode in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR2-SDRAM address to acknowledge this command.
24. Write the refresh rate into the COUNT field in the Refresh Timer register (MPDDRC_RTR). To compute the value, see MPDDRC Refresh Timer Register.

After initialization, the DDR2-SDRAM devices are fully functional.

23.4.4. DDR3-SDRAM/DDR3L-SDRAM Initialization

Set the DDR I/O calibration codes prior to initializing the DDR-SDRAM in register SFR_CAL1. Write 0x0000_0126 for DDR3(L)-SDRAM.

The initialization sequence is generated by software. The DDR3-SDRAM devices are initialized by the following sequence:

1. Program the memory device type in the Memory Device register (MPDDRC_MD).

2. Program the shift sampling value in the Read Data Path register (MPDDRC_RD_DATA_PATH).
3. Program MPDDRC_DDR3_CAL, MPDDRC_DDR3_TIM_CAL, MPDDRC_IO_CALIBR.
4. Set all registers depending on the mode selected (DLL On or DLL Off).
5. Program the features of the DDR3-SDRAM device in the Configuration register (MPDDRC_CR) (number of columns, rows, banks, CAS latency and output driver impedance control, supply value) and in the Timing Parameter 0 register/Timing Parameter 1 register (MPDDRC_TPR0/1) (asynchronous timing - TRC, TRAS, etc.).
6. A NOP command is issued to the DDR3-SDRAM. Program the NOP command in the Mode register (MPDDRC_MR). The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR3-SDRAM address to acknowledge this command. The clocks which drive the DDR3-SDRAM device are now enabled.
7. A pause of at least 500 µs must be observed before a signal toggle.
8. A NOP command is issued to the DDR3-SDRAM. Program the NOP command in the MPDDRC_MR. The application must configure the MODE field to 1 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR3-SDRAM address to acknowledge this command. CKE is now driven high.
9. An Extended Mode Register Set (EMRS2) cycle is issued to choose between commercial or high temperature operations. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR3-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[2] is set to 0, BA[1] is set to 1 and signal BA[0] is set to 0. For example: with a 16-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the DDR3-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x040000000; with a 32-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x08000000.
   Note: This address is given as an example only. The real address depends on the implementation in the product.
10. An Extended Mode Register Set (EMRS3) cycle is issued to set the Extended Mode register to 0. The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR3-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[2] is set to 0, BA[1] is set to 1 and signal BA[0] is set to 1. For example: with a 16-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the DDR3-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x06000000; with a 32-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x0C000000.
11. Set MPDDRC_CR.DIS_DLL (Disable DLL) to 1 in DLL Off mode, or to 0 in DLL On mode.
12. An Extended Mode Register Set (EMRS1) cycle is issued to disable and to program ODS (output drive strength). The application must configure the MODE field to 5 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR3-SDRAM to acknowledge this command. The write address must be chosen so that signal BA[2:1] is set to 0 and signal BA[0] is set to 1. For example: with a 16-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the DDR3-SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x02000000; with a 32-bit, 1-Gbit, DDR3-SDRAM (14 rows, 10 columns, 8 banks), the SDRAM write access should be done at the address: BASE_ADDRESS_DDR + 0x04000000.
13. Write a ‘1’ to the DLL bit (enable DLL reset) in the Configuration register (MPDDRC_CR).
14. A Mode Register Set (MRS) cycle is issued to reset DLL. The application must configure the MODE field to 3 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler

instruction just after the read. Perform a write access to the DDR3-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[2:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.

15. A Calibration command (MRS) is issued to calibrate RTT and RON values for the Process Voltage Temperature (PVT). The application must configure the MODE field to 6 in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to the DDR3-SDRAM to acknowledge this command. The write address must be chosen so that signals BA[2:0] are set to 0. For example, the SDRAM write access should be done at the address: BASE_ADDRESS_DDR.
16. A Normal Mode command is provided. Program the Normal mode in the MPDDRC_MR. Read the MPDDRC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any DDR3-SDRAM address to acknowledge this command.
17. Write the refresh rate into the COUNT field in the Refresh Timer register (MPDDRC_RTR). To compute the value, see MPDDRC Refresh Timer Register.

After initialization, the DDR3-SDRAM devices are fully functional.

23.5. Functional Description

23.5.1. DDR-SDRAM Controller Write Cycle

The MPDDRC provides burst access or single access in Normal mode (MPDDRC_MR.MODE = 0). Whatever the access type, the MPDDRC keeps track of the active row in each bank, thus maximizing performance.

The DDR-SDRAM device is programmed with a burst length (bl) equal to 8. This determines the length of a sequential data input by the write command that is set to 8. The latency from write command to data input depends on the memory type, as shown in the following table.

Table 23.1. CAS Write Latency

Memory Devices	CAS Write Latency (CWL)
DDR2-SDRAM	2/3/4
DDR3-SDRAM (DLL On)	6
DDR3-SDRAM (DLL Off)	5

To initiate a single access, the MPDDRC checks if the page access is already open. If row/bank addresses match with the previous row/bank addresses, the controller generates a write command. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a write command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/write (t_{RCD}) commands. As the burst length is set to 8, in case of single access, it has to stop the burst, otherwise seven invalid values may be written. In case of the DDR-SDRAM device, the Burst Stop command is not supported for the burst write operation. So, in order to interrupt the write operation, the DM (data mask) input signal must be set to 1 to mask invalid data (see the following figures), and DQS must continue to toggle.

To initiate a burst access, the MPDDRC uses the transfer type signal provided by the host requesting the access. If the next access is a sequential write access, writing to the DDR-SDRAM device is carried out. If the next access is a write non-sequential access, then an automatic access break is inserted, the MPDDRC generates a precharge command, activates the new row and initiates a write command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/write (t_{RCD}) commands.

For the definition of timing parameters, see MPDDRC Timing Parameter 0 Register.

Write accesses to the DDR-SDRAM device are burst oriented and the burst length is programmed to 8. It determines the maximum number of column locations that can be accessed for a given write

command. When the write command is issued, eight columns are selected. All accesses for that burst take place within these eight columns, thus the burst wraps within these eight columns if a boundary is reached. These eight columns are selected by addr[13:3]. addr[2:0] is used to select the starting location within the block.

In case of incrementing burst (INCR/INCR4/INCR8/INCR16), the addresses can cross the 16-byte boundary of the DDR-SDRAM device. For example, when a transfer (INCR4) starts at address 0x0C, the next access is 0x10, but since the burst length is programmed to 8, the next access is at 0x00. Since the boundary is reached, the burst is wrapped. The MPDDRC takes this feature of the DDR-SDRAM device into account. In case of a transfer starting at address 0x04/0x08/0x0C or starting at address 0x10/0x14/0x18/0x1C, two write commands are issued to avoid wrapping when the boundary is reached. The last write command is subject to DM input logic level. If DM is registered high, the corresponding data input is ignored and the write access is not done. This avoids additional writing.

Figure 23.2. Single Write Access, Row Closed, DDR2-SDRAM Devices

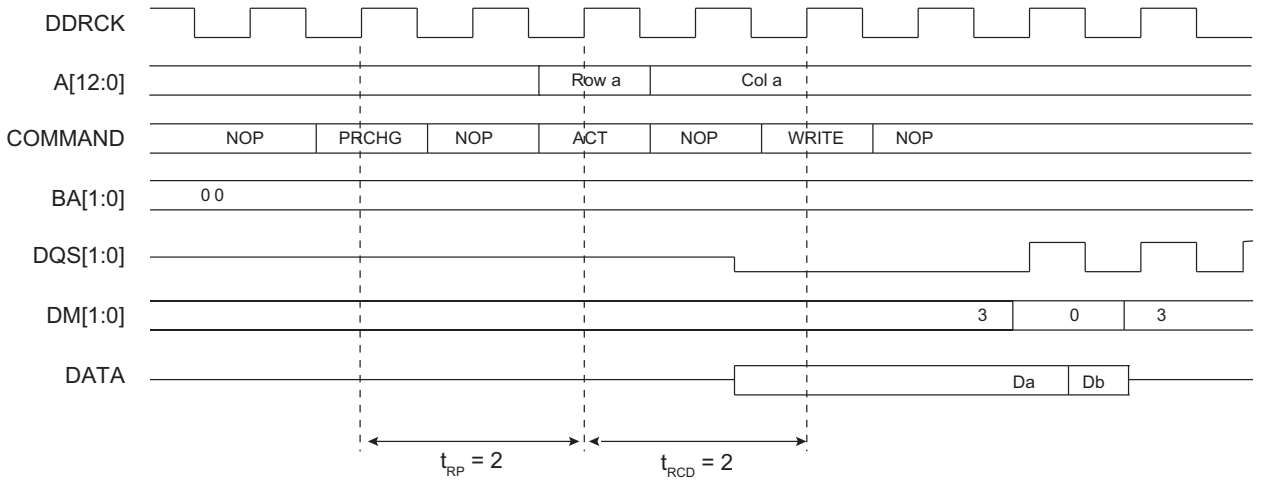
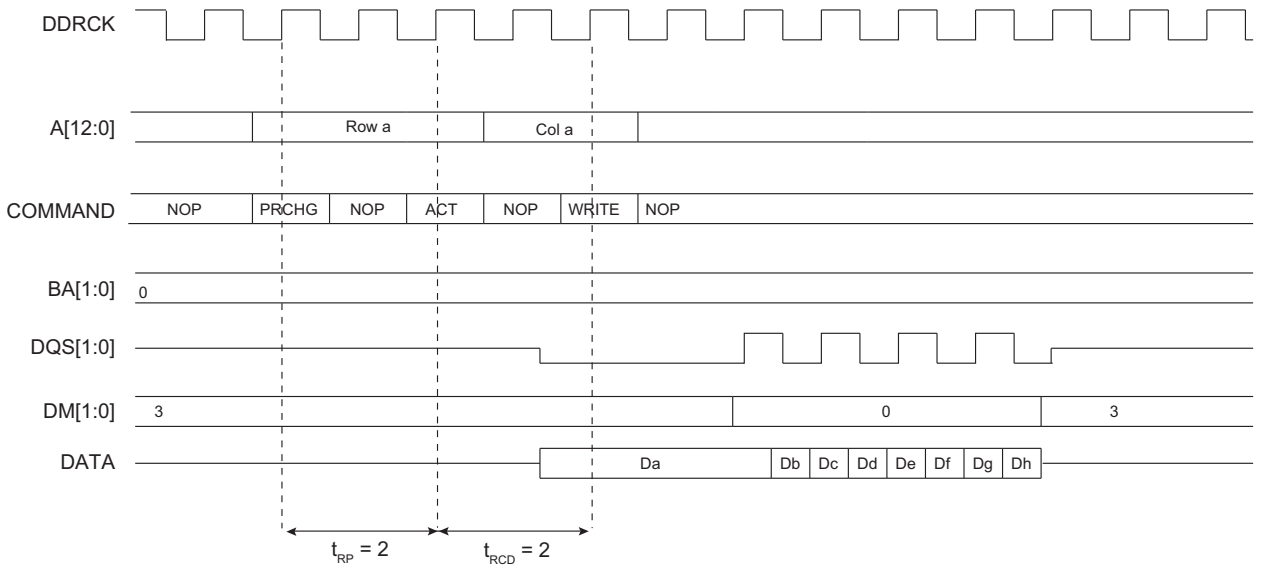
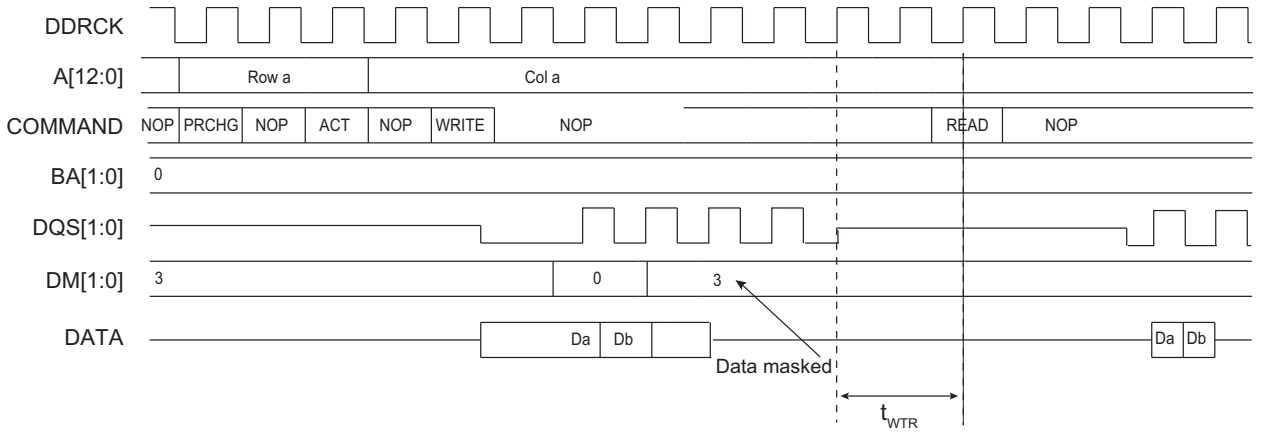


Figure 23.3. Burst Write Access, Row Closed, DDR2-SDRAM Devices



A write command can be followed by a read command. To avoid breaking the current write burst, t_{WTR}/t_{WRD} (bl/2 + 2 = 6 cycles) should be met. See the figure below.

Figure 23.4. Single Write Access Followed by a Read Access, DDR2-SDRAM Devices



23.5.2. DDR-SDRAM Controller Read Cycle

The MPDDRC provides burst access or single access in Normal mode (MPDDRC_MR.MODE = 0). Whatever the access type, the MPDDRC keeps track of the active row in each bank, thus maximizing performance of the MPDDRC.

The DDR-SDRAM devices are programmed with a burst length equal to 8 which determines the length of a sequential data output by the read command that is set to 8. The latency from read command to data output depends on the memory type, as shown in the following table. This value is programmed during the initialization phase (see Product Dependencies, Initialization Sequence).

Table 23.2. CAS Read Latency

Memory Devices	CAS Read Latency
DDR2-SDRAM	3/4
DDR3-SDRAM (DLL Off/DLL On)	5/6

To initiate a single access, the MPDDRC checks if the page access is already open. If row/bank addresses match with the previous row/bank addresses, the controller generates a read command. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a read command. To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/read (t_{RCD}) commands. After a read command, additional wait states are generated to comply with CAS latency. The MPDDRC supports a CAS latency delay of 3, 4, 5 or 6 clock cycles.

To initiate a burst access, the MPDDRC checks the transfer type signal. If the next accesses are sequential read accesses, reading to the SDRAM device is carried out. If the next access is a read non-sequential access, then an automatic page break can be inserted. If the bank addresses are not identical or if bank addresses are identical but the row addresses are not identical, the controller generates a precharge command, activates the new row and initiates a read command. If page access is already open, a read command is generated.

To comply with DDR-SDRAM timing parameters, additional clock cycles are inserted between precharge/active (t_{RP}) commands and active/read (t_{RCD}) commands. The MPDDRC supports a CAS latency delay of 3, 4, 5 or 6 clock cycles. During this delay, the controller uses internal signals to anticipate the next access and improve the performance of the controller. Depending on the latency, the MPDDRC anticipates 3, 4, 5 or 6 read accesses. In case of burst of specified length, accesses are not anticipated, but if the burst is broken (border, Busy mode, etc.), the next access is treated as an incrementing burst of unspecified length, and depending on the latency, the MPDDRC anticipates 3, 4, 5 or 6 read accesses.

For the definition of timing parameters, see MPDDRC Configuration Register.

Read accesses to the DDR-SDRAM are burst oriented and the burst length is programmed to 8. The burst length determines the maximum number of column locations that can be accessed for a given read command. When the read command is issued, eight columns are selected. All accesses for that burst take place within these eight columns, meaning that the burst wraps within these eight columns if the boundary is reached. These eight columns are selected by addr[13:3]; addr[2:0] is used to select the starting location within the block.

In case of incrementing burst (INCR/INCR4/INCR8/INCR16), the addresses can cross the 16-byte boundary of the DDR-SDRAM device. For example, when a transfer (INCR4) starts at address 0x0C, the next access is 0x10, but since the burst length is programmed to 8, the next access is 0x00. Since the boundary is reached, the burst wraps. The MPDDRC takes into account this feature of the SDRAM device. In case of the DDR-SDRAM device, transfers start at address 0x04/0x08/0x0C. Two read commands are issued to avoid wrapping when the boundary is reached. The last read command may generate additional reading (1 read cmd = 4 DDR words).

Figure 23.5. Single Read Access, Row Closed, Latency = 3, DDR2-SDRAM Devices

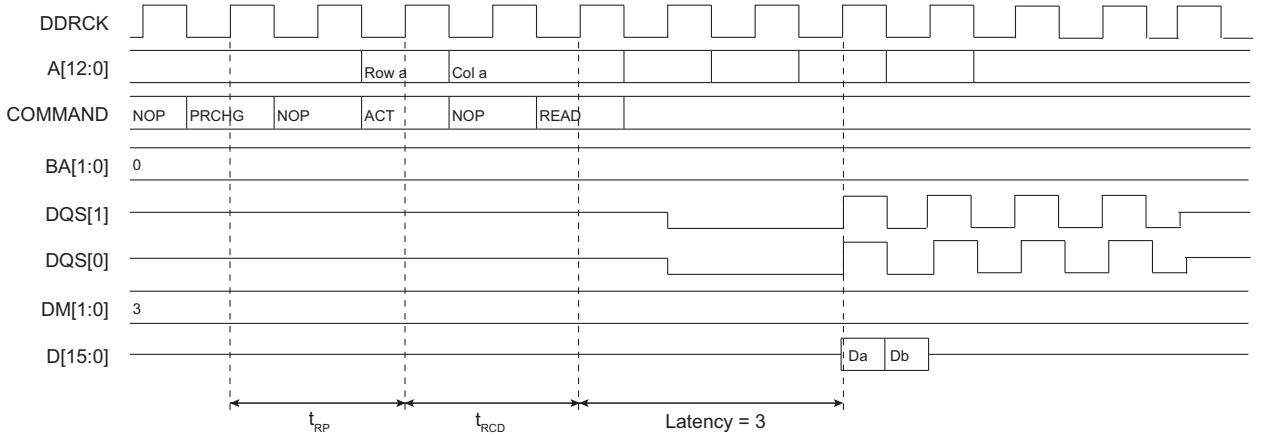
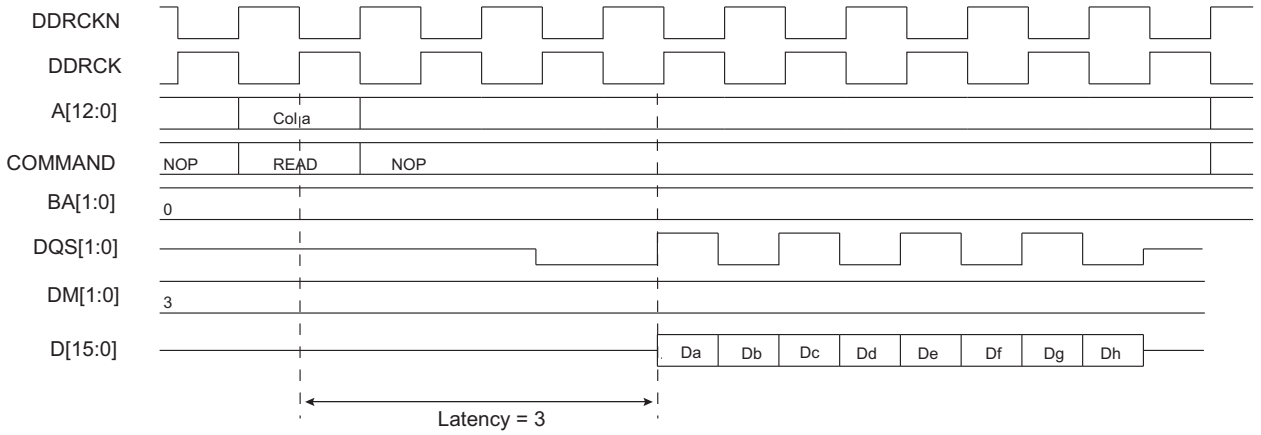


Figure 23.6. Burst Read Access, Latency = 3, DDR2-SDRAM Devices



23.5.3. Refresh (Auto-Refresh Command)

An Auto-refresh command is used to refresh the external SDRAM devices. Refresh addresses are generated internally by the DDR-SDRAM device and incremented automatically after each auto-refresh. The MPDDRC generates these auto-refresh commands periodically. A timer is loaded in the MPDDRC_RTR with the value which indicates the number of clock cycles between refresh cycles (see MPDDRC Refresh Timer Register). When the MPDDRC initiates a refresh of the DDR-SDRAM device,

internal memory accesses are not delayed. However, if the CPU tries to access the DDR-SDRAM device, the client indicates that the device is busy. A refresh request does not interrupt a burst transfer in progress.

23.5.4. Power Management

23.5.4.1. Self-Refresh Mode

This mode is activated by configuring the Low-power Command bit (LPCB) to 1 in the MPDDRC Low-Power Register (MPDDRC_LPR).

Self-refresh mode is used in Power-down mode, that is, when no access to the DDR-SDRAM device is possible. In this case, power consumption is very low. In Self-refresh mode, the DDR-SDRAM device retains data without external clocking and provides its own internal clocking, thus performing its own auto-refresh cycles. During the self-refresh period, CKE is driven low. As soon as the DDR-SDRAM device is selected, the MPDDRC provides a sequence of commands and exits Self-refresh mode.

The MPDDRC re-enables Self-refresh mode as soon as the DDR-SDRAM device is not selected. It is possible to define when Self-refresh mode is to be enabled by configuring the TIMEOUT field in the MPDDRC_LPR:

0: Self-refresh mode is enabled as soon as the DDR-SDRAM device is not selected.

1: Self-refresh mode is enabled 64 clock cycles after completion of the last access.

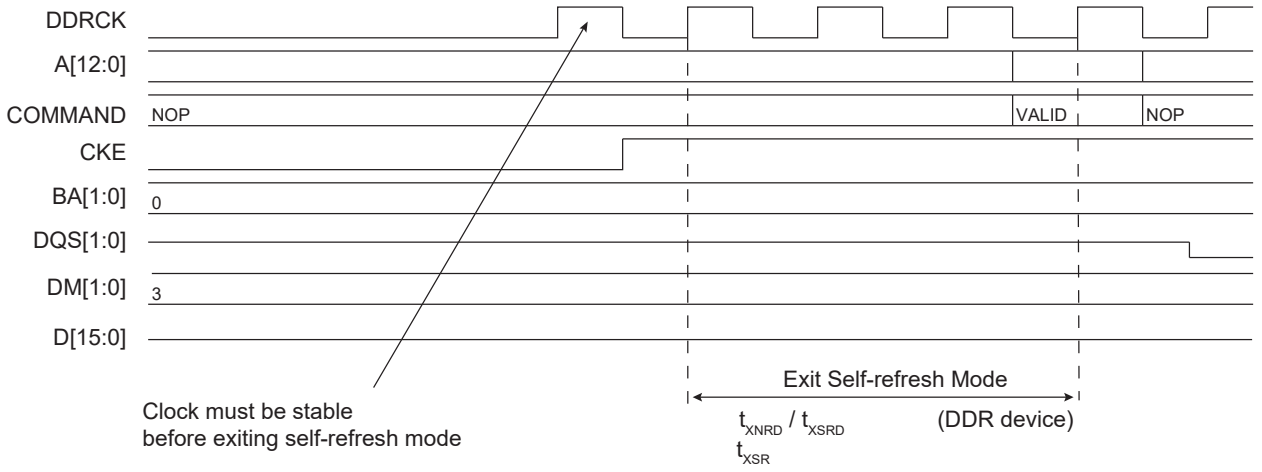
2: Self-refresh mode is enabled 128 clock cycles after completion of the last access.

Disabled banks are not refreshed in Self-refresh mode. This feature permits to reduce the self-refresh current.

The DDR2-SDRAM must remain in Self-refresh mode during the minimum of t_{CKE} periods (refer to the memory device data sheet), and may remain in Self-refresh mode for an indefinite period.

The DDR3-SDRAM must remain in Self-refresh mode for the minimum of t_{CKESR} periods (refer to the memory device data sheet) and may remain in Self-refresh mode for an indefinite period.

Figure 23.7. Self-Refresh Mode Exit



23.5.4.2. Power-Down Mode

This mode is activated by configuring the Low-power Command bit (LPCB) to 2 in the MPDDRC Low-Power Register (MPDDRC_LPR).

Power-down mode is used when no access to the DDR-SDRAM device is possible. In this mode, power consumption is greater than in Self-refresh mode. This state is similar to Normal mode (no Low-power mode/no Self-refresh mode), but the CKE pin is low and the input and output buffers are deactivated as soon the DDR-SDRAM device is no longer accessible. In contrast to Self-refresh mode, the DDR-SDRAM device cannot remain in Low-power mode longer than one refresh period (64 ms/32 ms). As no auto-refresh operations are performed in this mode, the MPDDRC carries out the refresh operation. For DDR-SDRAM devices, a NOP command must be generated for a minimum period defined in the TXP field of MPDDRC_TPR1 (see MPDDRC Timing Parameter 1 Register) and in the TXARD and TXARDS fields of MPDDRC_TPR2 (see MPDDRC Timing Parameter 2 Register) for DDR2_SDRAM devices.

The exit procedure is faster than in Self-refresh mode. See the following figure. The MPDDRC returns to Power-down mode as soon as the DDR-SDRAM device is not selected. It is possible to define when Power-down mode is enabled by configuring the TIMEOUT field in the MPDDRC_LPR:

0: Power-down mode is enabled as soon as the DDR-SDRAM device is not selected.

1: Power-down mode is enabled 64 clock cycles after completion of the last access.

2: Power-down mode is enabled 128 clock cycles after completion of the last access.

23.5.4.3. Change Frequency During Self-Refresh Mode with DDR2-SDRAM and DDR3-SDRAM Devices

To change frequency, Self-refresh mode must be activated. This is done by configuring the Low-power Command bit (LPCB) to 1 and writing a '1' to the Change Frequency Command bit (CHG_FR) in the Low-power register (MPDDRC_LPR).

Once the DDR-SDRAM device is in Self-refresh mode, the user must make sure there is no access in progress. Then, the user can change the clock frequency. The device input clock frequency changes only within minimum and maximum operating frequencies as specified by the DDR2-SDRAM and DDR3-SDRAM providers. The MPDDRC_RTR (COUNT), MPDDRC_CR (CAS, etc.), MPDDRC_TPR0 (TRC, TRP, TRAS, etc.), MPDDRC_TPR1 (TRFC, TXSNR, TXSRD, TXP) and MPDDRC_TPR2 (TRTP, TFAW, etc.) registers can be updated according to the new clock frequency. Once the input clock frequency is changed, new stable clocks must be provided to the device before exiting from Self-refresh mode.

To exit from Self-refresh mode, the DDR-SDRAM device must be selected. The MPDDRC provides a sequence of commands and exits Self-refresh mode.

During a change frequency procedure, MPDDRC_LPR.CHG_FR is set to 0 automatically.

Before changing frequency, make sure the processor clock (PCK) value is twice the system bus clock (MCK) value.

23.5.4.4. DDR3 Device DLL On/Off Switching Procedure

The JEDEC standard allows switching between DLL Off and DLL On modes, and hence beetwen low and high frequencies.

To enable DDR3 DLL Off mode, set MPDDRC_CR.DIS_DLL to 1. To enable DDR3 DLL On mode, set MPDDRC_CR.DIS_DLL to 0 during DDR3-SDRAM initialization.

23.5.4.4.1. DLL On to DLL Off Procedure

To switch from DLL On mode to DLL Off mode, follow the procedure below:

1. Make sure no access is in progress and Self-refresh mode is disabled.
2. Set MPDDRC_CR.DIS_DLL to 1.
3. Enable Self-refresh mode and set MPDDRC_LPR.UPD_MR to 2.
4. Wait until self-refresh is performed, then read MPDDRC_LPR.SELF_DONE.
5. Set MPDDRC_LPR.CHG_FREQ to 1 and MPDDRC_LPR.UPD_MR to 0.
6. Change the frequency. The maximum value is 125 MHz, but some providers allow 200 MHz.

7. In MPDDRC_CR, update appropriate values: Reset DLL (DLL), CAS read latency (CAS), CAS write latency (CAS_WR). DLL must be set to 0. Update other bit fields depending on the Frequency and DLL Off parameter settings.
8. Update all asynchronous timings (TRAS, TRP, TXSRD, etc.) in MPDDRC_TPR0, MPDDRC_TPR1 and MPDDRC_TPR2 depending on the Frequency and DLL Off parameter settings
9. Perform an access to DRAM.
10. DLL is now disabled.

23.5.4.4.2.DLL Off to DLL On Procedure

To switch from DLL Off mode to DLL On mode, follow the procedure below:

1. Make sure no access is in progress and Self-refresh mode is disabled.
2. Enable Self-refresh mode and set MPDDRC_LPR.CHG_FREQ to 1.
3. Wait until self-refresh is performed, then read MPDDRC_LPR.SELF_DONE.
4. Change the frequency. The minimum value is 303 MHz.
5. In MPDDRC_CR, update appropriate values: Reset DLL (DLL), CAS read latency (CAS), CAS write latency (CAS_WR). DIS_DLL must be set to 0 and DLL must be set to 1. Update other bit fields depending on the Frequency and DLL On parameter settings.
6. Update all asynchronous timings (TRAS, TRP, TXSRD, etc.) in MPDDRC_TPR0, MPDDRC_TPR1 and MPDDRC_TPR2 depending on the Frequency and DLL On parameter settings.
7. Perform an access to DRAM.
8. DLL is now enabled.

23.5.4.5.Reset Mode

The Reset mode is a feature of DDR2-SDRAM. This mode is activated by configuring the Low-power Command bit (LPCB) to 3 and writing a '1' to the Clock Frozen Command bit (CLK_FR) in the Low-power register (MPDDRC_LPR).

When this mode is enabled, the MPDDRC leaves Normal mode (MPDDRC_MR.MODE = 0) and the controller is frozen. Before enabling this mode, the user must make sure there is no access in progress.

To exit Reset mode, the Low-power Command bit (LPCB) must be configured to 0, the Clock Frozen Command bit (CLK_FR) must be written to '0' and the initialization sequence must be generated by software (see DDR2-SDRAM Initialization).

23.5.5. Optimized Access Functionality

The DDR-SDRAM protocol imposes a check of timings prior to performing a read or a write access, thus decreasing system performance. An access to DDR-SDRAM is performed if banks and rows are open (or active). To activate a row in a particular bank, the last open row must be deactivated and a new row must be open. Two DDR-SDRAM commands must be performed to open a bank: Precharge command and Activate command with respect to T_{RP} timing. Before performing a read or write command, T_{RCD} timing must be checked.

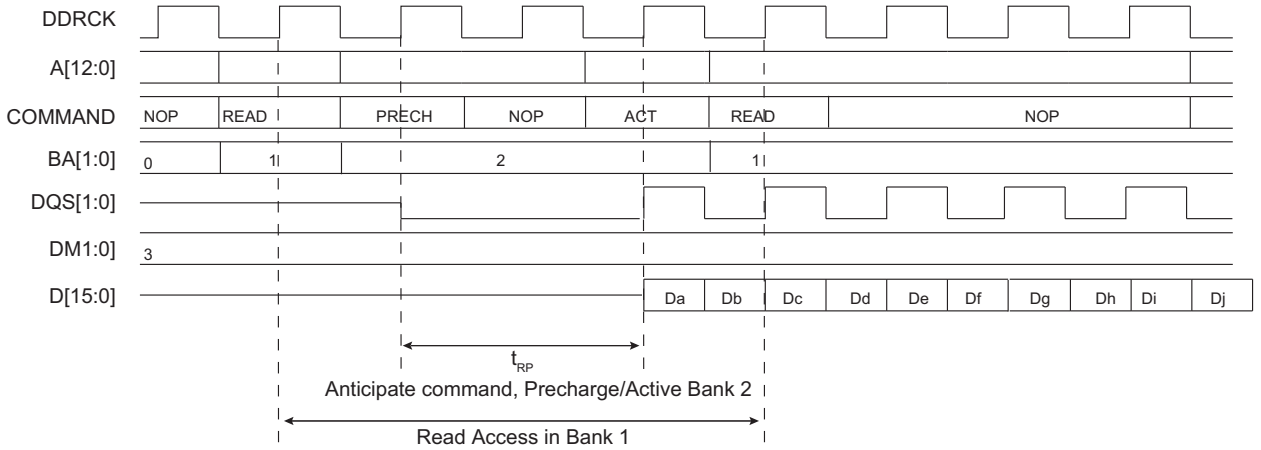
This operation generates a significant bandwidth loss.

The controller is designed to mask these timings and thus improve the system bandwidth.

The MPDDRC is a controller whereby 7 hosts can simultaneously reach the controller. This feature improves the bandwidth of the system because it can detect 7 requests on the system bus client inputs and thus anticipate the commands that follow, Precharge command and Activate command in bank X during the current access in bank Y. This masks t_{RP} and t_{RCD} timings (see the following figure). In the best case, all accesses are done as if the banks and rows were already open. The best condition is met when the 7 hosts work in different banks. In the case of 7 simultaneous read accesses, when the four or eight banks and associated rows are open, the controller reads with a

continuous flow and masks the CAS latency for each access. To allow a continuous flow, the read command must be set at 3, 4, 5 or 6 cycles (CAS latency) before the end of the current access. The arbitration scheme must be changed since the round-robin arbitration cannot be respected. If the controller anticipates a read access, and thus a host with a high priority arises before the end of the current access, then this host will not be serviced.

Figure 23.8. Anticipate Precharge/Activate Command in Bank 2 during Read Access in Bank 1



MPDDRC embeds three arbitration mechanisms based on round-robin arbitration which allows to share the external device between different hosts when two or more hosts try to access the DDR-SDRAM device at the same time.

The three arbitration types are round-robin arbitration and two weighted round-robin arbitrations. For weighted round-robin arbitrations, the priority can be given either depending on the number of requests or words per port, or depending on the required bandwidth per port. The type of arbitration can be chosen by setting the ARB field in the Configuration Arbiter register (MPDDRC_CONF_ARBITER) (see MPDDRC Configuration Arbiter Register).

23.5.5.1. Round-robin Arbitration

Round-robin arbitration is used when the ARB field is set to 0 (see MPDDRC Configuration Arbiter Register). This algorithm dispatches the requests from different hosts to the DDR-SDRAM device in a round-robin manner. If two or more host requests arise at the same time, the host with the lowest number is serviced first, then the others are serviced in a round-robin manner.

To avoid burst breaking and to provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Idle cycles: when no host is connected to the DDR-SDRAM device.
2. Single cycles: when a client is currently doing a single access.
3. End of Burst cycles: when the current cycle is the last cycle of a burst transfer:
   - For bursts of defined length, predicted end of burst matches the size of the transfer.
   - For bursts of undefined length, predicted end of burst is generated at the end of each four-beat boundary inside the INCR transfer.
4. Anticipated Access: when an anticipated read access is done while the current access is not complete, the arbitration scheme can be changed if the anticipated access is not the next access serviced by the arbitration scheme.

23.5.5.2. Request-word Weighted Round-robin Arbitration

In request-word weighted round-robin arbitration, the weight is the number of requests or the number of words per port.

This arbitration scheme is enabled by configuring the ARB field to 1 (see MPDDRC Configuration Arbiter Register). This algorithm grants a port for X[1] consecutive first transfer (htrans = NON SEQUENTIAL) of a burst or X single transfer, or for X word transfers. It is possible to choose between an arbitration scheme by request or by word per port by setting the RQ_WD_Px field (see MPDDRC Configuration Arbiter Register).

Note: 1. X is an integer value provided by some host modules to the arbiter.

It is also possible for the user to provide the number of requests or words (by overwriting the information provided by a host) on host basis by configuring the MA_PR_Px field. Depending on the application, the number of these requests or words can be reduced or increased using the NRD_NWD_BDW_Px fields (see MPDDRC Configuration Arbiter Register).

The TIMEOUT_Px field defines the delay between two accesses on the same port in number of cycles before rearbitrating the access to another port. This field avoids a timeout on the system, as some hosts add idle cycles between two consecutive accesses (see MPDDRC Configuration Arbiter Register).

This algorithm dispatches the requests from different hosts to the DDR-SDRAM device in a round-robin manner. If two or more host requests arise at the same time, the host with the lowest number is serviced first, then the others are serviced in a round-robin manner when the number of requests or words is reached or when the timeout value is reached.

To avoid burst breaking and to provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Timeout is reached: the delay between two accesses is equal to TIMEOUT_Px.
2. The number of requests or words is reached: when the current cycle is the last cycle of a transfer.

23.5.5.3. Bandwidth Weighted Round-robin Arbitration

In bandwidth weighted round-robin arbitration, a minimum bandwidth is guaranteed per port.

This arbitration scheme is enabled when the ARB field is set to 2 (see MPDDRC Configuration Arbiter Register).

This algorithm grants to each port a percentage of the bandwidth. The NRD_NWD_BDW_Px field defines the percentage allocated to each port.

The percentage of the bandwidth is programmed with the NRD_NWD_BDW_Px fields (see MPDDRC Configuration Arbiter Register).

The TIMEOUT_Px field defines the delay between two accesses on the same port in number of cycles rearbitrating the access to another port. This field avoids a timeout on the system, as some hosts add idle cycles between two consecutive accesses (see MPDDRC Configuration Arbiter Register).

This algorithm dispatches the requests from different hosts to the DDR-SDRAM device in a round-robin manner. If two or more host requests arise at the same time, the host with the lowest number is serviced first, then the others are serviced in a round-robin manner when the allocated bandwidth is reached or when the timeout value is reached.

The BDW_BURST field allows to arbitrate either when the current host reaches exactly the programmed bandwidth, or when the current host reaches exactly the programmed bandwidth and the current access is ended (see MPDDRC Configuration Arbiter Register).

To provide the maximum throughput for the DDR-SDRAM device, arbitration must only take place during the following cycles:

1. Timeout is reached: the delay between two accesses is equal to TIMEOUT_Px.
2. The allocated bandwidth is reached although the current cycle is not ended.
3. The allocated bandwidth is reached and the current cycle is the last cycle of a transfer.

23.5.5.4. Quality Of Service Arbitration

This arbitration scheme is enabled when the ARB field is set to 3 (see MPDDRC Configuration Arbiter Register).

The arbitration scheme is organized in priority pools corresponding each to an access criticality class as shown in the corresponding Latency Quality of Service column in the following table. When the Latency Quality of Service is enabled for a host-client pair through the system bus matrix (refer to the section "Bus Matrix (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 system bus matrix user interface. Refer to “MATRIX Priority Register A For Clients” in the section "Bus Matrix (MATRIX)".

Table 23.3. 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, refer to the section "Bus Matrix (MATRIX)"). 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 belong 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 system bus hosts.

Intermediate priority pools enable fine priority tuning. Typically, a latency-sensitive host or a bandwidth-sensitive host uses such a priority level. The higher the priority level (MxPR value, refer to the section "Bus Matrix (MATRIX)"), the higher the host priority.

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.

Some hosts, such as LCD or DMA, drive a signal named HNBREQ on the system bus to indicate the number of transfers to be performed. When the field MPDDRC_CONF_ARBITER.KEEP_LAYER is set to 1, the host with the highest LQOS value and a HNBREQ value different from 0 continues to be granted, even during a last data phase with IDLE cycles.

23.5.6. Scrambling/Unscrambling Function

The external data bus 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 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, KEY1 in the "MPDDRC OCMS KEY1 Register" and KEY2 in the "MPDDRC OCMS KEY2 Register". These key registers are only accessible in Write mode.

The key must be securely stored in a reliable non-volatile 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 can be enabled or disabled by programming the "MPDDRC OCMS Register".

23.5.7. Clearing Scrambling Keys on Tamper Event

On tamper detection event on WKUP pins, it is possible to perform an immediate clear of the scrambling keys (MPDDRC_OCMS_KEY1 and MPDDRC_OCMS_KEY2) if bit MPDDRC_OCMS.TAMPCLR = 1.

23.5.8. Register Write Protection

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

If a write access to a write-protected register is detected, the WPVS flag in the MPDDRC Write Protection Status Register (MPDDRC_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 MPDDRC_WPSR.

The following registers are write-protected when the bit WPEN is set:

- MPDDRC Mode Register
- MPDDRC Refresh Timer Register
- MPDDRC Configuration Register
- MPDDRC Timing Parameter 0 Register
- MPDDRC Timing Parameter 1 Register
- MPDDRC Memory Device Register
- MPDDRC OCMS Register
- MPDDRC OCMS KEY1 Register
- MPDDRC OCMS KEY2 Register

The following registers are write-protected when the bit WPITEN is set:

- MPDDRC Interrupt Enable Register
- MPDDRC Interrupt Disable Register

23.5.9. Bus Monitor, Performance Monitor

The MPDDRC embeds a bus monitor which collects bus transaction information from 7 MPDDRC ports. This information, such as accumulated latency (MPDDRC_MINFOx (TOTAL_LATENCY)) or number of transfers (MPDDRC_MINFOx (NB_TRANSFERS)), can be used to calculate the latency and bandwidth for each port by using a metric counter.

The following metrics are computed for each port within a configurable address range:

- Total number of write transfers
- Total number of read transfers
- Total number of read/write transfers
- Total read latencies

- Total write latencies
- Total write/read latencies
- Maximum latencies per port
- Total number of cycles
- Total latency vs QoS value

The Monitor Configuration (MPDDRC_MCFGR) and Monitor Address High/Low Port x (MPDDRC_MADDRx) registers define the type of transaction collected (read, write or read/write) and the address range to monitor.

MPDDRC address mapping is from 0x2000_0000 to 0x2FFF_FFFF. To define the address range analyzed by the monitor, the address bits [27:13] are used to configure the lower and upper bounds. The lower bound can be modified by writing the field MPDDRC_MADDRx.ADDR_LOW_PORTx. The upper bound can be modified by writing the field MPDDRC_MADDRx.ADDR_HIGH_PORTx. The minimum range that can be analyzed is 8 Kbytes.

By default, the monitor is enabled and the address range is 0x2000_0000 to 0x2FFF_FFFF.

Monitor use example:

1. Clear the configuration register: write 0x00000000 in MPDDRRC_MCFGR.
2. Enable the monitor: write 0x00000001 in MPDDRRC_MCFGR.
3. Reset the monitor: write 0x00000003 in MPDDRRC_MCFGR.
4. Enable the monitor: write 0x00000001 in MPDDRRC_MCFGR.
5. Define the address range in MPDDRC_MADDRx or use the default value.
6. Start monitoring: write 0x00000011 in MPDDRRC_MCFGR.
7. Monitoring is launched. An event can be used to stop monitoring.
8. Stop profiling: write 0x00000001 in MPDDRRC_MCFGR.
9. To know the number of tranfers per port, write 0x00000801 in MPDDRRC_MCFGR and read MPDDRC_MINFOx (NB_TRANSFERS).
10. To know the total number of cycles while monitoring is in progress, write 0x0003001 in MPDDRRC_MCFGR and read MPDDRC_MINFOx (TOTAL_CYCLE_COUNT).

Use the following formulas to compute the latency and bandwidth for each port (in percent):

- Latency:
  (Px_TOTAL_LATENCY/TOTAL_CYCLE_COUNT) *100
- Bandwidth:
  (Px_TOTAL_Px_NB_TRANSFERS/TOTAL_CYCLE_COUNT) *100

23.5.10. Security and Safety Analysis and Reports

Several types of checks are performed when the MPDDRC is accessing the memory device.

The registers listed below are monitored with a single error checker and, in case of a single error event, the error is reported in MPDDRC_WPSR.SEQE:

- Mode register (MPDDRC_MR)
- Memory Device register (MPDDRC_MD
- Refresh Timer register (MPDDRC_RTR)
- Timing Parameter 0/1/2 registers (MPDDRC_TPR0/1/2)
- Low-Power register (MPDDRC_LPR)
- Configuration register (MPDDRC_CR)

- OCMS KEY1 register (MPDDRC_OCMS_KEY1)
- OCMS KEY2 register (MPDDRC_OCMS_KEY2)

The peripheral clock of the MPDDRC is monitored by specific circuitry to detect abnormal waveforms on the internal clock net that may affect the behavior of the MPDDRC. Corruption on the triggering edge of the clock or a pulse with a minimum duration may be identified. If the flag MPDDRC_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 MPDDRC is also monitored and if an abnormal state is detected, the flag MPDDRC_WPSR.SEQE is set. This flag is not set under normal operating conditions.

If the flag MPDDRC_WPSR.CGD = 1, a clock glitch has been detected. This flag is not set under normal operating conditions.

The software accesses to the MPDDRC are monitored and if an incorrect access is performed, the flag MPDDRC_WPSR.SWE is set. The type of incorrect/abnormal software access is reported in the MPDDRC_WPSR.SWETYP field (see MPDDRC Write Protection Status Register (MPDDRC_WPSR) for details), e.g., writing a new configuration (MPDDRC_CR, MPDDRC_TPR0/1/2, MPDDRC_MD, MPDDRC_OCMS, MPDDRC_OCMS_KEY1/2) after the initialization of the MPDDRC (i.e., if MPDDRC_TR.COUNT > 0) is an error. MPDDRC_WPSR.ECLASS is an indicator reporting the criticality of the SWETYP report.

The flags CGD, SEQE, SWE and WPVS are automatically cleared when MPDDRC_WPSR is read.

If one of these flags is set, the flag MPDDRC_ISR.SECE is set and can trigger an interrupt if the MPDDRC_IMR.SECE bit is ‘1’. SECE is cleared by reading MPDDRC_ISR.

The MPDDRC embeds an automatic periodic check of an address of the memory device. This function can be enabled by writing a 1 to the MPDDRC_SAFETY.EN bit. The address to be checked can be configured by writing the field MPDDRC_SAFETY.ADDRESS. When MPDDRC_SAFETY.EN = 1, the MPDDRC performs read and write accesses with specific, predetermined, data patterns to the configured address, with no impact for the application software.

MPDDRC_SAFETY.EN must be set immediately after the initialization sequence.

23.6. Software Interface/SDRAM Organization, Address Mapping

The DDR-SDRAM address space is organized into banks, rows and columns. The MPDDRC maps different memory types depending on values set in the Configuration register (MPDDRC_CR) (see MPDDRC Configuration Register). The tables that follow illustrate the relation between CPU addresses and columns, rows and banks addresses for 16-bit memory data bus widths.

The MPDDRC supports address mapping in Linear mode.

Sequential mode is a method for address mapping where banks alternate at each last DDR-SDRAM page of the current bank.

Interleaved mode is a method for address mapping where banks alternate at each DDR-SDRAM end of page of the current bank.

The MPDDRC makes the DDR-SDRAM device access protocol transparent to the user. The tables that follow illustrate the DDR-SDRAM device memory mapping seen by the user in correlation with the device structure. Various configurations are illustrated.

23.6.1. DDR-SDRAM Address Mapping for 16-bit Memory Data Bus Width

Table 23.4. Sequential Mapping for DDR-SDRAM Configuration, 2K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
						Bk[1:0]		Row[10:0]											Column[7:0]								M0
					Bk[1:0]			Row[10:0]											Column[8:0]								M0
				Bk[1:0]				Row[10:0]											Column[9:0]								M0
			Bk[1:0]					Row[10:0]											Column[10:0]								M0
		Bk[1:0]						Row[10:0]											Column[11:0]								M0

Table 23.5. Interleaved Mapping for DDR-SDRAM Configuration, 2K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
							Row[10:0]										Bk[1:0]		Column[7:0]								M0
						Row[10:0]										Bk[1:0]			Column[8:0]								M0
					Row[10:0]										Bk[1:0]				Column[9:0]								M0
				Row[10:0]										Bk[1:0]					Column[10:0]								M0
			Row[10:0]										Bk[1:0]						Column[11:0]								M0

Table 23.6. Sequential Mapping for DDR-SDRAM Configuration: 4K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
					Bk[1:0]		Row[11:0]												Column[7:0]								M0
				Bk[1:0]			Row[11:0]												Column[8:0]								M0
			Bk[1:0]				Row[11:0]												Column[9:0]								M0
		Bk[1:0]					Row[11:0]												Column[10:0]								M0
	Bk[1:0]						Row[11:0]												Column[11:0]								M0

Table 23.7. Interleaved Mapping for DDR-SDRAM Configuration: 4K Rows, 256/512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
							Row[11:0]										Bk[1:0]		Column[7:0]								M0
						Row[11:0]										Bk[1:0]			Column[8:0]								M0
					Row[11:0]										Bk[1:0]				Column[9:0]								M0
				Row[11:0]										Bk[1:0]					Column[10:0]								M0
			Row[11:0]										Bk[1:0]						Column[11:0]								M0

Table 23.8. Sequential Mapping for DDR-SDRAM Configuration: 8K Rows, 512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
			Bk[1:0]		Row[12:0]														Column[8:0]								M0
		Bk[1:0]			Row[12:0]														Column[9:0]								M0
	Bk[1:0]		Row[12:0]																Column[10:0]								M0
Bk[1:0]		Row[12:0]																	Column[11:0]								M0

Table 23.9. Interleaved Mapping for DDR-SDRAM Configuration: 8K Rows, 512/1024/2048/4096 Columns, 4 Banks

CPU Address Line																										
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
			Row[12:0]													Bk[1:0]		Column[8:0]									M0
		Row[12:0]													Bk[1:0]		Column[9:0]										M0
	Row[12:0]													Bk[1:0]		Column[10:0]											M0
Row[12:0]													Bk[1:0]		Column[11:0]												M0

Table 23.10. Sequential Mapping for DDR-SDRAM Configuration: 16K Rows, 512/1024/2048 Columns, 4 Banks

CPU Address Line																										
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
		Bk[1:0]		Row[13:0]													Column[8:0]										M0
	Bk[1:0]		Row[13:0]														Column[9:0]										M0
Bk[1:0]		Row[13:0]															Column[10:0]										M0

Table 23.11. Interleaved Mapping for DDR-SDRAM Configuration: 16K Rows, 512/1024/2048 Columns, 4 Banks

CPU Address Line																										
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
		Row[13:0]														Bk[1:0]		Column[8:0]									M0
	Row[13:0]														Bk[1:0]		Column[9:0]										M0
Row[13:0]														Bk[1:0]		Column[10:0]											M0

Table 23.12. Sequential Mapping for DDR-SDRAM Configuration: 8K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
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
	Bk[2:0]			Row[12:0]													Column[9:0]										M0

Table 23.13. Interleaved Mapping for DDR-SDRAM Configuration: 8K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
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
	Row[12:0]													Bk[2:0]			Column[9:0]										M0

Table 23.14. Sequential Mapping for DDR-SDRAM Configuration: 16K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
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
Bk[2:0]			Row[13:0]														Column[9:0]										M0

Table 23.15. Interleaved Mapping for DDR-SDRAM Configuration: 16K Rows, 1024 Columns, 8 Banks

CPU Address Line																										
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
Row[13:0]														Bk[2:0]			Column[9:0]										M0

23.7. Register Summary

The User Interface is connected to the APB bus. The MPDDRC is programmed using the registers listed in the following table.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MPDDRC_MR	31:24								
		23:16								
		15:8								
		7:0						MODE[2:0]		
0x04	MPDDRC_RTR	31:24								
		23:16								
		15:8					COUNT[11:8]			
		7:0	COUNT[7:0]							
0x08	MPDDRC_CR	31:24				CAS_WR[2:0]				
		23:16	UNAL	DECOD	NDQS	NB				DQMS
		15:8	SUP_DDR3	OCD[2:0]					DIS_DLL	DIC_DS
		7:0	DLL	CAS[2:0]			NR[1:0]		NC[1:0]	
0x0C	MPDDRC_TPR0	31:24	TMRD[3:0]					TWTR[2:0]		
		23:16	TRRD[3:0]				TRP[3:0]			
		15:8	TRC[3:0]				TWR[3:0]			
		7:0	TRCD[3:0]				TRAS[3:0]			
0x10	MPDDRC_TPR1	31:24					TXP[3:0]			
		23:16	TXSRD[7:0]							
		15:8	TXSNR[7:0]							
		7:0		TRFC[6:0]						
0x14	MPDDRC_TPR2	31:24								
		23:16	TMOD[3:0]				TFAW[3:0]			
		15:8		TRTP[2:0]			TRPA[3:0]			
		7:0	TXARDS[3:0]				TXARD[3:0]			
0x18 ... 0x1B	Reserved									
0x1C	MPDDRC_LPR	31:24						DISTOEN_DONE	SELF_DONE	CHG_FRQ
		23:16			UPD_MR[1:0]				ASR	APDE
		15:8	SRT	SELFAUTO	TIMEOUT[1:0]					
		7:0						CLK_FR	LPCB[1:0]	
0x20	MPDDRC_MD	31:24								
		23:16								
		15:8								
		7:0				DBW		MD[2:0]		
0x24 ... 0x2B	Reserved									
0x2C	MPDDRC_DDR3_CAL	31:24								
		23:16								
		15:8	COUNT_CAL[15:8]							
		7:0	COUNT_CAL[7:0]							
0x30	MPDDRC_DDR3_TIM_CAL	31:24								
		23:16								
		15:8								
		7:0	ZQCS[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	MPDDRC_IO_CALIBR	31:24								
		23:16	CALCODEN[3:0]				CALCODEP[3:0]			
		15:8		TZQIO[8:2]						
		7:0	TZQIO[1:0]		EN_CALIB	CK_F_RANGE[4:0]				
0x38	MPDDRC_OCMS	31:24								
		23:16								
		15:8								
		7:0				TAMPCLR				SCR_EN
0x3C	MPDDRC_OCMS_KEY1	31:24	KEY1[31:24]							
		23:16	KEY1[23:16]							
		15:8	KEY1[15:8]							
		7:0	KEY1[7:0]							
0x40	MPDDRC_OCMS_KEY2	31:24	KEY2[31:24]							
		23:16	KEY2[23:16]							
		15:8	KEY2[15:8]							
		7:0	KEY2[7:0]							
0x44	MPDDRC_CONF_ARBITER	31:24		BDW_BURST_P6	BDW_BURST_P5	BDW_BURST_P4	BDW_BURST_P3	BDW_BURST_P2	BDW_BURST_P1	BDW_BURST_P0
		23:16		MA_PR_P6	MA_PR_P5	MA_PR_P4	MA_PR_P3	MA_PR_P2	MA_PR_P1	MA_PR_P0
		15:8		RQ_WD_P6	RQ_WD_P5	RQ_WD_P4	RQ_WD_P3	RQ_WD_P2	RQ_WD_P1	RQ_WD_P0
		7:0					BDW_MAX_CUR	KEEP_LAYER	ARB[1:0]	
0x48	MPDDRC_TIMEOUT	31:24					TIMEOUT_P6[3:0]			
		23:16	TIMEOUT_P5[3:0]				TIMEOUT_P4[3:0]			
		15:8	TIMEOUT_P3[3:0]				TIMEOUT_P2[3:0]			
		7:0	TIMEOUT_P1[3:0]				TIMEOUT_P0[3:0]			
0x4C	MPDDRC_REQ_PORT_0123	31:24	NRQ_NWD_BDW_P3[7:0]							
		23:16	NRQ_NWD_BDW_P2[7:0]							
		15:8	NRQ_NWD_BDW_P1[7:0]							
		7:0	NRQ_NWD_BDW_P0[7:0]							
0x50	MPDDRC_REQ_PORT_456	31:24								
		23:16	NRQ_NWD_BDW_P6[7:0]							
		15:8	NRQ_NWD_BDW_P5[7:0]							
		7:0	NRQ_NWD_BDW_P4[7:0]							
0x54	MPDDRC_BDW_PORT_0123	31:24		BDW_P3[6:0]						
		23:16		BDW_P2[6:0]						
		15:8		BDW_P1[6:0]						
		7:0		BDW_P0[6:0]						
0x58	MPDDRC_BDW_PORT_456	31:24								
		23:16	BDW_P6[7:0]							
		15:8	BDW_P5[7:0]							
		7:0	BDW_P4[7:0]							
0x5C	MPDDRC_RD_DATA_PATH	31:24								
		23:16								
		15:8								
		7:0							SHIFT_SAMPLING[1:0]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x60	MPDDRC_MCFGR	31:24								
		23:16								
		15:8			INFO[2:0]			REFR_CALIB	READ_WRITE[1:0]	
		7:0				RUN			SOFT_RESET	EN_MONI
0x64	MPDDRC_MADDR0	31:24	ADDR_HIGH_PORT0[15:8]							
		23:16	ADDR_HIGH_PORT0[7:0]							
		15:8	ADDR_LOW_PORT0[15:8]							
		7:0	ADDR_LOW_PORT0[7:0]							
0x68	MPDDRC_MADDR1	31:24	ADDR_HIGH_PORT1[15:8]							
		23:16	ADDR_HIGH_PORT1[7:0]							
		15:8	ADDR_LOW_PORT1[15:8]							
		7:0	ADDR_LOW_PORT1[7:0]							
0x6C	MPDDRC_MADDR2	31:24	ADDR_HIGH_PORT2[15:8]							
		23:16	ADDR_HIGH_PORT2[7:0]							
		15:8	ADDR_LOW_PORT2[15:8]							
		7:0	ADDR_LOW_PORT2[7:0]							
0x70	MPDDRC_MADDR3	31:24	ADDR_HIGH_PORT3[15:8]							
		23:16	ADDR_HIGH_PORT3[7:0]							
		15:8	ADDR_LOW_PORT3[15:8]							
		7:0	ADDR_LOW_PORT3[7:0]							
0x74	MPDDRC_MADDR4	31:24	ADDR_HIGH_PORT4[15:8]							
		23:16	ADDR_HIGH_PORT4[7:0]							
		15:8	ADDR_LOW_PORT4[15:8]							
		7:0	ADDR_LOW_PORT4[7:0]							
0x78	MPDDRC_MADDR5	31:24	ADDR_HIGH_PORT5[15:8]							
		23:16	ADDR_HIGH_PORT5[7:0]							
		15:8	ADDR_LOW_PORT5[15:8]							
		7:0	ADDR_LOW_PORT5[7:0]							
0x7C	MPDDRC_MADDR6	31:24	ADDR_HIGH_PORT6[15:8]							
		23:16	ADDR_HIGH_PORT6[7:0]							
		15:8	ADDR_LOW_PORT6[15:8]							
		7:0	ADDR_LOW_PORT6[7:0]							
0x80 ... 0x83	Reserved									
0x84	MPDDRC_MINFO0 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT0_WAITING[15:8]							
		7:0	MAX_PORT0_WAITING[7:0]							
0x84	MPDDRC_MINFO0 (NB_TRANSFERS)	31:24	P0_NB_TRANSFERS[31:24]							
		23:16	P0_NB_TRANSFERS[23:16]							
		15:8	P0_NB_TRANSFERS[15:8]							
		7:0	P0_NB_TRANSFERS[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY)	31:24	P0_TOTAL_LATENCY[31:24]							
		23:16	P0_TOTAL_LATENCY[23:16]							
		15:8	P0_TOTAL_LATENCY[15:8]							
		7:0	P0_TOTAL_LATENCY[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY_QOS01)	31:24	P0_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P0_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P0_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P0_TOTAL_LATENCY_QOS0[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_LATENCY_QOS23)	31:24	P0_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P0_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P0_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P0_TOTAL_LATENCY_QOS2[7:0]							
0x84	MPDDRC_MINFO0 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0x88	MPDDRC_MINFO1 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT1_WAITING[15:8]							
		7:0	MAX_PORT1_WAITING[7:0]							
0x88	MPDDRC_MINFO1 (NB_TRANSFERS)	31:24	P1_NB_TRANSFERS[31:24]							
		23:16	P1_NB_TRANSFERS[23:16]							
		15:8	P1_NB_TRANSFERS[15:8]							
		7:0	P1_NB_TRANSFERS[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY)	31:24	P1_TOTAL_LATENCY[31:24]							
		23:16	P1_TOTAL_LATENCY[23:16]							
		15:8	P1_TOTAL_LATENCY[15:8]							
		7:0	P1_TOTAL_LATENCY[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY_QOS01)	31:24	P1_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P1_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P1_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P1_TOTAL_LATENCY_QOS0[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_LATENCY_QOS23)	31:24	P1_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P1_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P1_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P1_TOTAL_LATENCY_QOS2[7:0]							
0x88	MPDDRC_MINFO1 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0x8C	MPDDRC_MINFO2 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT2_WAITING[15:8]							
		7:0	MAX_PORT2_WAITING[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x8C	MPDDRC_MINFO2 (NB_TRANSFERS)	31:24	P2_NB_TRANSFERS[31:24]							
		23:16	P2_NB_TRANSFERS[23:16]							
		15:8	P2_NB_TRANSFERS[15:8]							
		7:0	P2_NB_TRANSFERS[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY)	31:24	P2_TOTAL_LATENCY[31:24]							
		23:16	P2_TOTAL_LATENCY[23:16]							
		15:8	P2_TOTAL_LATENCY[15:8]							
		7:0	P2_TOTAL_LATENCY[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY_QOS01)	31:24	P2_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P2_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P2_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P2_TOTAL_LATENCY_QOS0[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_LATENCY_QOS23)	31:24	P2_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P2_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P2_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P2_TOTAL_LATENCY_QOS2[7:0]							
0x8C	MPDDRC_MINFO2 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0x90	MPDDRC_MINFO3 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT3_WAITING[15:8]							
		7:0	MAX_PORT3_WAITING[7:0]							
0x90	MPDDRC_MINFO3 (NB_TRANSFERS)	31:24	P3_NB_TRANSFERS[31:24]							
		23:16	P3_NB_TRANSFERS[23:16]							
		15:8	P3_NB_TRANSFERS[15:8]							
		7:0	P3_NB_TRANSFERS[7:0]							
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY)	31:24	P3_TOTAL_LATENCY[31:24]							
		23:16	P3_TOTAL_LATENCY[23:16]							
		15:8	P3_TOTAL_LATENCY[15:8]							
		7:0	P3_TOTAL_LATENCY[7:0]							
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY_QOS01)	31:24	P3_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P3_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P3_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P3_TOTAL_LATENCY_QOS0[7:0]							
0x90	MPDDRC_MINFO3 (TOTAL_LATENCY_QOS23)	31:24	P3_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P3_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P3_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P3_TOTAL_LATENCY_QOS2[7:0]							
0x90	MPDDRC_MINFO3 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x94	MPDDRC_MINFO4 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT4_WAITING[15:8]							
		7:0	MAX_PORT4_WAITING[7:0]							
0x94	MPDDRC_MINFO4 (NB_TRANSFERS)	31:24	P4_NB_TRANSFERS[31:24]							
		23:16	P4_NB_TRANSFERS[23:16]							
		15:8	P4_NB_TRANSFERS[15:8]							
		7:0	P4_NB_TRANSFERS[7:0]							
0x94	MPDDRC_MINFO4 (TOTAL_LATENCY)	31:24	P4_TOTAL_LATENCY[31:24]							
		23:16	P4_TOTAL_LATENCY[23:16]							
		15:8	P4_TOTAL_LATENCY[15:8]							
		7:0	P4_TOTAL_LATENCY[7:0]							
0x94	MPDDRC_MINFO4 (TOTAL_LATENCY_QOS01)	31:24	P4_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P4_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P4_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P4_TOTAL_LATENCY_QOS0[7:0]							
0x94	MPDDRC_MINFO4 (TOTAL_LATENCY_QOS23)	31:24	P4_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P4_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P4_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P4_TOTAL_LATENCY_QOS2[7:0]							
0x94	MPDDRC_MINFO4 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0x98	MPDDRC_MINFO5 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT5_WAITING[15:8]							
		7:0	MAX_PORT5_WAITING[7:0]							
0x98	MPDDRC_MINFO5 (NB_TRANSFERS)	31:24	P5_NB_TRANSFERS[31:24]							
		23:16	P5_NB_TRANSFERS[23:16]							
		15:8	P5_NB_TRANSFERS[15:8]							
		7:0	P5_NB_TRANSFERS[7:0]							
0x98	MPDDRC_MINFO5 (TOTAL_LATENCY)	31:24	P5_TOTAL_LATENCY[31:24]							
		23:16	P5_TOTAL_LATENCY[23:16]							
		15:8	P5_TOTAL_LATENCY[15:8]							
		7:0	P5_TOTAL_LATENCY[7:0]							
0x98	MPDDRC_MINFO5 (TOTAL_LATENCY_QOS01)	31:24	P5_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P5_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P5_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P5_TOTAL_LATENCY_QOS0[7:0]							
0x98	MPDDRC_MINFO5 (TOTAL_LATENCY_QOS23)	31:24	P5_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P5_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P5_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P5_TOTAL_LATENCY_QOS2[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x98	MPDDRC_MINFO5 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0x9C	MPDDRC_MINFO6 (MAX_WAIT)	31:24						LQOS[1:0]		READ_WRITE
		23:16		SIZE[2:0]				BURST[2:0]		
		15:8	MAX_PORT6_WAITING[15:8]							
		7:0	MAX_PORT6_WAITING[7:0]							
0x9C	MPDDRC_MINFO6 (NB_TRANSFERS)	31:24	P6_NB_TRANSFERS[31:24]							
		23:16	P6_NB_TRANSFERS[23:16]							
		15:8	P6_NB_TRANSFERS[15:8]							
		7:0	P6_NB_TRANSFERS[7:0]							
0x9C	MPDDRC_MINFO6 (TOTAL_LATENCY)	31:24	P6_TOTAL_LATENCY[31:24]							
		23:16	P6_TOTAL_LATENCY[23:16]							
		15:8	P6_TOTAL_LATENCY[15:8]							
		7:0	P6_TOTAL_LATENCY[7:0]							
0x9C	MPDDRC_MINFO6 (TOTAL_LATENCY_QOS01)	31:24	P6_TOTAL_LATENCY_QOS1[15:8]							
		23:16	P6_TOTAL_LATENCY_QOS1[7:0]							
		15:8	P6_TOTAL_LATENCY_QOS0[15:8]							
		7:0	P6_TOTAL_LATENCY_QOS0[7:0]							
0x9C	MPDDRC_MINFO6 (TOTAL_LATENCY_QOS23)	31:24	P6_TOTAL_LATENCY_QOS3[15:8]							
		23:16	P6_TOTAL_LATENCY_QOS3[7:0]							
		15:8	P6_TOTAL_LATENCY_QOS2[15:8]							
		7:0	P6_TOTAL_LATENCY_QOS2[7:0]							
0x9C	MPDDRC_MINFO6 (TOTAL_CYCLE_COUNT)	31:24	TOTAL_CYCLE_COUNT[31:24]							
		23:16	TOTAL_CYCLE_COUNT[23:16]							
		15:8	TOTAL_CYCLE_COUNT[15:8]							
		7:0	TOTAL_CYCLE_COUNT[7:0]							
0xA0 ... 0xBF	Reserved									
0xC0	MPDDRC_IER	31:24								
		23:16								
		15:8								
		7:0							RD_ERR	SEC
0xC4	MPDDRC_IDR	31:24								
		23:16								
		15:8								
		7:0							RD_ERR	SEC
0xC8	MPDDRC_IMR	31:24								
		23:16								
		15:8								
		7:0							RD_ERR	SEC

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xCC	MPDDRC_ISR	31:24								
		23:16								
		15:8								
		7:0							RD_ERR	SEC
0xD0	MPDDRC_SAFETY	31:24				EN	ADDRESS[27:24]			
		23:16	ADDRESS[23:16]							
		15:8	ADDRESS[15:8]							
		7:0	ADDRESS[7:0]							
0xD4 ... 0xE3	Reserved									
0xE4	MPDDRC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				FIRSTE			WPITEN	WPEN
0xE8	MPDDRC_WPSR	31:24	ECLASS						SWETYP[1:0]	
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0					SWE	SEQE	CGD	WPVS

23.7.1. MPDDRC Mode Register

Name: MPDDRC_MR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC 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
						MODE[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – MODE[2:0] MPDDRC Command Mode
This field defines the command issued by the MPDDRC when the SDRAM device is accessed. This register is used to initialize the SDRAM device and to activate Deep Power-down mode.

Value	Name	Description
0	NORMAL_CMD	Normal Mode. Any access to the MPDDRC is decoded normally. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
1	NOP_CMD	The MPDDRC issues a NOP command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
2	PRCGALL_CMD	The MPDDRC issues the All Banks Pre-charge command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the SDRAM.
3	LMR_CMD	The MPDDRC issues a Load Mode Register command when the DDR-SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
4	RFSH_CMD	The MPDDRC issues an Auto-refresh command when the DDR-SDRAM device is accessed regardless of the cycle. Previously, an All Banks Pre-charge command must be issued. To activate this mode, the command must be followed by a write to the DDR-SDRAM.
5	EXT_LMR_CMD	The MPDDRC issues an Extended Load Mode Register command when the SDRAM device is accessed regardless of the cycle. To activate this mode, the command must be followed by a write to the DDR-SDRAM. The write in the DDR-SDRAM must be done in the appropriate bank.
6	CALIB_CMD	Calibration command: to calibrate RTT and RON values for the Process Voltage Temperature (PVT) (DDR3-SDRAM device)

23.7.2. MPDDRC Refresh Timer Register

Name: MPDDRC_RTR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC 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
					COUNT[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
	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

Bits 11:0 – COUNT[11:0] MPDDRC Refresh Timer Count

This 12-bit field is loaded into a timer which generates the refresh pulse. Each time the refresh pulse is generated, a refresh sequence is initiated.

The SDRAM requires auto-refresh cycles at an average periodic interval of T_{refi}. The value to be loaded depends on the MPDDRC clock frequency MCK (main system bus clock) and average periodic interval of T_{refi}.

For example, for an SDRAM with T_{refi} = 7.8 µs and a 133 MHz (7.5 ns) main system bus clock, the value of the COUNT field is configured: $((7.8 \times 10^{-6}) / (7.5 \times 10^{-9}))$ = 1040 or 0x0410.

23.7.3. MPDDRC Configuration Register

Name: MPDDRC_CR
Offset: 0x08
Reset: 0x00207024
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
				CAS_WR[2:0]				
Access				R/W	R/W	R/W		
Reset				0	0	0		

Bit	23	22	21	20	19	18	17	16
	UNAL	DECOD	NDQS	NB				DQMS
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	1	0				0

Bit	15	14	13	12	11	10	9	8
	SUP_DDR3	OCD[2:0]					DIS_DLL	DIC_DS
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	1	1	1			0	0

Bit	7	6	5	4	3	2	1	0
	DLL	CAS[2:0]			NR[1:0]		NC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	1	0	0

Bits 28:26 – CAS_WR[2:0] CAS Write Latency
This field is unique to DDR3-SDRAM.

Value	Name	Description
5	DDR3_CAS5	DDR3 CAS write latency 5, DLL must be enabled, DLL On mode
6	DDR3_CAS6	DDR3 CAS write latency 6, DLL enabled or not, DLL On/Off mode

Bit 23 – UNAL This bit must always be written to 1.

Bit 22 – DECOD Type of Decoding

Value	Name	Description
0	SEQUENTIAL	Method for address mapping where banks alternate at each last DDR-SDRAM page of the current bank.
1	INTERLEAVED	Method for address mapping where banks alternate at each DDR-SDRAM end of page of the current bank.

Bit 21 – NDQS Not DQS.
This bit is found in DDR2-SDRAM devices, in Extended Mode register 1. DQS may be used in Single-ended mode or paired with optional complementary signal NDQS.

Value	Name	Description
0	ENABLED	'Not DQS' is enabled.
1	DISABLED	'Not DQS' is disabled.

Bit 20 – NB Number of Banks

Value	Name	Description
0	4_BANKS	4-bank memory devices
1	8_BANKS	8 banks. Only possible when using DDR2-SDRAM,, DDR3-SDRAM devices.

Bit 16 – DQMS Mask Data is Shared

Value	Name	Description
0	NOT_SHARED	DQM is not shared with another controller
1	SHARED	DQM is shared with another controller

Bit 15 – SUP_DDR3 Supply DDR3-SDRAM or DDR3L-SDRAM
This value is used during the power-up sequence.

Value	Description
0	1.35V DDR3L-SDRAM is used.
7	1.5V DDR3-SDRAM is used.

Bits 14:12 – OCD[2:0] Off-chip Driver
SDRAM Controller supports only two values for OCD (default calibration and exit from calibration). These values MUST always be programmed during the initialization sequence. The default calibration must be programmed first, after which the exit calibration and maintain settings must be programmed.
This field is found only in the DDR2-SDRAM devices.

Value	Name	Description
0	DDR2_EXITCALIB	Exit from OCD Calibration mode and maintain settings
7	DDR2_DEFAULT_CALIB	OCD calibration default

Bit 9 – DIS_DLL Disable DLL
This value is used during the power-up sequence. It is only found in DDR2-SDRAM devices and DDR3-SDRAM devices.

Value	Description
0	Enable DLL.
1	Disable DLL.

Bit 8 – DIC_DS Output Driver Impedance Control (Drive Strength)
This bit name is described as “DS” in some memory data sheets. It defines the output drive strength. This value is used during the power-up sequence.
For DDR3-SDRAM devices, this field is equivalent to ODS, Output Drive Strength.
This bit is found only in DDR2-SDRAM devices and DDR3-SDRAM devices.

Value	Name	Description
0	DDR2_NORMALSTRENGTH_DDR3_RZQ_6	Normal drive strength (DDR2) - RZQ_6 (40 [NOM], DDR3)
1	DDR2_WEAKSTRENGTH_DDR3_RZQ_7	Weak drive strength (DDR2) - RZQ_7 (34 [NOM], DDR3)

Bit 7 – DLL Reset DLL
This bit defines the value of Reset DLL. It is found only in DDR2-SDRAM and DDR3-SDRAM devices. This value is used during the power-up sequence.

Value	Name	Description
0	RESET_DISABLED	Disable DLL reset
1	RESET_ENABLED	Enable DLL reset

Bits 6:4 – CAS[2:0] CAS Latency

Value	Name	Description
3	DDR_CAS3	DDR2 CAS Latency 3

Value	Name	Description
4	DDR_CAS4	DDR2 CAS Latency 4
5	DDR_CAS5	DDR2/DDR3 CAS Latency 5
6	DDR_CAS6	DDR3 CAS Latency 6

Bits 3:2 – NR[1:0] Number of Row Bits

Value	Name	Description
0	11_ROW_BITS	11 bits to define the row number, up to 2048 rows
1	12_ROW_BITS	12 bits to define the row number, up to 4096 rows
2	13_ROW_BITS	13 bits to define the row number, up to 8192 rows
3	14_ROW_BITS	14 bits to define the row number, up to 16384 rows

Bits 1:0 – NC[1:0] Number of Column Bits

Value	Name	Description
0	9_COL_BITS	9 bits to define the column number, up to 512 columns, for DDR2/DDR3-SDRAM
1	10_COL_BITS	10 bits to define the column number, up to 1024 columns, for DDR2/DDR3-SDRAM
2	11_COL_BITS	11 bits to define the column number, up to 2048 columns, for DDR2/DDR3-SDRAM
3	12_COL_BITS	12 bits to define the column number, up to 4096 columns, for DDR2/DDR3-SDRAM

23.7.4. MPDDRC Timing Parameter 0 Register

Name: MPDDRC_TPR0
Offset: 0x0C
Reset: 0x20227225
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	TMRD[3:0]					TWTR[2:0]		
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	1	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	TRRD[3:0]				TRP[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	1	0

Bit	15	14	13	12	11	10	9	8
	TRC[3:0]				TWR[3:0]			
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

Bit	7	6	5	4	3	2	1	0
	TRCD[3:0]				TRAS[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	1	0	1

Bits 31:28 – TMRD[3:0] Load Mode Register Command to Activate or Refresh Command
This field defines the delay between a Load mode register command and an Activate or Refresh command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 26:24 – TWTR[2:0] Internal Write to Read Delay
This field defines the internal Write to Read command time in number of DDRCK clock cycles. The number of cycles is between 1 and 7.

Bits 23:20 – TRRD[3:0] Active BankA to Active BankB
This field defines the delay between an Activate command in BankA and an Activate command in BankB in number of DDRCK clock cycles. The number of cycles is between 1 and 15.

Bits 19:16 – TRP[3:0] Row Precharge Delay
This field defines the delay between a Precharge command and another command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 15:12 – TRC[3:0] Row Cycle Delay
This field defines the delay between an Activate command and a Refresh command in number of DDRCK clock cycles. The number of cycles is between 0 and 15. When TRC is greater than or equal to 60 ns, the maximum supported frequency is 250 MHz. When the frequency is higher than 200 MHz, an offset of one is added automatically. In such case, TRC must be set to 15 to match the value imposed by DDR-SDRAM devices. If the frequency is higher than 200 MHz, the CAS latency must be set higher than 3 (see “CAS: CAS Latency” in MPDDRC_CR).

Bits 11:8 – TWR[3:0] Write Recovery Delay
This field defines the Write Recovery Time in number of DDRCK clock cycles. The number of cycles is between 1 and 15.

Bits 7:4 – TRCD[3:0] Row to Column Delay
This field defines the delay between an Activate command and a Read/Write command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 3:0 – TRAS[3:0] Active to Precharge Delay
This field defines the delay between an Activate command and a Precharge command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

23.7.5. MPDDRC Timing Parameter 1 Register

Name: MPDDRC_TPR1
Offset: 0x10
Reset: 0x03C80808
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
					TXP[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	1	1

Bit	23	22	21	20	19	18	17	16
	TXSRD[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

Bit	15	14	13	12	11	10	9	8
	TXSNR[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	7	6	5	4	3	2	1	0
		TRFC[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	1	0	0	0

Bits 27:24 – TXP[3:0] Exit Power-down Delay to First Command
This field defines the delay between CKE set high and a valid command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.

Bits 23:16 – TXSRD[7:0] Exit Self-refresh Delay to Read Command
This field defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 255.
This field is found only in DDR2-SDRAM and DDR3-SDRAM devices.
In case of DDR3-SDRAM, this field is equivalent to t_{XSDLL}. In DLL Off mode, this timing is not used.
The field must be set to 0.

Bits 15:8 – TXSNR[7:0] Exit Self-refresh Delay to Non-Read Command
This field defines the delay between CKE set high and a Non Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 255. This field is used by the DDR-SDRAM devices. In case of DDR3-SDRAM, this field is equivalent to t_{XS}.

Bits 6:0 – TRFC[6:0] Row Refresh Cycle
This field defines the delay between a Refresh command or a Refresh and Activate command in number of DDRCK clock cycles. The number of cycles is between 0 and 127.

23.7.6. MPDDRC Timing Parameter 2 Register

Name: MPDDRC_TPR2
Offset: 0x14
Reset: 0x00042062
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	TMOD[3:0]				TFAW[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	1	0	0

Bit	15	14	13	12	11	10	9	8
		TRTP[2:0]			TRPA[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	1	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TXARDS[3:0]				TXARD[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	0	1	0

Bits 23:20 – TMOD[3:0] Mode Register Set Command Update Delay
Defines the delay between a Mode Register Set command and another command in terms of number of DDRCK clock cycles. The number of cycles is between 0 and 15. This field is found only on DDR3-SDRAM devices.

Bits 19:16 – TFAW[3:0] Four Active Windows
DDR2 and DDR3 devices with eight banks (1 Gbit or larger) have an additional requirement concerning t_{FAW} timing. This requires that no more than four Activate commands may be issued in any given t_{FAW} (MIN) period.
The number of cycles is between 0 and 15.
This field is found only in DDR2-SDRAM and DDR3-SDRAM devices.

Bits 14:12 – TRTP[2:0] Read to Precharge
Defines the delay between a Read command and a Precharge command in number of DDRCK clock cycles.
The number of cycles is between 0 and 7.

Bits 11:8 – TRPA[3:0] Row Precharge All Delay
Defines the delay between a Precharge All Banks command and another command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

Bits 7:4 – TXARDS[3:0] Exit Active Power Down Delay to Read Command in Mode “Slow Exit”
Defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

Bits 3:0 – TXARD[3:0] Exit Active Power Down Delay to Read Command in Mode “Fast Exit”
Defines the delay between CKE set high and a Read command in number of DDRCK clock cycles. The number of cycles is between 0 and 15.
This field is found only in the DDR2-SDRAM devices.

23.7.7. MPDDRC Low-Power Register

Name: MPDDRC_LPR
Offset: 0x1C
Reset: 0x00010000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						DISTOEN_DONE	SELF_DONE	CHG_FRQ
Access						R	R	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
			UPD_MR[1:0]				ASR	APDE
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	1

Bit	15	14	13	12	11	10	9	8
	SRT	SELFAUTO	TIMEOUT[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
						CLK_FR	LPCB[1:0]	
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 26 – DISTOEN_DONE DLL Disabled to DLL Enabled is Done
Indicates that the DLL Off (disabled) to DLL On (enabled) switching procedure is completed.

Bit 25 – SELF_DONE Self-Refresh is Done
Indicates that external device is in Self-refresh mode.

Bit 24 – CHG_FRQ Change Clock Frequency During Self-Refresh Mode
This mode is used to change the DDR2-SDRAM or DDR3-SDRAM input clock frequency. This mode is unique to the DDR2-SDRAM and DDR3-SDRAM devices.

Bits 21:20 – UPD_MR[1:0] Update Load Mode Register and Extended Mode Register
Used to enable or disable automatic update of the Load Mode register and Extended Mode register. This update depends on the MPDDRC integration in a system. The MPDDRC can either share or not an external bus with another controller.

Value	Name	Description
0	NO_UPDATE	Update of Load Mode and Extended Mode registers is disabled.
1	UPDATE_SHAREDBUS	The MPDDRC shares an external bus. Automatic update is done during a refresh command and a pending read or write access in the SDRAM device.
2	UPDATE_NOSHAREDBUS	The MPDDRC does not share an external bus. Automatic update is done before entering Self-refresh mode.
3	–	Reserved

Bit 17 – ASR Auto Self-Refresh
This mode is unique to DDR3-SDRAM devices supporting an extended temperature range. In this mode, SRT must be disabled.

Value	Description
0	DRAM manages Self-refresh entry in either the normal or extended temperature range. In this mode, DRAM manages Self-refresh power consumption when operating conditions change - lower at low temperatures and higher at high temperatures.
1	Manual Self-refresh reference must be applied.

Bit 16 – APDE Active Power Down Exit Time

This mode is unique to the DDR2-SDRAM and DDR3-SDRAM devices.
This mode manages the active Power-down mode which determines performance versus power saving.
After the initialization sequence, as soon as the APDE field is modified, the Extended Mode register (located in the memory of the external device) is accessed automatically and APDE bits are updated. Depending on the UPD_MR bit, update is done before entering Self-refresh mode or during a refresh command and a pending read or write access

Value	Name	Description
0	DDR2_FAST_EXIT	Fast exit from power-down. DDR2-SDRAM and DDR3-SDRAM devices only.
1	DDR2_SLOW_EXIT	Slow exit from power-down. DDR2-SDRAM and DDR3-SDRAM devices only.

Bit 15 – SRT High Temperature Self-Refresh Rate

This mode is unique to DDR2-SDRAM and DDR3-SDRAM devices supporting the extended temperature range.

Value	Description
1	2x refresh rate. Provides a faster rate on industrial and automotive devices if temperature exceeds 85°C.
0	1x refresh rate. Industrial and automotive devices with temperatures that do not exceed 85°C.

Bit 14 – SELFAUTO Self-Refresh Exit Auto-Refresh

Value	Description
1	Upon exiting Self-refresh mode, auto-refresh command is immediately performed after t_{XSNR}.
0	Upon exiting Self-refresh mode, active command is immediately performed after t_{XSNR}. The auto-refresh command is issued every 15.6 µs or less.

Bits 13:12 – TIMEOUT[1:0] Time Between Last Transfer and Low-Power Mode

Defines when Low-power mode is activated.

Value	Name	Description
0	NONE	SDRAM Low-power mode is activated immediately after the end of the last transfer.
1	DELAY_64_CLK	SDRAM Low-power mode is activated 64 clock cycles after the end of the last transfer.
2	DELAY_128_CLK	SDRAM Low-power mode is activated 128 clock cycles after the end of the last transfer.
3	–	Reserved

Bit 2 – CLK_FR Clock Frozen Command Bit

Sets the clock low during Power-down mode. Some DDR-SDRAM devices do not support freezing the clock during Power-down mode. Refer to the relevant DDR-SDRAM device data sheet for details.

Value	Name	Description
0	DISABLED	Clock(s) is/are not frozen.
1	ENABLED	Clock(s) is/are frozen.

Bits 1:0 – LPCB[1:0] Low-power Command Bit

Value	Name	Description
0	NOLOWPOWER	Low-power feature is inhibited. No Power-down, Self-refresh and Deep power modes are issued to the DDR-SDRAM device.

Value	Name	Description
1	SELFREFRESH	The MPDDRC issues a self-refresh command to the DDR-SDRAM device, the clock(s) is/are deactivated and the CKE signal is set low. The DDR-SDRAM device leaves the Self-refresh mode when accessed and reenters it after the access.
2	POWERDOWN	The MPDDRC issues a Power-down command to the DDR-SDRAM device after each access, the CKE signal is set low. The DDR-SDRAM device leaves the Power-down mode when accessed and reenters it after the access.

23.7.8. MPDDRC Memory Device Register

Name: MPDDRC_MD
Offset: 0x20
Reset: 0x00000013
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC 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
				DBW		MD[2:0]		
Access				R/W		R/W	R/W	R/W
Reset				1		0	1	1

Bit 4 – DBW Data Bus Width

Value	Name	Description
0	–	Reserved
1	DBW_16_BITS	Data bus width is 16 bits.

Bits 2:0 – MD[2:0] Memory Device

Value	Name	Description
4	DDR3_SDRAM	DDR3-SDRAM
6	DDR2_SDRAM	DDR2-SDRAM

23.7.9. MPDDRC DDR3 Calibration Register

Name: MPDDRC_DDR3_CAL
Offset: 0x2C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC 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
	COUNT_CAL[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
	COUNT_CAL[7:0]							
Access	R/W	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 – COUNT_CAL[15:0] DDR3 Calibration Timer Count

This 16-bit field is loaded into a timer which generates the calibration pulse. Each time the calibration pulse is generated, a ZQCS calibration sequence is initiated. The ZQCS Calibration command is used to calibrate DRAM Ron values over PVT. One ZQCS command can effectively correct at least 1.5% of output impedance errors within T_{zqcs}.

One method for calculating the interval between ZQCS commands gives the temperature ($T_{driftrate}$) and voltage ($V_{driftrate}$) drift rates to which the SDRAM is subject in the application. The interval could be defined by the following formula:

- $ZQCorrection/((T_{Sens} \times T_{driftrate}) + (V_{Sens} \times V_{driftrate}))$

where T_{Sens} = max(dRONdTM) and V_{Sens} = max(dRONdVM) define the SDRAM temperature and voltage sensitivities.

For example, if T_{Sens} = 0.75%/C, V_{Sens} = 0.2%/mV, $T_{driftrate}$ = 1C/sec and $V_{driftrate}$ = 15 mV/s, then the interval between ZQCS commands is calculated as:

- $1.5/((0.75 \times 1) + (0.2 \times 15)) = 0.4s$

In this example, the devices require a calibration every 0.4s. The value to be loaded depends on the average time between the REFRESH commands, t_{REF}. For example, for a device with the time between refresh of 7.8 µs, the value of the COUNT_CAL field is programmed as follows: $(0.4/7.8 \times 10^{-6}) = 0xC852$.

T_{Sens} and V_{Sens} are provided by the manufacturer (Output Driver Sensitivity definition). $T_{driftrate}$ and $V_{driftrate}$ are defined by the end user.

23.7.10. MPDDRC DDR3 Timing Calibration Register

Name: MPDDRC_DDR3_TIM_CAL
Offset: 0x30
Reset: 0x00000006
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
	ZQCS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bits 7:0 – ZQCS[7:0] ZQ Calibration Short
Defines the delay between the ZQ Calibration command and any valid command in number of DDRCK clock cycles.
The number of cycles is between 0 and 255. This field applies to DDR3 devices.

23.7.11. MPDDRC I/O Calibration Register

Name: MPDDRC_IO_CALIBR
Offset: 0x34
Reset: 0x00870000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	CALCODEN[3:0]				CALCODEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1

Bit	15	14	13	12	11	10	9	8
		TZQIO[8:2]						
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
	TZQIO[1:0]		EN_CALIB	CK_F_RANGE[4:0]				
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:20 – CALCODEN[3:0] Number of N-type Transistors
Gives the number of N-type transistors to perform the calibration.

Bits 19:16 – CALCODEP[3:0] Number of P-type Transistors
Gives the number of P-type transistors to perform the calibration.

Bits 14:6 – TZQIO[8:0] IO Calibration
Defines the delay between the start up of the amplifier and the beginning of the calibration, in number of DDRCK clock cycles. The value of this field must be set to 1.5 µs. The number of cycles is between 0 and 512.
The TZQIO configuration code must be set correctly depending on the clock frequency using the following formula:

- $TZQIO = (DDRCK \times (1.5 \times 10^{-6})) + 1$

where the DDRCK frequency is in Hz.
For example, for a frequency of 266 MHz, the value of the TZQIO field is configured $(266 \times 10^{6}) \times (1.5 \times 10^{-6}) + 1 = 400$.

Bit 5 – EN_CALIB Enable Calibration
Enables calibration for the DDR2 devices. When the calibration is enabled, it is recommended to define the COUNT_CAL field (see "COUNT_CAL: LPDDR2 LPDDR3 and DDR3 Calibration Timer Count").
This 16-bit field is loaded into a timer which generates the calibration pulse. Each time the calibration pulse is generated, a calibration sequence is initiated.

Value	Name	Description
0	DISABLE_CALIBRATION	Calibration is disabled.

Value	Name	Description
1	ENABLE_CALIBRATION	Calibration is enabled.

Bits 4:0 – CK_F_RANGE[4:0] DDRCK Maximum Clock Frequency Range
This field is written only once at the initialization sequence and is always written to 1F whatever the frequency configured on DDRCK.

23.7.12. MPDDRC OCMS Register

Name: MPDDRC_OCMS
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the MPDDRC 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
				TAMPCLR				SCR_EN
Access				R/W				R/W
Reset				0				0

Bit 4 – TAMPCLR Tamper Clear Enable

Value	Description
0	A tamper detection event has no effect on MPDDRC scrambling keys.
1	A tamper detection event immediately clears MPDDRC scrambling keys.

Bit 0 – SCR_EN Scrambling Enable

Value	Description
0	Disables “Off-chip” scrambling for SDRAM access.
1	Enables “Off-chip” scrambling for SDRAM access.

23.7.13. MPDDRC OCMS KEY1 Register

Name: MPDDRC_OCMS_KEY1
Offset: 0x3C
Reset: –
Property: Write-only

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

This register can only be written once.

Bit	31	30	29	28	27	26	25	24
	KEY1[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY1[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY1[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY1[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY1[31:0] Off-chip Memory Scrambling (OCMS) Key Part 1
When Off-chip Memory Scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

23.7.14. MPDDRC OCMS KEY2 Register

Name: MPDDRC_OCMS_KEY2
Offset: 0x40
Reset: –
Property: Write-only

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

This register can only be written once.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – KEY2[31:0] Off-chip Memory Scrambling (OCMS) Key Part 2
When Off-chip Memory Scrambling is enabled, the data scrambling depends on KEY1 and KEY2 values.

23.7.15. MPDDRC Configuration Arbiter Register

Name: MPDDRC_CONF_ARBITER
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		BDW_BURST_P6	BDW_BURST_P5	BDW_BURST_P4	BDW_BURST_P3	BDW_BURST_P2	BDW_BURST_P1	BDW_BURST_P0
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
		MA_PR_P6	MA_PR_P5	MA_PR_P4	MA_PR_P3	MA_PR_P2	MA_PR_P1	MA_PR_P0
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
		RQ_WD_P6	RQ_WD_P5	RQ_WD_P4	RQ_WD_P3	RQ_WD_P2	RQ_WD_P1	RQ_WD_P0
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
					BDW_MAX_CUR	KEEP_LAYER	ARB[1:0]	
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 24, 25, 26, 27, 28, 29, 30 – BDW_BURST_Px Bandwidth Arbitration Mode on Port X

Value	Description
0	The arbitration is done when the bandwidth limit defined in MPDDRC_BDW_PORT_0123/456.BDW_Px is reached. If the bandwidth limit is reached during a burst access, the burst is completed.
1	The arbitration is done when the bandwidth limit defined in MPDDRC_BDW_PORT_0123/456.BDW_Px is reached. If the bandwidth limit is reached during a burst access, the burst is broken.

Bits 16, 17, 18, 19, 20, 21, 22 – MA_PR_Px Host or Software Provide Information

Value	Description
0	Number of requests or words is provided by the host, if the host supports this feature.
1	Number of requests or words is provided by software, see "NRQ_NWD_BDW_Px: Number of Requests, Number of Words or Bandwidth Allocation from Port 0-1-2-3".

Bits 8, 9, 10, 11, 12, 13, 14 – RQ_WD_Px Request or Word from Port X

Value	Description
0	Number of requests is selected.
1	Number of words is selected.

Bit 3 – BDW_MAX_CUR Bandwidth Max or Current

Displays the maximum bandwidth or the current bandwidth for each port.
The maximum bandwidth is computed when at least two ports of MPDDRC are used.
This information is provided in MPDDRC Current/Maximum Bandwidth Port 0-1-2-3 Register and MPDDRC Current/Maximum Bandwidth Port 4-5-6-7 Register.

Value	Description
0	Current bandwidth is displayed.
1	Maximum bandwidth is displayed.

Bit 2 – KEEP_LAYER Layer (Port) Kept by Host

Selects the type of arbitration: round-robin, number of requests per port or bandwidth per port.

Value	Description
0	No impact on arbitration scheme
1	Some hosts, such as LCD or DMA, drive a signal named HNBREQ on the system bus to indicate the number of transfers to be performed. The host with the highest LQOS value and a HNBREQ value different from 0 continues to be granted, even during a last data phase with IDLE cycles. This function is effective if the type of arbitration is LQOS.

Bits 1:0 – ARB[1:0] Type of Arbitration

Selects the type of arbitration: round-robin, number of requests per port or bandwidth per port.

Value	Name	Description
0	ROUND	Round-robin policy
1	NB_REQUEST	Request policy
2	BANDWIDTH	Bandwidth policy
3	LQOS	Quality of Service policy

23.7.16. MPDDRC Timeout Register

Name: MPDDRC_TIMEOUT
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
					TIMEOUT_P6[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
	TIMEOUT_P5[3:0]				TIMEOUT_P4[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
	TIMEOUT_P3[3:0]				TIMEOUT_P2[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
	TIMEOUT_P1[3:0]				TIMEOUT_P0[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 – TIMEOUT_Px Timeout for Ports 0, 1, 2, 3, 4, 5, 6
Some hosts insert an idle state between two accesses. This field defines the delay between two accesses on the same port in number of DDRCK clock cycles before arbitration and handing the access over to another port.
This field is not used with round-robin and bandwidth arbitrations.
The number of cycles is between 1 and 15.

23.7.17. MPDDRC Request Port 0-1-2-3 Register

Name: MPDDRC_REQ_PORT_0123
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	NRQ_NWD_BDW_P3[7:0]							
Access	R/W	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
	NRQ_NWD_BDW_P2[7:0]							
Access	R/W	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
	NRQ_NWD_BDW_P1[7:0]							
Access	R/W	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
	NRQ_NWD_BDW_P0[7:0]							
Access	R/W	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 – NRQ_NWD_BDW_Px Number of Requests, Number of Words or Bandwidth Allocation from Port 0-1-2-3

The number of requests corresponds to the number of start transfers. For example, setting this field to 2 performs two burst accesses regardless of the burst type (INCR4, INCR8, etc.). The number of words corresponds exactly to the number of accesses; setting this field to 2 performs two accesses. In this example, burst accesses will be broken.
These values depend on scheme arbitration (see MPDDRC Configuration Arbiter Register).
In case of round-robin arbitration, this field is not used. In case of “bandwidth arbitration”, this field corresponds to percentage allocated for each port. In case of “request” arbitration, this field corresponds to number of start transfers or to number of accesses allocated for each port.

23.7.18. MPDDRC Request Port 4-5-6 Register

Name: MPDDRC_REQ_PORT_456
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
	NRQ_NWD_BDW_P6[7:0]							
Access	R/W	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
	NRQ_NWD_BDW_P5[7:0]							
Access	R/W	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
	NRQ_NWD_BDW_P4[7:0]							
Access	R/W	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 – NRQ_NWD_BDW_Px Number of Requests, Number of Words or Bandwidth allocation from port 4-5-6

The number of requests corresponds to the number of start transfers. For example, setting this field to 2 performs two burst accesses regardless of the burst type (INCR4, INCR8,etc.). The number of words corresponds exactly to the number of accesses; setting this field to 2 performs two accesses. In this example, burst accesses will be broken.
These values depend on scheme arbitration (see MPDDRC Configuration Arbiter Register).
In case of round-robin arbitration, this field is not used. In case of “bandwidth arbitration”, this field corresponds to percentage allocated for each port. In case of “request” arbitration, this field corresponds to number of start transfers or to number of accesses allocated for each port.

23.7.19. MPDDRC Current/Maximum Bandwidth Port 0-1-2-3 Register

Name: MPDDRC_BDW_PORT_0123
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		BDW_P3[6:0]						
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
		BDW_P2[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
		BDW_P1[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
		BDW_P0[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 0:6, 8:14, 16:22, 24:30 – BDW_Px Current/Maximum Bandwidth from Port 0-1-2-3
Displays the current bandwidth or the maximum bandwidth for each port. This information is provided in the "BDW_MAX_CUR: Bandwidth Max or Current" field description.

23.7.20. MPDDRC Current/Maximum Bandwidth Port 4-5-6 Register

Name: MPDDRC_BDW_PORT_456
Offset: 0x58
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
	BDW_P6[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
	BDW_P5[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
	BDW_P4[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 – BDW_Px Current/Maximum Bandwidth from Port 4-5-6
Displays the current bandwidth or the maximum bandwidth for each port. This information is given in the "BDW_MAX_CUR: Bandwidth Max or Current" field description.

23.7.21. MPDDRC Read Data Path Register

Name: MPDDRC_RD_DATA_PATH
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
							SHIFT_SAMPLING[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – SHIFT_SAMPLING[1:0] Shift Sampling Point of Data
Shifts the sampling point of data coming from the memory device. The higher the memory device clock frequency, the higher the SHIFT_SAMPLING value. Refer to the section "Electrical Characteristics".

Value	Name	Description
0	NO_SHIFT	Initial sampling point.
1	SHIFT_ONE_CYCLE	Sampling point is shifted by one cycle.
2	SHIFT_TWO_CYCLES	Sampling point is shifted by two cycles.
3	SHIFT_THREE_CYCLES	Sampling point is shifted by three cycles, unique for DDR2 and DDR3.

23.7.22. MPDDRC Monitor Configuration Register

Name: MPDDRC_MCFGR
Offset: 0x60
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
			INFO[2:0]			REFR_CALIB	READ_WRITE[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
				RUN			SOFT_RESET	EN_MONI
Access				R/W			R/W	R/W
Reset				0			0	0

Bits 13:11 – INFO[2:0] Information Type
Reports information such as latency and the number of transfers monitored on port x [x = 0..6].

Value	Name	Description
0	MAX_WAIT	Information concerning the transfer with the longest waiting time
1	NB_TRANSFERS	Number of transfers on the port
2	TOTAL_LATENCY	Total latency on the port
3	–	Reserved
4	MAX_WAIT_QOS01	Information concerning the transfer with the longest waiting time, depending on QOS values (0 and 1)
5	MAX_WAIT_QOS23	Information concerning the transfer with the longest waiting time, depending on QOS values (2 and 3)
6	TOTAL_CYCLE_COUNT	Indicates the total number of cycles from beginning to end of monitoring.

Bit 10 – REFR_CALIB Refresh Calibration

Value	Description
0	Monitoring does not depend on Auto-refresh mode, Self-refresh mode, Power-down mode, DLL nor calibration impact.
1	Monitoring depends on Auto-refresh mode, Self-refresh mode, Power-down mode, DLL and calibration impact.

Bits 9:8 – READ_WRITE[1:0] Read/Write Access
Used to monitor different types of access.

Value	Name	Description
0	TRIG_RD_WR	Read and Write accesses are triggered.
1	TRIG_WR	Only Write accesses are triggered.
2	TRIG_RD	Only Read accesses are triggered.

Value	Name	Description
3	–	Reserved

Bit 4 – RUN Control Monitor

Value	Description
0	Monitoring is halted. All counters are stopped.
1	Monitoring is launched.

Bit 1 – SOFT_RESET Soft Reset

Value	Description
0	Soft reset is not performed.
1	Soft reset is performed.

Bit 0 – EN_MONI Enable Monitor

Value	Description
0	Monitor is disabled.
1	Monitor is enabled.

23.7.23. MPDDRC Monitor Address High/Low Port x Register

Name: MPDDRC_MADDRx
Offset: 0x64 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ADDR_HIGH_PORTx[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_HIGH_PORTx[7:0]							
Access	R/W	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_LOW_PORTx[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_LOW_PORTx[7:0]							
Access	R/W	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 – ADDR_HIGH_PORTx[15:0] Address High on Port x
Address high which defines the interval to be monitored on port x. This address must be programmed according to the memory mapping of the product.

Bits 15:0 – ADDR_LOW_PORTx[15:0] Address Low on Port x
Address low which defines the interval to be monitored on port x. This address must be programmed according to the memory mapping of the product.

23.7.24. MPDDRC Monitor Information Port x Register (MAX_WAIT)

Name: MPDDRC_MINFOx (MAX_WAIT)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
						LQOS[1:0]		READ_WRITE
Access						R	R	R
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
		SIZE[2:0]				BURST[2:0]		
Access		R	R	R		R	R	R
Reset		0	0	0		0	0	0

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

Bits 26:25 – LQOS[1:0] Value of Quality Of Service on Port x
Can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.
This field reports the Quality of Service value for the maximum waiting time.

Value	Name	Description
0	BACKGROUND	Background transfers
1	BANDWIDTH	Bandwidth sensitive
2	SENSITIVE_LAT	Latency sensitive
3	CRITICAL_LAT	Latency critical

Bit 24 – READ_WRITE Read or Write Access on Port x
Can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.
Reports the transfer direction for the maximum waiting time.

Value	Description
0	Read transfer
1	Write transfer

Bits 22:20 – SIZE[2:0] Transfer Size on Port x
Can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.
Reports the size of the transfer for the maximum waiting time.

Value	Name	Description
0	8BITS	Byte transfer
1	16BITS	Halfword transfer
2	32BITS	Word transfer
3	64BITS	Dword transfer
4–7	–	Reserved

Bits 18:16 – BURST[2:0] Type of Burst on Port x

Can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.
Reports the type of burst for the maximum waiting time.

Value	Name	Description
0	SINGLE	Single transfer
1	INCR	Incrementing burst of unspecified length
2	WRAP4	4-beat wrapping burst
3	INCR4	4-beat incrementing burst
4	WRAP8	8-beat wrapping burst
5	INCR8	8-beat incrementing burst
6	WRAP16	16-beat wrapping burst
7	INCR16	16-beat incrementing burst

Bits 15:0 – MAX_PORTx_WAITING[15:0] Address High on Port x

Can be read if the INFO field in the MPDDRC Monitor Configuration register is set to 0.
Reports the maximum waiting time and the associated type of transfer (burst, size, read or write).

23.7.25. MPDDRC Monitor Information Port x Register (NB_TRANSFERS)

Name: MPDDRC_MINFOx (NB_TRANSFERS)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	Px_NB_TRANSFERS[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
	Px_NB_TRANSFERS[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
	Px_NB_TRANSFERS[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
	Px_NB_TRANSFERS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Px_NB_TRANSFERS[31:0] Number of Transfers on Port x
Can be read if the INFO field is set to 1.
Reports the number of transfers performed within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used.

23.7.26. MPDDRC Monitor Information Port x Register (TOTAL_LATENCY)

Name: MPDDRC_MINFOx (TOTAL_LATENCY)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY[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
	Px_TOTAL_LATENCY[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
	Px_TOTAL_LATENCY[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
	Px_TOTAL_LATENCY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Px_TOTAL_LATENCY[31:0] Total Latency on Port x
Can be read if the INFO field is set to 2.
Reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used.

23.7.27. MPDDRC Monitor Information Port x Register (TOTAL_LATENCY_QOS01)

Name: MPDDRC_MINFOx (TOTAL_LATENCY_QOS01)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY_QOS1[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
	Px_TOTAL_LATENCY_QOS1[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
	Px_TOTAL_LATENCY_QOS0[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
	Px_TOTAL_LATENCY_QOS0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – Px_TOTAL_LATENCY_QOS1[15:0] Total Latency on Port x when value of qos is 1
Can be read if the INFO field is set to 4.
Reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 1.

Bits 15:0 – Px_TOTAL_LATENCY_QOS0[15:0] Total Latency on Port x when value of qos is 0
Can be read if the INFO field is set to 4.
Reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 0.

23.7.28. MPDDRC Monitor Information Port x Register (TOTAL_LATENCY_QOS23)

Name: MPDDRC_MINFOx (TOTAL_LATENCY_QOS23)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	Px_TOTAL_LATENCY_QOS3[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
	Px_TOTAL_LATENCY_QOS3[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
	Px_TOTAL_LATENCY_QOS2[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
	Px_TOTAL_LATENCY_QOS2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – Px_TOTAL_LATENCY_QOS3[15:0] Total Latency on Port x when value of qos is 3
Can be read if the INFO field is set to 5.
Reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 3.

Bits 15:0 – Px_TOTAL_LATENCY_QOS2[15:0] Total Latency on Port x when value of qos is 2
Can be read if the INFO field is set to 5.
Reports the total latency within an interval (ADDR_HIGH_PORT and ADDR_LOW_PORT) when the port is used with qos = 2.

23.7.29. MPDDRC Monitor Information Port x Register (TOTAL_CYCLE_COUNT)

Name: MPDDRC_MINFOx (TOTAL_CYCLE_COUNT)
Offset: 0x84 + x*0x04 [x=0..6]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TOTAL_CYCLE_COUNT[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
	TOTAL_CYCLE_COUNT[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
	TOTAL_CYCLE_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
	TOTAL_CYCLE_COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TOTAL_CYCLE_COUNT[31:0] Total Cycle Count
Can be read if the INFO field is set to 6.
This field reports the total number of cycles from monitor launch to monitor halt.

23.7.30. MPDDRC Interrupt Enable Register

Name: MPDDRC_IER
Offset: 0xC0
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MPDDRC 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
							RD_ERR	SEC
Access							W	W
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Enable

Bit 0 – SEC Security and /or Safety Interrupt Enable

23.7.31. MPDDRC Interrupt Disable Register

Name: MPDDRC_IDR
Offset: 0xC4
Reset: –
Property: Write-only

This register can only be written if the WPITEN bit is cleared in the MPDDRC 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
							RD_ERR	SEC
Access							W	W
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Disable

Bit 0 – SEC Security and /or Safety Interrupt Disable

23.7.32. MPDDRC Interrupt Mask Register

Name: MPDDRC_IMR
Offset: 0xC8
Reset: –
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
							RD_ERR	SEC
Access							R	R
Reset							–	–

Bit 1 – RD_ERR Read Error Interrupt Mask

Bit 0 – SEC Security and /or Safety Interrupt Mask

23.7.33. MPDDRC Interrupt Status Register

Name: MPDDRC_ISR
Offset: 0xCC
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
							RD_ERR	SEC
Access							R	R
Reset							–	–

Bit 1 – RD_ERR Read Error

Value	Description
0	There is no error during memory check.
1	There is one error during memory check.

Bit 0 – SEC Security and /or Safety Event

Value	Description
0	There is no security report in MPDDRC_WPSR.
1	One security flag is set in MPDDRC_WPSR.

23.7.34. MPDDRC Safety Register

Name: MPDDRC_SAFETY
Offset: 0xD0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
				EN	ADDRESS[27:24]			
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
	ADDRESS[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
	ADDRESS[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
	ADDRESS[7:0]							
Access	R/W	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 – EN Enable Periodic Check of Memory Device

Value	Description
0	Memory check is disabled.
1	Memory check is enabled.

Bits 27:0 – ADDRESS[27:0] Memory Device Address
Memory device address to be checked.

23.7.35. MPDDRC Write Protection Mode Register

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

See Register Write Protection for the list of registers that can be 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
				FIRSTE			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
0x444452	PASSWD	Writing any other value in this field aborts the write operation of the WPEN and WPITEN bits. Always reads as 0.

Bit 4 – FIRSTE First Error Report Enable

Value	Description
0	The last write protection violation source is reported in MPDDRC_WPSR.WPVSRC and the last software control error type is reported in MPDDRC_WPSR.SWETYP. The MPDDRC_ISR.SEC flag is set at the first error occurrence within a series.
1	Only the first write protection violation source is reported in MPDDRC_WPSR.WPVSRC and only the first software control error type is reported in MPDDRC_WPSR.SWETYP. The MPDDRC_ISR.SEC flag is set at the first error occurrence within a series.

Bit 1 – WPITEN Write Protection Interruption Enable

Value	Description
0	Disables the write protection on interrupt registers if WPKEY corresponds to 0x444452 (“DDR” in ASCII).
1	Enables the write protection on interrupt registers if WPKEY corresponds to 0x444452 (“DDR” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x444452 (“DDR” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x444452 (“DDR” in ASCII).

23.7.36. MPDDRC Write Protection Status Register

Name: MPDDRC_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 is performed.
1 (ERROR): An access is performed into some registers after memory device initialization sequence.

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	MPDDRC 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	W_AFTER_INIT	Abnormal use of MPDDRC user interface when memory device is already configured and initialized, i.e., if MPDDRC_RTR.COUNT > 0 (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 MPDDRC_WPSR.
1	A software error has occurred since the last read of MPDDRC_WPSR. The field SWE 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 or Non-Correctable Error on Configuration Registers (cleared on read)

Value	Description
0	In the registers listed below, no peripheral internal sequencer error has occurred or no single error has occurred since the last read of MPDDRC_WPSR: • Mode register (MPDDRC_MR) • Memory Device register (MPDDRC_MD • Refresh Timer register (MPDDRC_RTR) • Timing Parameter 0/1/2 registers (MPDDRC_TPR0/1/2) • Low-Power register (MPDDRC_LPR) • Configuration register (MPDDRC_CR) • OCMS KEY1 register (MPDDRC_OCMS_KEY1) • OCMS KEY2 register (MPDDRC_OCMS_KEY2)
1	A peripheral internal sequencer error has occurred since the last read of MPDDRC_WPSR. In the registers listed below, this flag can only be set under abnormal operating conditions or if a non-correctable error has occurred since the last read of MPDDRC_WPSR: • Mode register (MPDDRC_MR) • Memory Device register (MPDDRC_MD • Refresh Timer register (MPDDRC_RTR) • Timing Parameter 0/1/2 registers (MPDDRC_TPR0/1/2) • Low-Power register (MPDDRC_LPR) • Configuration register (MPDDRC_CR) • OCMS KEY1 register (MPDDRC_OCMS_KEY1) • OCMS KEY2 register (MPDDRC_OCMS_KEY2) **Note:** The error is not corrected and the software must reconfigure all registers.

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 MPDDRC_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of MPDDRC_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Enable

Value	Description
0	No write protection violation occurred since the last read of this register (MPDDRC_WPSR).
1	A write protection violation occurred since the last read of this register (MPDDRC_WPSR). If this violation is an unauthorized attempt to write a control register, the associated violation is reported into the WPVSRC field.

24. OTP Memory Controller (OTPC)

24.1. Description

The One-Time Programmable (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:

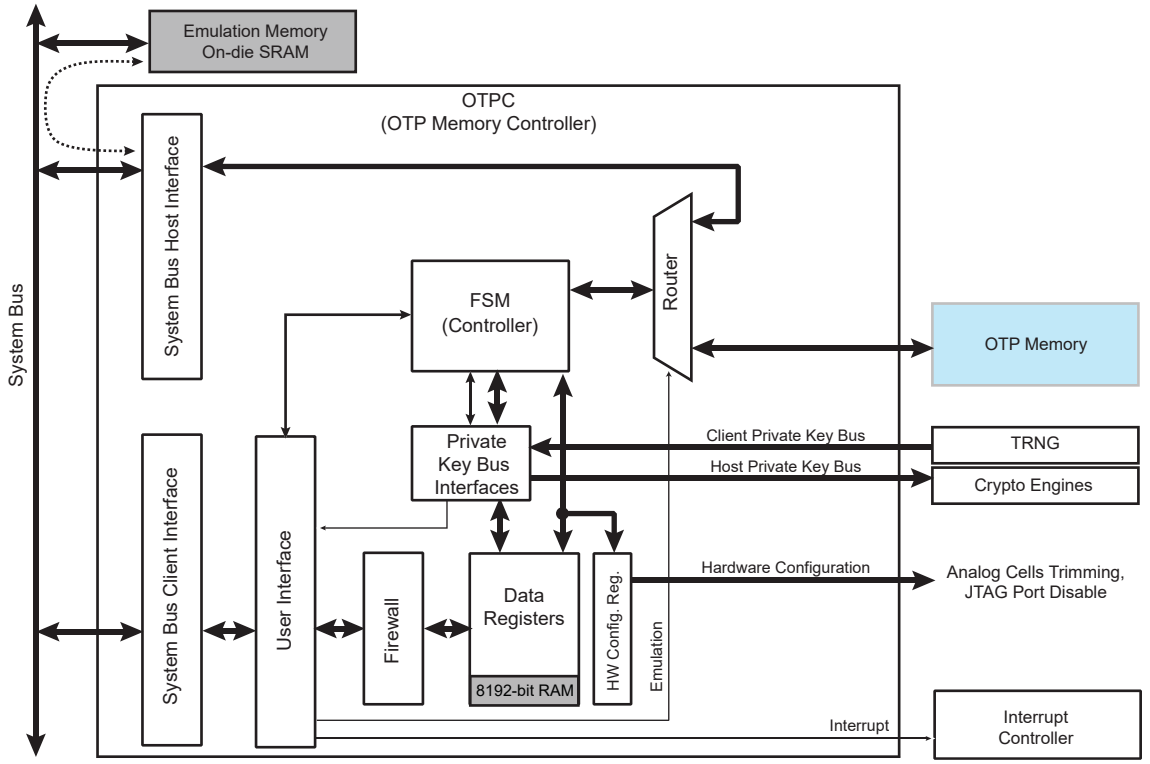
- calibration bits for analog cells (for example, RC oscillators, etc.),
- hardware configuration settings (for example, JTAG disable, etc.),
- chip identifiers,
- key data invisible by software,
- user data.

24.2. Embedded Characteristics

- Programs and Reads the Memory by Software
- Emulation Mode
- Automatic Check of Programmed Bits on Start-up for Safe Operation
- OTP Live Repair
- 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

24.3. Block Diagram

Figure 24.1. OTPC Block Diagram



24.4. Product Dependencies

24.4.1. Power Management

The OTPC is clocked through the Power Management Controller (PMC). The user must power on the main RC oscillator and enable the peripheral clock of the OTPC prior to reading or writing the OTP memory.

24.4.2. 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.

24.5. Functional Description

24.5.1. 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). 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.

24.5.2. OTP Memory Partitioning

The OTP memory is partitioned into different areas:

- Reserved area
- 10-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.

24.5.3. User Area

24.5.3.1. 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.

24.5.3.2. Area Mapping

The entire User area space is mapped into 32-bit words. Each 32-bit word is part of one packet.

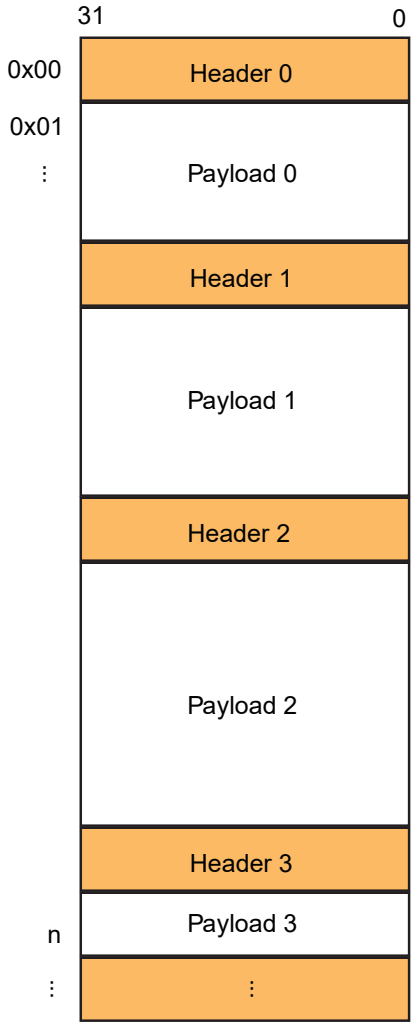
Each packet contains a 32-bit header and 32-bit words of payload (data). The number of payload words is defined in the header.

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 24.2. User Area Memory Mapping



The number of packets depends on the size of the User area and the size of each packet.

24.5.3.3.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.

24.5.3.3.1.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	–	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.
- 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.

24.5.3.3.2. “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 24.1. “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 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.

24.5.3.4.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, 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.

At the end of the init sequence, the value of the last HARDWARE_CONFIGURATION “special“ packet found is applied to the hardware.

24.5.3.5.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’.

24.5.3.5.1.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.

24.5.3.5.2.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.

24.5.3.6.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.

24.5.3.6.1.'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.

24.5.3.6.2.'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).

24.5.3.7.Write (Program) Access

The User area can be programmed at any time after a reset until OTPC_MR.WRDIS has been set.

24.5.3.7.1.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.

24.5.3.7.2.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.

24.5.3.7.3.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).

24.5.3.7.4.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.

24.5.3.7.5.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.

24.5.3.8.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.

24.5.3.9."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).

24.5.3.10. "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.

24.5.4. 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 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.

24.5.5. 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.

24.5.6. 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

24.6. 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			REPAIR			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			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	MNT						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		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 ... 0x3F	Reserved									
0x40	OTPC_LRMR	31:24	KEY[15:8]							
		23:16	KEY[7:0]							
		15:8								
		7:0				EN			FREQ[1:0]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44 ... 0x4F	Reserved									
0x50	OTPC_UHC0R	31:24								
		23:16								
		15:8								
		7:0	JTAGDIS[7:0]							
0x54	OTPC_UHC1R	31:24								
		23:16							URFDIS	CPGDIS
		15:8	CLKDIS	CINVDIS				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

24.6.1. 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			REPAIR			KBSTOP	KBSTART
Access	W			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. To start a live repair, the key code must be 0x7364.

Bit 15 – REFRESH Refresh the Area

Value	Description
0	No effect.
1	Starts a refresh of the area.

Bit 12 – REPAIR Live Repair

Value	Description
0	No effect.
1	Starts OTP Live Repair.

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.

Value	Description
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.

24.6.2. 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			WRDIS	RDDIS
Access	R/W			R/W			R/W	R/W
Reset	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.

Bit 12 – KBDST Key Bus Destination

Value	Name	Description
0	TDES	The TDES is the destination of the key transfer.
1	AES	The AES 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.

Value	Description
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.

24.6.3. 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
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.

24.6.4. 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
	MNT						ONEF	HIDE
Access	R						R	R
Reset	0						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 15 – MNT Maintenance On-Going

Value	Description
0	The OTP live repair is not running.
1	The OTP live repair is running. The OTP maintenance is running.

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.

Value	Description
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.
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.

24.6.5. 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

24.6.6. 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

24.6.7. 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

24.6.8. 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, see 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.

Value	Description
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.

24.6.9. 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		INVLD[1:0]		LOCK	PACKET[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

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.

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

24.6.10. 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.

24.6.11. 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.

24.6.12. 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.

24.6.13. OTPC Live Repair Mode Register

Name: OTPC_LRMR
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	KEY[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
	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	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 31:16 – KEY[15:0] Programming Key
This field must be written with the correct key code (0x7364) to change the register value. This field cannot be read.

Bit 4 – EN Automatic Live Repair Enable

Value	Description
0	The automatic start of the OTP live repair is disabled.
1	The automatic start of the OTP live repair is enabled.

Bits 1:0 – FREQ[1:0] Automatic Live Repair Frequency

Value	Name	Description
0	ONE_DAY	The live repair is started every day.
1	FOUR_DAYS	The live repair is started every 4 days.
2	EIGHT_DAYS	The live repair is started every 8 days.
3	SIXTEEN_DAYS	The live repair is started every 16 days.

24.6.14. OTPC User Hardware Configuration 0 Register

Name: OTPC_UHC0R
Offset: 0x50
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on the 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
Access								
Reset								

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 7:0 – JTAGDIS[7:0] JTAG Disable

Value	Description
0	The JTAG is enabled.
Non-zero	The JTAG is disabled.

24.6.15. OTPC User Hardware Configuration 1 Register

Name: OTPC_UHC1R
Offset: 0x54
Reset: 0x00000000
Property: Read-only

Note: The reset value depends on the 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				SBCPGDIS	SBCLKDIS	SBCINVDIS
Access	R	R				R	R	R
Reset	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 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.
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.

24.6.16. 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 the 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.

24.6.17. 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).

24.6.18. 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)

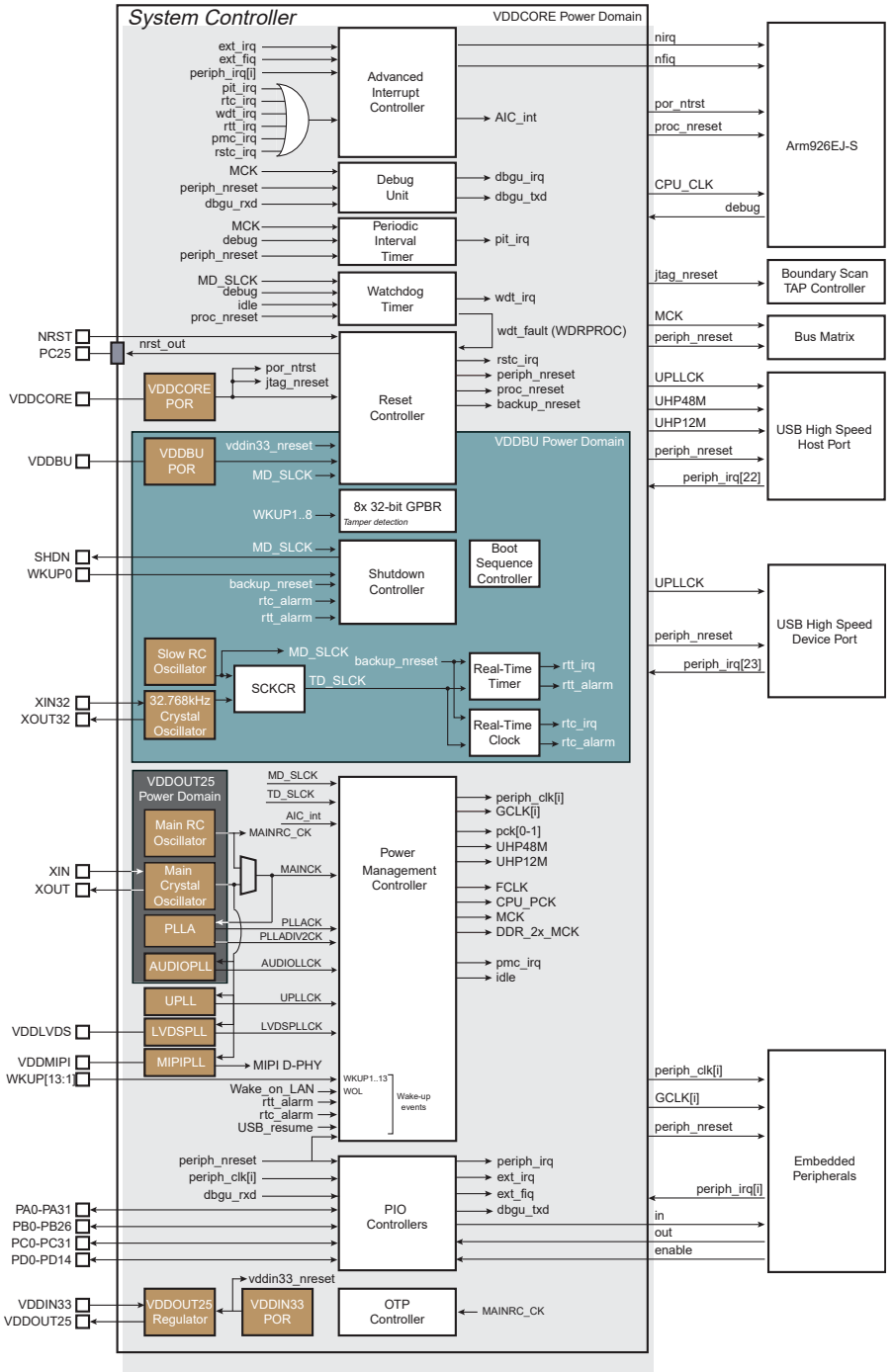
Value	Description
0	No write protection violation has occurred since the last read of OTPC_WPSR.
1	A write protection violation has occurred since the last read of OTPC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into WPVSRC.

System Controller

25. Overview

25.1. Block Diagram

Figure 25.1. System Controller Block Diagram



25.2. Components

- Peripherals powered by VDDBU:
  - 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)
- Peripherals powered by VDDCORE:
  - Watchdog Timer (WDT)
  - Chip Identifier (CHIPID)
  - OTP Memory Controller (OTPC)
  - Special Function Registers (SFR)
  - Clock Generator and Power Management Controller (PMC)
  - Parallel Input/Output Controller (PIO)
  - Periodic Interval Timer (PIT)

25.3. Product Dependencies

25.3.1. Clocks

The system controller peripherals are always on, except:

- Parallel Input/Output Controllers (PIOs) that have one clock control (PIOA) for all PIO controllers
- Special Function Registers (SFR)
- Periodic Interval Timer (PIT)

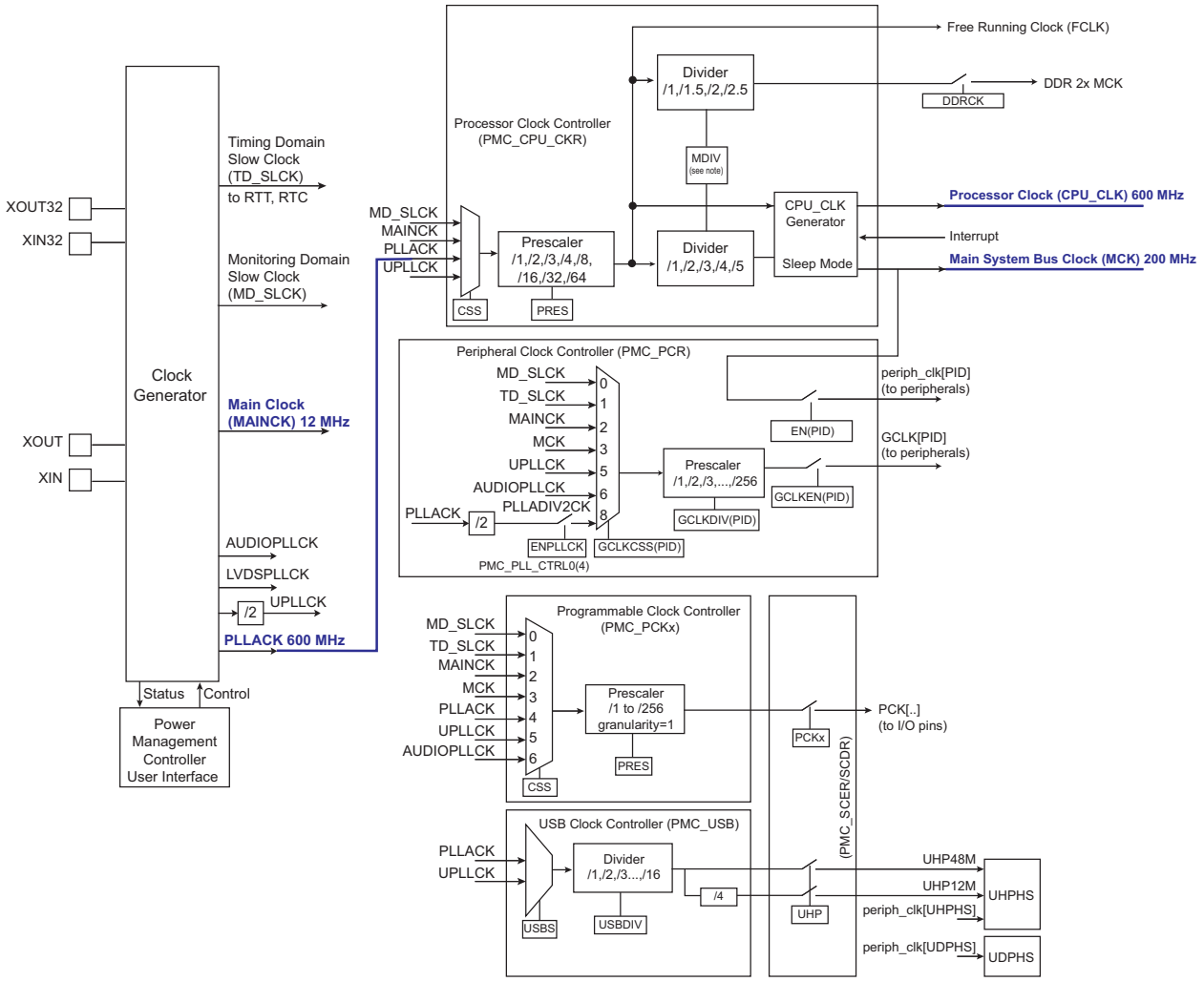
25.3.2. Changing System Frequencies

After ROM code execution, the system clock is set as follows:

- MAINCK is fed by the main RC oscillator, i.e., 12 MHz.
- PLLA is set to 1.2 GHz.
- CPU_CLK is set to 600 MHz.
- MCK is set to 200 MHz.

This is illustrated in the following figure.

Figure 25.2. PMC Rom Code Configuration



Note: MDIV should always be different from 0 when using DDR memories. If MDIV must be set to 0, first switch the DDR memories to Self-refresh mode and disable DDRCLK.

To avoid any overclocking during system clock modification, it is recommended to change system frequencies in two steps.

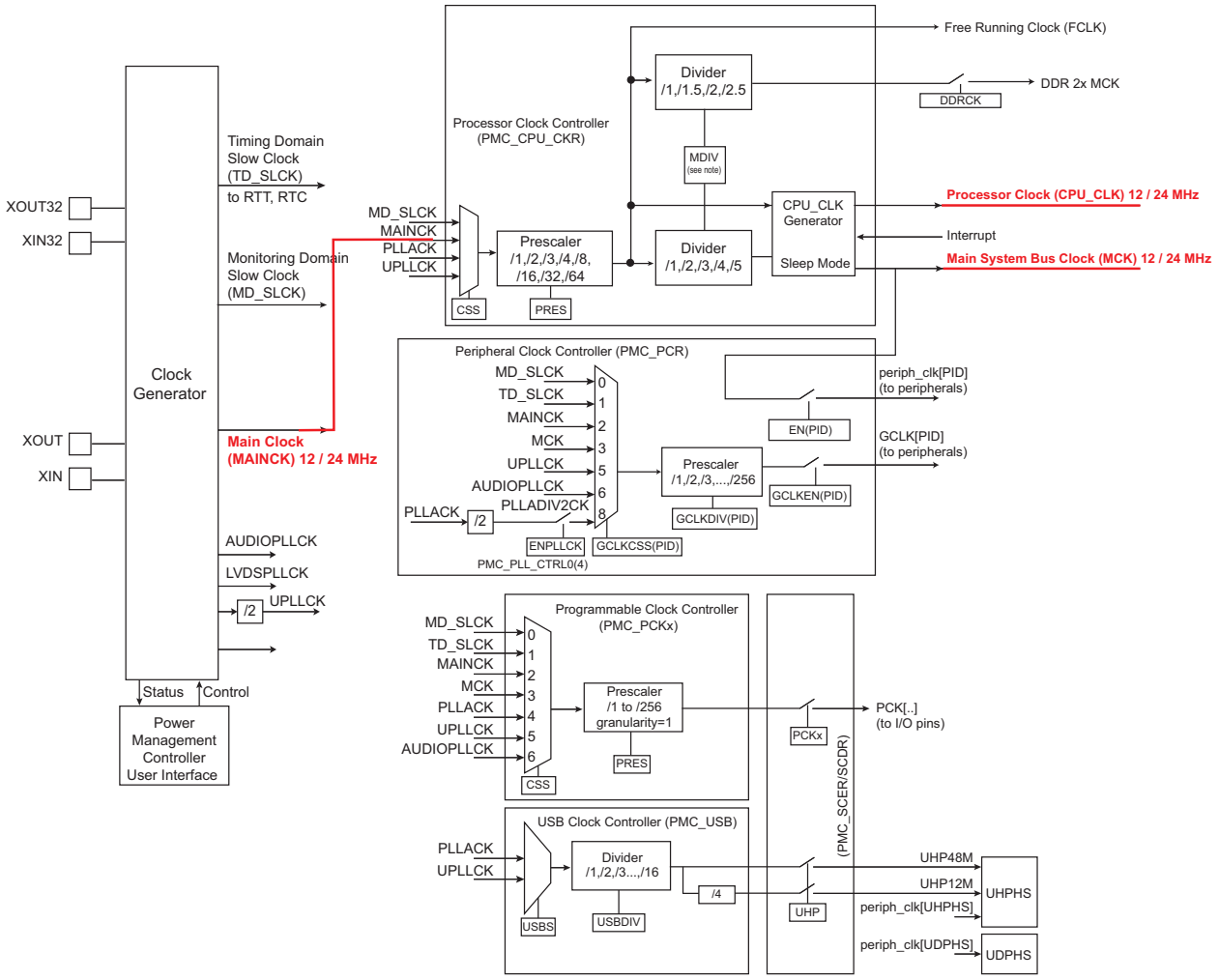
The first step leads to a known and basic intermediate state where all clocks are in a low-frequency range.

For example, the following sequence can be performed:

1. Set MCK to MAINCK, i.e., 12 MHz.
2. Set MAINCK to Crystal Oscillator, typically 24 MHz. MCK then runs at 24 MHz.

This intermediate state is illustrated in the following figure.

Figure 25.3. PMC Intermediate Configuration



Note: MDIV should always be different from 0 when using DDR memories. If MDIV must be set to 0, first switch the DDR memories to Self-refresh mode and disable DDRCLK.

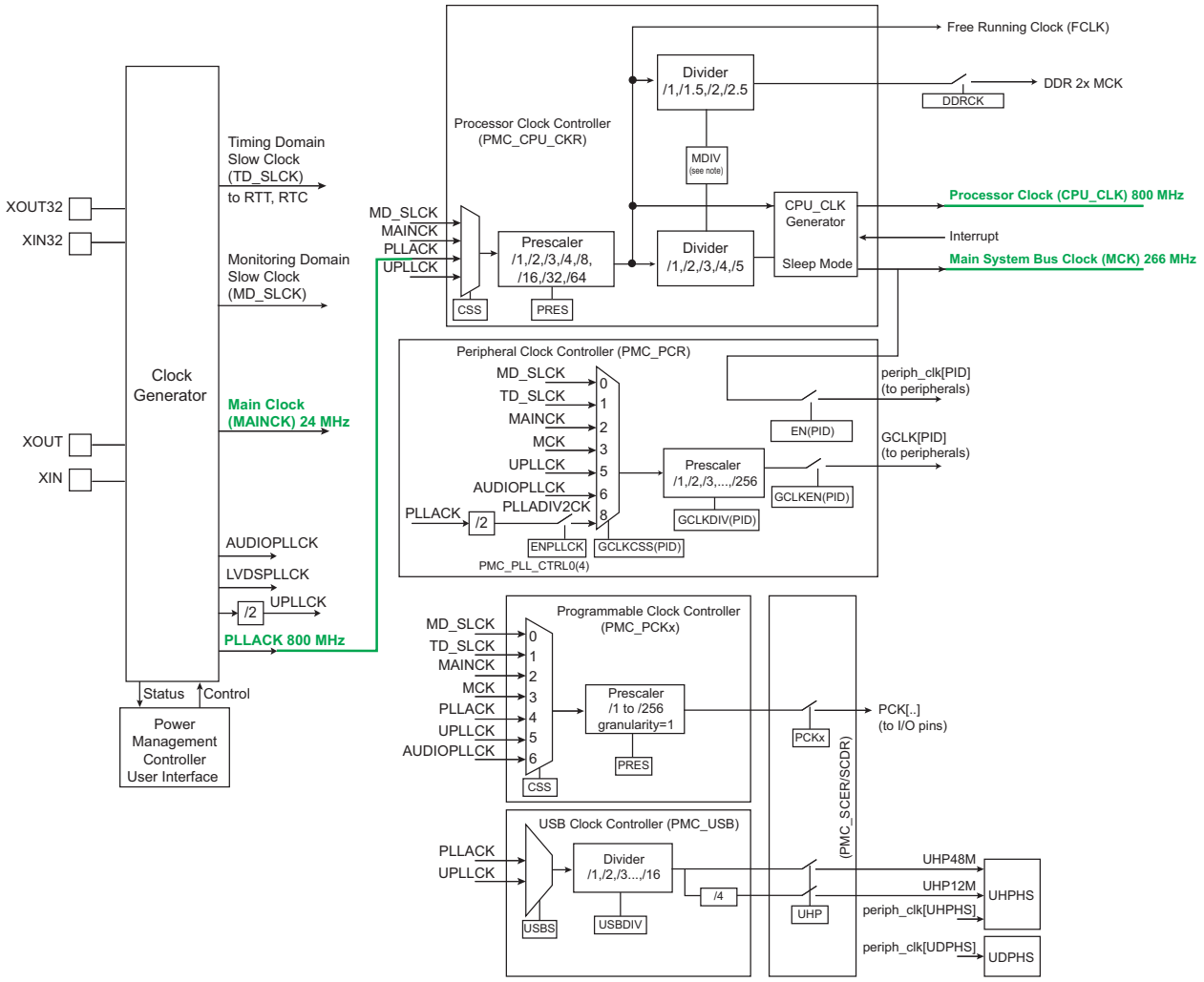
The second step leads to the final expected state.

Perform the following sequence:

1. Set PLL to 1.6 GHz, and set PLLACK to 800 MHz.
2. Set CPU_CLK to 800 MHz.
3. Set MCK to CPU_CLK/3, i.e., 266 MHz.

The final state is illustrated in the following figure.

Figure 25.4. PMC Final Configuration



Note: MDIV should always be different from 0 when using DDR memories. If MDIV must be set to 0, first switch the DDR memories to Self-refresh mode and disable DDRCLK.

25.3.3. Interrupts

Each peripheral has its own interrupt ID, except CHIP_ID, SYSCWP and GPBR, that have no interrupt line.

25.3.4. Reset

System controller peripherals are connected to the processor and peripherals reset line.

25.3.5. I/Os

For the applicable I/O type and power supply, refer to the table Pin Description.

25.3.6. Power Supplies

The 32.768 kHz crystal oscillator and the slow RC oscillator are powered by VDDBU.

The main crystal oscillator, the main RC oscillator, PLLA and AUDIOPLL are powered by VDDANA.

LVDSPLL and MIPIPLL are powered by VDDOUT25 (internally or externally).

UPLL is powered by VDDIN33.

25.4. Special Functions in SFR

None.

26. System Controller Write Protection (SYSCWP)

26.1. Functional Description

26.1.1. System Controller Peripheral Mapping

Table 26.1. 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
0x040-0x04C	Period Interval Timer	PIT
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

26.1.2. 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), Periodic Interval Timer (PIT), 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:

- WDT Control Register
- WDT Mode Register
- RSTC Mode Register
- SHDWC Mode Register
- SHDWC Wakeup Inputs Register
- PIT Mode 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

26.2. 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

26.2.1. 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).

26.2.2. 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.

27. Advanced Interrupt Controller (AIC)

27.1. Description

The Advanced Interrupt Controller (AIC) is an 8-level priority, individually maskable, vectored interrupt controller providing handling of up to one hundred and twenty-eight interrupt sources. It is designed to substantially reduce the software and real-time overhead in handling internal and external interrupts.

The AIC drives the nFIQ (fast interrupt request) and the nIRQ (standard interrupt request) inputs of an Arm processor. Inputs of the AIC are either internal peripheral interrupts or external interrupts coming from the product's pins.

The 8-level Priority Controller allows the user to define the priority for each interrupt source, thus permitting higher priority interrupts to be serviced even if a lower priority interrupt is being processed.

Internal interrupt sources can be programmed to be level-sensitive or edge-triggered. External interrupt sources can be programmed to be rising-edge or falling-edge triggered or high-level or low-level sensitive.

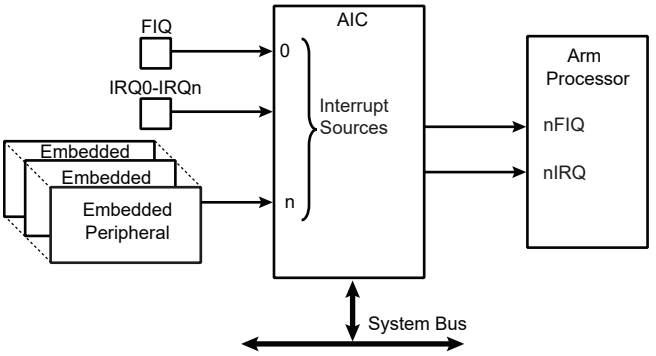
The fast-forcing feature redirects any internal or external interrupt source to provide a fast interrupt rather than a normal interrupt.

27.2. Embedded Characteristics

- Controls the Interrupt Lines (nIRQ and nFIQ) of an Arm Processor
- 61 Individually Maskable and Vectored Interrupt Sources
  - Source 0 is reserved for the fast interrupt input (FIQ)
  - Source 1 is reserved for system peripheral interrupts
  - Source 2 to Source 60 control up to 126 embedded peripheral interrupts or external interrupts
  - Programmable edge-triggered or level-sensitive internal sources
  - Programmable rising/falling edge-triggered or high/low level-sensitive external sources
- 8-level Priority Controller
  - Drives the normal interrupt of the processor
  - Handles priority of the interrupt sources 1 to 60
  - Higher priority interrupts can be served during service of lower priority interrupt
- Vectoring
  - Optimizes interrupt service routine branch and execution
  - One 32-bit vector register for all interrupt sources
  - Interrupt vector register reads the corresponding current interrupt vector or the current interrupt number
- Protect Mode
  - Easy debugging by preventing automatic operations when protect models are enabled
- Fast Forcing
  - Permits redirecting any normal interrupt source to the fast interrupt of the processor
- General Interrupt Mask
  - Provides processor synchronization on events without triggering an interrupt
- Register Write Protection

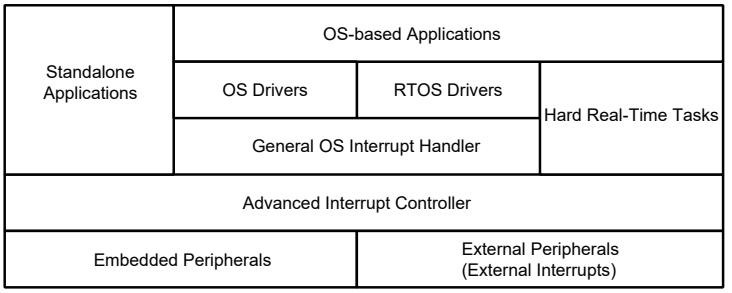
27.3. Block Diagram

Figure 27.1. AIC Block Diagram



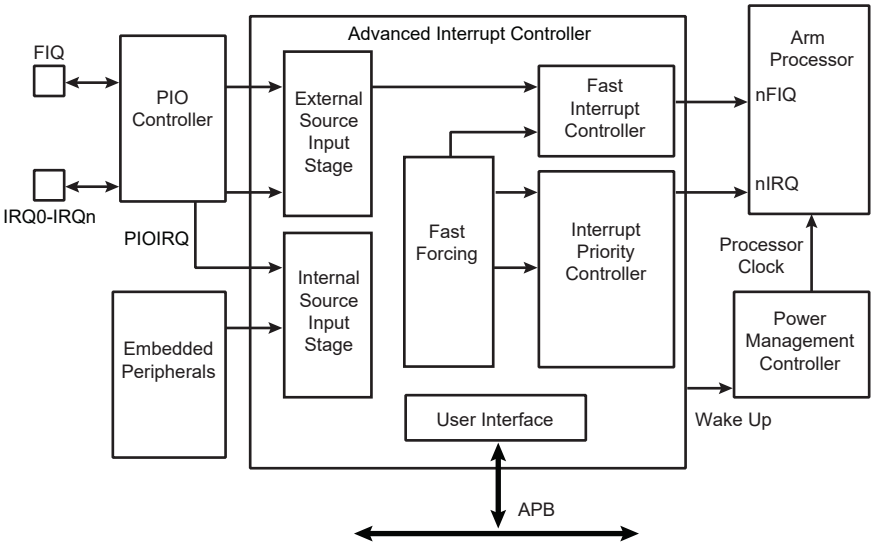
27.4. Application Block Diagram

Figure 27.2. AIC Application Block Diagram



27.5. Detailed Block Diagram

Figure 27.3. AIC Detailed Block Diagram



27.6. I/O Line Description

Table 27.1. I/O Line Description

Pin Name	Pin Description	Type
FIQ	Fast Interrupt	Input
IRQ0–IRQn	Interrupt 0–Interrupt n	Input

27.7. Product Dependencies

27.7.1. I/O Lines

The interrupt signals FIQ and IRQ0 to IRQn are normally multiplexed through the PIO controllers. Depending on the features of the PIO controller used in the product, the pins must be programmed in accordance with their assigned interrupt functions. This is not applicable when the PIO controller used in the product is transparent on the input path.

27.7.2. Power Management

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

The assertion of the AIC outputs, either nIRQ or nFIQ, wakes up the Arm processor while it is in Idle mode. The General Interrupt Mask feature enables the AIC to wake up the processor without asserting the interrupt line of the processor, thus providing synchronization of the processor on an event.

27.7.3. Interrupt Sources

FIQ always drives Interrupt Source 0.

The System Controller interrupt drives Interrupt Source 1.

The System Controller interrupt is the result of the OR-wiring of the System Controller interrupt lines. When a System Controller interrupt occurs, the service routine must first distinguish the cause of the interrupt. This is performed by reading successively the status registers of the System Controller peripherals.

Interrupt sources 2 to 60 can either be connected to the interrupt outputs of an embedded user peripheral, or to external interrupt lines. The external interrupt lines can be connected either directly or through the PIO Controller.

PIO controllers are considered as user peripherals in the scope of interrupt handling. Accordingly, the PIO controller interrupt lines are connected to interrupt sources 2 to 60.

The peripheral identification defined at the product level corresponds to the interrupt source number (as well as the bit number controlling the clock of the peripheral). Consequently, to simplify the description of the functional operations and the user interface, the interrupt sources are named FIQ, SYS, and PID2 to PID60.

27.8. Functional Description

27.8.1. Interrupt Source Control

27.8.1.1. Interrupt Source Mode

The AIC independently programs each interrupt source. The SRCTYPE field of the Source Mode register (AIC_SMR) selects the interrupt condition of the interrupt source selected by the INTSEL field of the Source Select register (AIC_SSR).

Note: Configuration registers such as AIC_SMR and AIC_SSR return the values corresponding to the interrupt source selected by INTSEL.

The internal interrupt sources wired on the interrupt outputs of the embedded peripherals can be programmed either in Level-Sensitive mode or in Edge-Triggered mode. The active level of the internal interrupts is not important for the user.

The external interrupt sources can be programmed either in High Level-Sensitive or Low Level-Sensitive modes, or in Rising Edge-Triggered or Negative Edge-Triggered modes.

27.8.1.2. Interrupt Source Enabling

Each interrupt source, including the FIQ in source 0, can be enabled or disabled by using the command registers Interrupt Enable Command register (AIC_IECR) and Interrupt Disable Command register (AIC_IDCR). The interrupt mask of the selected interrupt source can be read in the Interrupt Mask register (AIC_IMR). A disabled interrupt does not affect servicing of other interrupts.

27.8.1.3. Interrupt Clearing and Setting

All interrupt sources programmed to be edge-triggered (including the FIQ in source 0) can be individually set or cleared by writing respectively the Interrupt Set Command register (AIC_ISCR) and Interrupt Clear Command register (AIC_ICCR). Clearing or setting interrupt sources programmed in Level-Sensitive mode has no effect.

The clear operation is perfunctory, as the software must perform an action to reset the “memorization” circuitry activated when the source is programmed in Edge-Triggered mode. However, the set operation is available for auto-test or software debug purposes. It can also be used to execute an AIC-implementation of a software interrupt.

The AIC features an automatic clear of the current interrupt when AIC_IVR (Interrupt Vector register) is read. Only the interrupt source being detected by the AIC as the current interrupt is affected by this operation. (See the section Priority Controller.) The automatic clear reduces the operations required by the interrupt service routine entry code to read AIC_IVR. Note that the automatic interrupt clear is disabled if the interrupt source has the Fast Forcing feature enabled, as it is considered uniquely as an FIQ source. (For further details, see the section ”Fast Forcing”).

The automatic clear of interrupt source 0 is performed when the FIQ Vector register (AIC_FVR) is read.

27.8.1.4. Interrupt Status

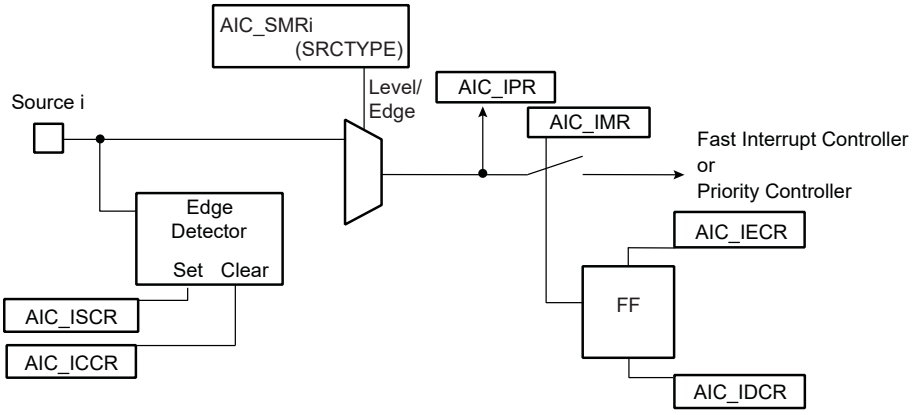
Interrupt Pending registers (AIC_IPR) represent the state of the interrupt lines, whether they are masked or not. AIC_IMR can be used to define the mask of the interrupt lines.

The Interrupt Status register (AIC_ISR) reads the number of the current interrupt (see the section Priority Controller) and the Core Interrupt Status register (AIC_CISR) gives an image of the nIRQ and nFIQ signals driven on the processor.

Each status referred to above can be used to optimize the interrupt handling of the systems.

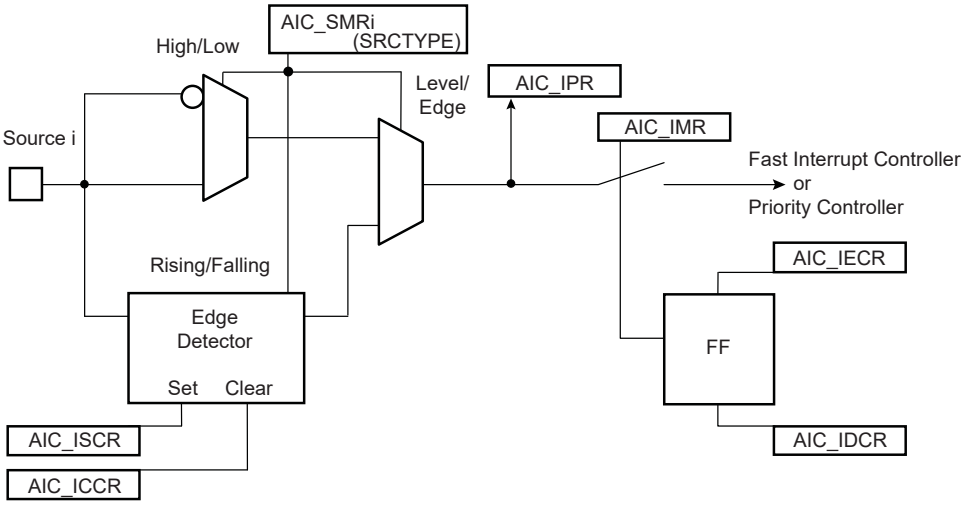
27.8.1.5.Internal Interrupt Source Input Stage

Figure 27.4. Internal Interrupt Source Input Stage



27.8.1.6.External Interrupt Source Input Stage

Figure 27.5. External Interrupt Source Input Stage



27.8.2. Interrupt Latencies

Global interrupt latencies depend on several parameters, including:

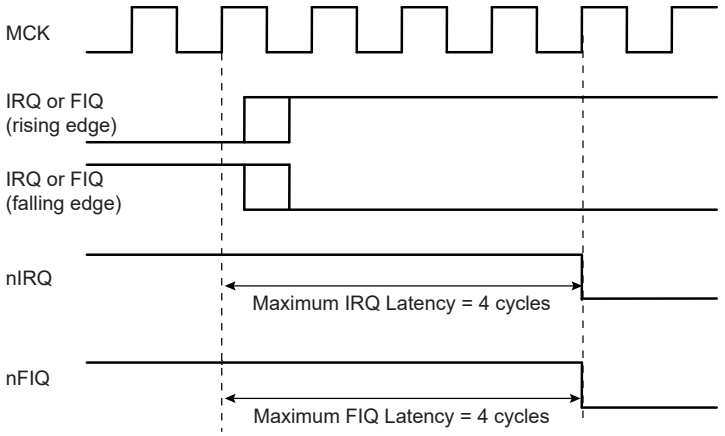
- The time the software masks the interrupts
- Occurrence, either at the processor level or at the AIC level
- The execution time of the instruction in progress when the interrupt occurs
- The treatment of higher priority interrupts and the resynchronization of the hardware signals

This section addresses hardware resynchronizations only. It gives details about the latency times between the events on an external interrupt leading to a valid interrupt (edge or level) or the assertion of an internal interrupt source and the assertion of the nIRQ or nFIQ line on the processor. The resynchronization time depends on the programming of the interrupt source and on its type (internal or external). For the standard interrupt, resynchronization times are given assuming there is no higher priority in progress.

The PIO Controller multiplexing has no effect on the interrupt latencies of the external interrupt sources.

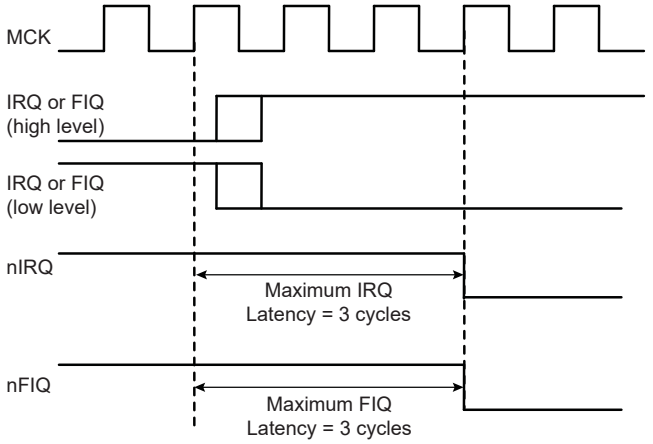
27.8.2.1.External Interrupt Edge Triggered Source

Figure 27.6. External Interrupt Edge Triggered Source



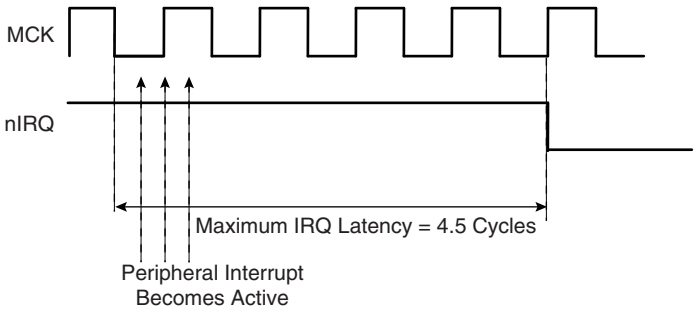
27.8.2.2.External Interrupt Level Sensitive Source

Figure 27.7. External Interrupt Level Sensitive Source



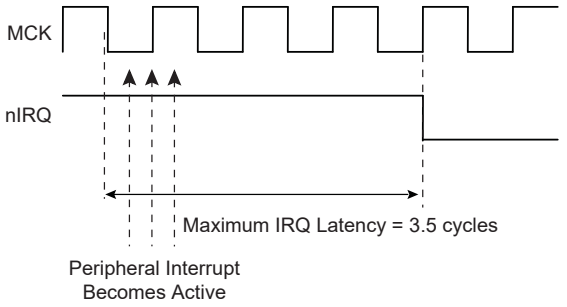
27.8.2.3.Internal Interrupt Edge Triggered Source

Figure 27.8. Internal Interrupt Edge Triggered Source



27.8.2.4. Internal Interrupt Level Sensitive Source

Figure 27.9. Internal Interrupt Level Sensitive Source



27.8.3. Normal Interrupt

27.8.3.1. Priority Controller

An 8-level priority controller drives the nIRQ line of the processor, depending on the interrupt conditions occurring on the interrupt sources 1 to 60 (except for those programmed in Fast Forcing).

Each interrupt source has a programmable priority level of 7 to 0, which is user-definable by writing AIC_SMR.PRIOR. Level 7 is the highest priority and level 0 the lowest.

As soon as an interrupt condition occurs, as defined by AIC_SMR.SRCTYPE, the nIRQ line is asserted. As a new interrupt condition might have happened on other interrupt sources since the nIRQ has been asserted, the priority controller determines the current interrupt at the time AIC_IVR is read. The read of AIC_IVR is the entry point of the interrupt handling which allows the AIC to consider that the interrupt has been taken into account by the software.

The current priority level is defined as the priority level of the current interrupt.

If several interrupt sources of equal priority are pending and enabled when AIC_IVR is read, the interrupt with the lowest interrupt source number is serviced first.

The nIRQ line can be asserted only if an interrupt condition occurs on an interrupt source with a higher priority. If an interrupt condition happens (or is pending) during the interrupt treatment in progress, it is delayed until the software indicates to the AIC the end of the current service by writing AIC_EOICR (End of Interrupt Command register). The write of AIC_EOICR is the exit point of the interrupt handling.

27.8.3.2. Interrupt Nesting

The priority controller utilizes interrupt nesting in order for the high priority interrupt to be handled during the service of lower priority interrupts. This requires the interrupt service routines of the lower interrupts to re-enable the interrupt at the processor level.

When an interrupt of a higher priority happens during an already occurring interrupt service routine, the nIRQ line is re-asserted. If the interrupt is enabled at the core level, the current execution is interrupted and the new interrupt service routine should read AIC_IVR. At this time, the current interrupt number and its priority level are pushed into an embedded hardware stack, so that they are saved and restored when the higher priority interrupt servicing is finished and AIC_EOICR is written.

The AIC is equipped with an 8-level wide hardware stack in order to support up to eight interrupt nestings to match the eight priority levels.

27.8.3.3. Interrupt Vectoring

The interrupt handler address corresponding to the interrupt source selected by the INTSEL field can be stored in AIC_SVR (Source Vector register). When the processor reads AIC_IVR, the value written into AIC_SVR corresponding to the current interrupt is returned. Optionally, the AIC_IVR

register can return the current interrupt number instead. This can be defined separately for each interrupt source using the AIC SVR Return Enable Register (AIC_SVRRER) and AIC SVR Return Disable Register (AIC_SVRRDR).

This feature offers a way to branch in one single instruction to the handler corresponding to the current interrupt, as AIC_IVR is mapped at the absolute address 0xFFFFF100 and thus accessible from the Arm interrupt vector at address 0x00000018 through the following instruction:

```
LDR PC,[PC,# -&F20]
```

When the processor executes this instruction, it loads the read value in AIC_IVR in its program counter, thus branching the execution on the correct interrupt handler.

27.8.3.4. Interrupt Handlers

This section gives an overview of the fast interrupt handling sequence when using the AIC. It is assumed that the programmer understands the architecture of the Arm processor, and especially the Processor Interrupt modes and the associated status bits.

It is assumed that:

1. The AIC has been programmed, AIC_SVR registers are loaded with corresponding interrupt service routine addresses and interrupts are enabled.
2. The instruction at the Arm interrupt exception vector address is required to work with the vectoring

   ```
   LDR PC, [PC, # -&F20]
   ```

When nIRQ is asserted, if the bit "I" of CPSR is 0, the sequence is as follows:

1. The CPSR is stored in SPSR_irq, the current value of the Program Counter is loaded in the Interrupt link register (R14_irq) and the Program Counter (R15) is loaded with 0x18. In the following cycle during fetch at address 0x1C, the Arm core adjusts R14_irq, decrementing it by four.
2. The Arm core enters Interrupt mode, if it has not already done so.
3. When the instruction loaded at address 0x18 is executed, the program counter is loaded with the value read in AIC_IVR. Reading AIC_IVR has the following effects:
   - Sets the current interrupt to be the pending and enabled interrupt with the highest priority. The current level is the priority level of the current interrupt.
   - De-asserts the nIRQ line on the processor. Even if vectoring is not used, AIC_IVR must be read in order to de-assert nIRQ.
   - Automatically clears the interrupt, if it has been programmed to be edge-triggered.
   - Pushes the current level and the current interrupt number on to the stack.
   - Returns the value written in AIC_SVR corresponding to the current interrupt.
4. The previous step has the effect of branching to the corresponding interrupt service routine. This should start by saving the link register (R14_irq) and SPSR_IRQ. The link register must be decremented by four when it is saved if it is to be restored directly into the program counter at the end of the interrupt. For example, the instruction `SUB PC, LR, #4` may be used.
5. Further interrupts can then be unmasked by clearing the "I" bit in CPSR, allowing re-assertion of the nIRQ to be taken into account by the core. This can happen if an interrupt with a higher priority than the current interrupt occurs.
6. The interrupt handler can then proceed as required, saving the registers that will be used and restoring them at the end. During this phase, an interrupt of higher priority than the current level will restart the sequence from step 1.
   Note: If the interrupt is programmed to be level-sensitive, the source of the interrupt must be cleared during this phase.

7. The “I” bit in CPSR must be set in order to mask interrupts before exiting to ensure that the interrupt is completed in an orderly manner.
8. AIC_EOICR must be written in order to indicate to the AIC that the current interrupt is finished. This causes the current level to be popped from the stack, restoring the previous current level if one exists on the stack. If another interrupt is pending, with lower or equal priority than the old current level but with higher priority than the new current level, the nIRQ line is re-asserted, but the interrupt sequence does not immediately start because the “I” bit is set in the core. SPSR_irq is restored. Finally, the saved value of the link register is restored directly into the PC. This has the effect of returning from the interrupt to whatever was being executed before, and of loading the CPSR with the stored SPSR, masking or unmasking the interrupts depending on the state saved in SPSR_irq.
   Note: The “I” bit in SPSR is significant. If it is set, it indicates that the Arm core was on the verge of masking an interrupt when the mask instruction was interrupted. Hence, when SPSR is restored, the mask instruction is completed (interrupt is masked).

27.8.4. Fast Interrupt

27.8.4.1. Fast Interrupt Source

Interrupt source 0 is the only source which can raise a fast interrupt request to the processor except if fast forcing is used. Interrupt source 0 is generally connected to a FIQ pin of the product, either directly or through a PIO Controller.

27.8.4.2. Fast Interrupt Control

The fast interrupt logic of the AIC has no priority controller. The mode of interrupt source 0 is programmed with AIC_SMR and INTSEL = 0; the PRIOR field of this register is not used even if it reads what has been written. AIC_SMR.SRCTYPE enables programming the fast interrupt source to be rising-edge triggered or falling-edge triggered or high-level sensitive or low-level sensitive.

Writing 0x1 in AIC_IECR and AIC_IDCR respectively enables and disables the fast interrupt when INTSEL = 0. Bit 0 of AIC_IMR indicates whether the fast interrupt is enabled or disabled.

27.8.4.3. Fast Interrupt Vectoring

The fast interrupt handler address can be stored through AIC_SVR. The value written into this register when INTSEL = 0 is returned when the processor reads AIC_FVR (FIQ Vector register). This offers a way to branch in one single instruction to the interrupt handler, as AIC_FVR is mapped at the absolute address 0xFFFF F104 and thus accessible from the Arm fast interrupt vector at address 0x0000 001C through the following instruction:

```
LDR PC,[PC,# -&F20]
```

When the processor executes this instruction, it loads the value read in AIC_FVR in its program counter, thus branching the execution on the fast interrupt handler. It also automatically performs the clear of the fast interrupt source if it is programmed in Edge-Triggered mode.

27.8.4.4. Fast Interrupt Handlers

This section gives an overview of the fast interrupt handling sequence when using the AIC. It is assumed that the programmer understands the architecture of the Arm processor, and especially the Processor Interrupt modes and associated status bits.

Assuming that:

1. The AIC has been programmed, AIC_SVR is loaded with the fast interrupt service routine address, and interrupt source 0 is enabled.
2. The Instruction at address 0x1C (FIQ exception vector address) is required to vector the fast interrupt:
   ```
   LDR PC,[PC, # -&F20]
   ```
3. The user does not need nested fast interrupts.

When nFIQ is asserted, if bit “F” of CPSR is 0, the sequence is:

1. The CPSR is stored in SPSR_fiq, the current value of the program counter is loaded in the FIQ link register (R14_FIQ) and the program counter (R15) is loaded with 0x1C. In the following cycle, during fetch at address 0x20, the Arm core adjusts R14_fiq, decrementing it by four.
2. The Arm core enters FIQ mode.
3. When the instruction loaded at address 0x1C is executed, the program counter is loaded with the value read in AIC_FVR. Reading AIC_FVR has the effect of automatically clearing the fast interrupt, if it has been programmed to be edge-triggered. In this case only, it de-asserts the nFIQ line on the processor.

```
FIQ_Handler_Branch
   mov r14, pc
   bx r0
```

4. The previous step enables branching to the corresponding interrupt service routine. It is not necessary to save the link register R14_fiq and SPSR_fiq if nested fast interrupts are not needed.
5. The Interrupt Handler can then proceed as required. It is not necessary to save registers R8 to R13 because the FIQ mode has its own dedicated registers and registers R8 to R13 are banked. The other registers, R0 to R7, must be saved before being used, and restored at the end (before the next step).
   Note: If the fast interrupt is programmed to be level-sensitive, the source of the interrupt must be cleared during this phase in order to de-assert interrupt source 0.
6. Finally, Link register R14_fiq is restored into the PC after decrementing it by four (with instruction `SUB PC, LR, #4` for example). This has the effect of returning from the interrupt to whatever was being executed before, loading the CPSR with the SPSR and masking or unmasking the fast interrupt depending on the state saved in the SPSR.
   Note: The “F” bit in SPSR is significant. If it is set, it indicates that the Arm core was just about to mask FIQ interrupts when the mask instruction was interrupted. Hence, when the SPSR is restored, the interrupted instruction is completed (FIQ is masked).

Another way to handle the fast interrupt is to map the interrupt service routine at the address of the Arm vector 0x1C. This method does not use vectoring, so that reading AIC_FVR must be performed at the very beginning of the handler operation. However, this method saves the execution of a branch instruction.

27.8.4.5. Fast Forcing

The Fast Forcing feature of the advanced interrupt controller provides redirection of any normal Interrupt source on the fast interrupt controller.

Fast Forcing is enabled or disabled by writing to the Fast Forcing Enable register (AIC_FFER) and the Fast Forcing Disable register (AIC_FFDR). Writing to these registers results in an update of the Fast Forcing Status register (AIC_FFSR) that controls the feature for each internal or external interrupt source.

When Fast Forcing is disabled, the interrupt sources are handled as described in the previous sections.

When Fast Forcing is enabled, the edge/level programming and, in certain cases, edge detection of the interrupt source is still active but the source cannot trigger a normal interrupt to the processor and is not seen by the priority handler.

If the interrupt source is programmed in Level-Sensitive mode and an active level is sampled, Fast Forcing results in the assertion of the nFIQ line to the core.

If the interrupt source is programmed in Edge-Triggered mode and an active edge is detected, Fast Forcing results in the assertion of the nFIQ line to the core.

The Fast Forcing feature does not affect the Source 0 pending bit in AIC_IPR.

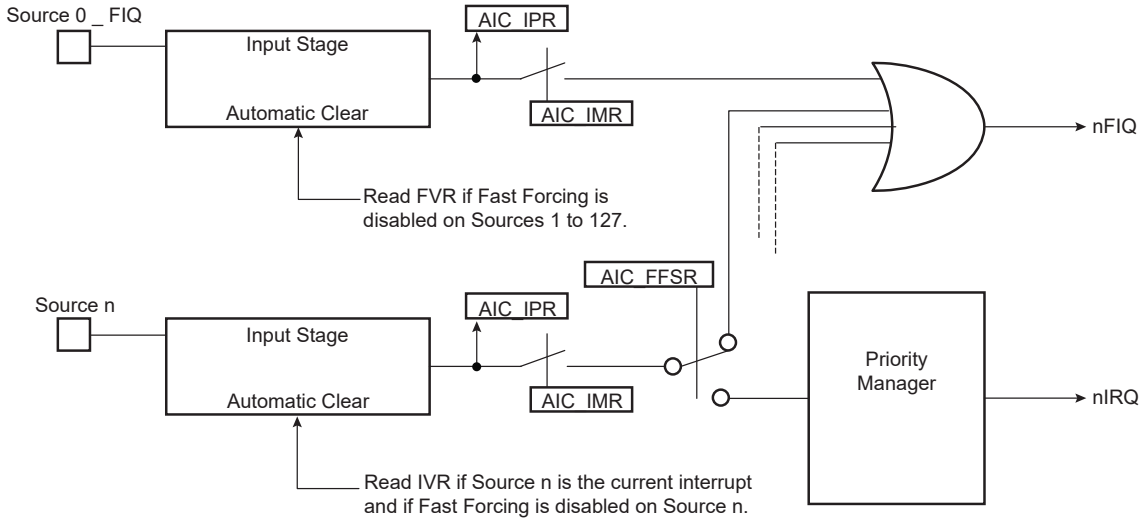
The FIQ Vector register (AIC_FVR) reads the contents of the Source Vector register (AIC_SVR), whatever the source of the fast interrupt may be. The read of the FVR does not clear Source 0 when the fast forcing feature is used and the interrupt source should be cleared by writing to AIC_ICCR.

All enabled and pending interrupt sources that have the fast forcing feature enabled and that are programmed in Edge-Triggered mode must be cleared by writing to the Interrupt Clear Command register. In doing so, they are cleared independently and thus lost interrupts are prevented.

The read of AIC_IVR does not clear the source that has the fast forcing feature enabled.

Source 0, reserved to the fast interrupt, continues operating normally and becomes one of the Fast Interrupt sources.

Figure 27.10. Fast Forcing



27.8.5. Protect Mode

The Protect mode is used to read the Interrupt Vector register without performing the associated automatic operations. This is necessary when working with a debug system. When a debugger, working either with a Debug Monitor or the Arm processor's ICE, stops the applications and updates the opened windows, it might read the AIC User Interface and thus the IVR. This has adverse consequences:

- If an enabled interrupt with a higher priority than the current one is pending, it is stacked.
- If there is no enabled pending interrupt, the spurious vector is returned.

In either case, an End of Interrupt command is necessary to acknowledge and restore the context of the AIC. This operation is generally not performed by the debug system, as the debug system would become strongly intrusive and cause the application to enter an undesired state.

This is avoided by using the Protect mode. Writing PROT in the Debug Control register (AIC_DCR) at 0x1 enables the Protect mode.

When the Protect mode is enabled, the AIC performs interrupt stacking only when a write access is performed on AIC_IVR. Therefore, the Interrupt Service Routines must write (arbitrary data) to AIC_IVR just after reading it. The new context of the AIC, including the value of AIC_ISR, is updated with the current interrupt only when AIC_IVR is written.

An AIC_IVR read on its own (e.g., by a debugger) modifies neither the AIC context nor AIC_ISR. Extra AIC_IVR reads perform the same operations. However, it is recommended to not stop the processor between the read and the write of AIC_IVR of the interrupt service routine to make sure the debugger does not modify the AIC context.

To summarize, in normal operating mode, the read of AIC_IVR performs the following operations within the AIC:

1. Calculates active interrupt (higher than current or spurious).
2. Determines and returns the vector of the active interrupt.
3. Memorizes the interrupt.
4. Pushes the current priority level onto the internal stack.
5. Acknowledges the interrupt.

However, while the Protect mode is activated, only operations 1 to 3 are performed when AIC_IVR is read. Operations 4 and 5 are only performed by the AIC when AIC_IVR is written.

Software that has been written and debugged using the Protect mode runs correctly in normal mode without modification. However, in normal mode, the AIC_IVR write has no effect and can be removed to optimize the code.

27.8.6. Spurious Interrupt

The AIC features a protection against spurious interrupts. A spurious interrupt is defined as being the assertion of an interrupt source long enough for the AIC to assert the nIRQ, but no longer present when AIC_IVR is read. This is most prone to occur when:

- An external interrupt source is programmed in Level-Sensitive mode and an active level occurs for only a short time.
- An internal interrupt source is programmed in level-sensitive and the output signal of the corresponding embedded peripheral is activated for a short time (as is the case for the watchdog).
- An interrupt occurs just a few cycles before the software begins to mask it, thus resulting in a pulse on the interrupt source.

The AIC detects a spurious interrupt at the time AIC_IVR is read while no enabled interrupt source is pending. When this happens, the AIC returns the value stored by the programmer in the Spurious Vector register (AIC_SPU). The programmer must store the address of a spurious interrupt handler in AIC_SPU as part of the application, to enable an as fast as possible return to the normal execution flow. This handler writes in AIC_EOICR and performs a return from interrupt.

27.8.7. General Interrupt Mask

The AIC features a General Interrupt Mask bit (AIC_DCR.GMSK) to prevent interrupts from reaching the processor. Both the nIRQ and the nFIQ lines are driven to their inactive state if AIC_DCR.GMSK is set. However, this mask does not prevent waking up the processor if it has entered Idle mode. This function facilitates synchronizing the processor on a next event and, as soon as the event occurs, performs subsequent operations without having to handle an interrupt. It is strongly recommended to use this mask with caution.

27.8.8. Register Write Protection

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

If a write access to a write-protected register is detected, the WPVS flag in the AIC Write Protection Status Register (AIC_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 AIC_WPSR.

The following registers can be write-protected:

- AIC Source Mode Register
- AIC Source Vector Register
- AIC Spurious Interrupt Vector Register
- AIC Debug Control Register

27.9. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	AIC_SSR	31:24								
		23:16								
		15:8								
		7:0		INTSEL[6:0]						
0x04	AIC_SMR	31:24								
		23:16								
		15:8								
		7:0		SRCTYPE[1:0]				PRIOR[2:0]		
0x08	AIC_SVR	31:24	VECTOR[31:24]							
		23:16	VECTOR[23:16]							
		15:8	VECTOR[15:8]							
		7:0	VECTOR[7:0]							
0x0C ... 0x0F	Reserved									
0x10	AIC_IVR	31:24	IRQV[31:24]							
		23:16	IRQV[23:16]							
		15:8	IRQV[15:8]							
		7:0	IRQV[7:0]							
0x14	AIC_FVR	31:24	FIQV[31:24]							
		23:16	FIQV[23:16]							
		15:8	FIQV[15:8]							
		7:0	FIQV[7:0]							
0x18	AIC_ISR	31:24								
		23:16								
		15:8								
		7:0		IRQID[6:0]						
0x1C ... 0x1F	Reserved									
0x20	AIC_IPR0	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	SYS	FIQ
0x24	AIC_IPR1	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
0x28	AIC_IPR2	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
0x2C	AIC_IPR3	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
0x30	AIC_IMR	31:24								
		23:16								
		15:8								
		7:0								INTM
0x34	AIC_CISR	31:24								
		23:16								
		15:8								
		7:0							NIRQ	NFIQ
0x38	AIC_EOICR	31:24								
		23:16								
		15:8								
		7:0								ENDIT

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	AIC_SPU	31:24	SIVR[31:24]							
		23:16	SIVR[23:16]							
		15:8	SIVR[15:8]							
		7:0	SIVR[7:0]							
0x40	AIC_IECR	31:24								
		23:16								
		15:8								
		7:0								INTEN
0x44	AIC_IDCR	31:24								
		23:16								
		15:8								
		7:0								INTD
0x48	AIC_ICCR	31:24								
		23:16								
		15:8								
		7:0								INTCLR
0x4C	AIC_ISCR	31:24								
		23:16								
		15:8								
		7:0								INTSET
0x50	AIC_FFER	31:24								
		23:16								
		15:8								
		7:0								FFEN
0x54	AIC_FFDR	31:24								
		23:16								
		15:8								
		7:0								FFDIS
0x58	AIC_FFSR	31:24								
		23:16								
		15:8								
		7:0								FFS
0x5C ... 0x5F	Reserved									
0x60	AIC_SVRRER	31:24								
		23:16								
		15:8								
		7:0								SVRREN
0x64	AIC_SVRRDR	31:24								
		23:16								
		15:8								
		7:0								SVRRDIS
0x68	AIC_SVRRSR	31:24								
		23:16								
		15:8								
		7:0								SVRRS
0x6C	AIC_DCR	31:24								
		23:16								
		15:8								
		7:0							GMSK	PROT
0x70 ... 0xE3	Reserved									
0xE4	AIC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xE8	AIC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

27.9.1. AIC Source Select Register

Name: AIC_SSR
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
		INTSEL[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 6:0 – INTSEL[6:0] Interrupt Line Selection
0–60 = Selects the interrupt line to handle.
See the section Interrupt Source Mode.

27.9.2. AIC Source Mode Register

Name: AIC_SMR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC 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
		SRCTYPE[1:0]				PRIOR[2:0]		
Access		R/W	R/W			R/W	R/W	R/W
Reset		0	0			0	0	0

Bits 6:5 – SRCTYPE[1:0] Interrupt Source Type
The active level or edge is not programmable for the internal interrupt source selected by INTSEL.

Value	Name	Description
0	INT_LEVEL_SENSITIVE	High-level sensitive for internal source. Low-level sensitive for external source.
1	EXT_NEGATIVE_EDGE	Negative-edge triggered for external source.
2	EXT_HIGH_LEVEL	High-level sensitive for internal source. High-level sensitive for external source.
3	EXT_POSITIVE_EDGE	Positive-edge triggered for external source.

Bits 2:0 – PRIOR[2:0] Priority Level
Programs the priority level of the source selected by INTSEL except FIQ source (source 0).
The priority level can be between 0 (lowest) and 7 (highest).
The priority level is not used for the FIQ.

27.9.3. AIC Source Vector Register

Name: AIC_SVR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	VECTOR[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
	VECTOR[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
	VECTOR[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
	VECTOR[7:0]							
Access	R/W	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 – VECTOR[31:0] Source Vector
The user may store in this register the address of the corresponding handler for the interrupt source selected by INTSEL.

27.9.4. AIC Interrupt Vector Register

Name: AIC_IVR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	IRQV[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
	IRQV[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
	IRQV[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
	IRQV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IRQV[31:0] Interrupt Vector Register
Contains the vector programmed by the user in AIC_SVR or the interrupt index corresponding to the current interrupt. (See the sections AIC_SVRRER, AIC_SVRRDR and AIX_SVRRSR.)
AIC_SVR is indexed using the current interrupt number when AIC_IVR is read.
When there is no current interrupt, AIC_IVR reads the value stored in AIC_SPU.

27.9.5. AIC FIQ Vector Register

Name: AIC_FVR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	FIQV[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
	FIQV[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
	FIQV[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
	FIQV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FIQV[31:0] FIQ Vector Register
Contains the vector programmed by the user in AIC_SVR when INTSEL = 0. When there is no fast interrupt, AIC_FVR reads the value stored in AIC_SPU.

27.9.6. AIC Interrupt Status Register

Name: AIC_ISR
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
		IRQID[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:0 – IRQID[6:0] Current Interrupt Identifier
Returns the current interrupt source number.

27.9.7. AIC Interrupt Pending Register 0

Name: AIC_IPR0
Offset: 0x20
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.

Bit	31	30	29	28	27	26	25	24
	PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
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
	PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
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
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
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
	PID7	PID6	PID5	PID4	PID3	PID2	SYS	FIQ
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 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 Interrupt Pending

PID2...PID31 refer to the identifiers as defined in the Peripheral Identifiers section.

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bit 1 – SYS Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bit 0 – FIQ Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

27.9.8. AIC Interrupt Pending Register 1

Name: AIC_IPR1
Offset: 0x24
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.
PID32...PID63 refer to the identifiers as defined in the Peripheral Identifiers section.

Bit	31	30	29	28	27	26	25	24
	PID63	PID62	PID61	PID60	PID59	PID58	PID57	PID56
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
	PID55	PID54	PID53	PID52	PID51	PID50	PID49	PID48
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
	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	PID36	PID35	PID34	PID33	PID32
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 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

27.9.9. AIC Interrupt Pending Register 2

Name: AIC_IPR2
Offset: 0x28
Reset: 0x00000000
Property: Read-only

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	PID76	PID75	PID74	PID73	PID72
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
	PID71	PID70	PID69	PID68	PID67	PID66	PID65	PID64
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bit 10 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

27.9.10. AIC Interrupt Pending Register 3

Name: AIC_IPR3
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

The reset value of this register depends on the level of the external interrupt source. All other sources are cleared at reset, thus not pending.
PID96...PID127 bit fields refer to the identifiers as defined in the Peripheral Identifiers section.

Bit	31	30	29	28	27	26	25	24
	PID127	PID126	PID125	PID124	PID123	PID122	PID121	PID120
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
	PID119	PID118	PID117	PID116	PID115	PID114	PID113	PID112
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
	PID111	PID110	PID109	PID108	PID107	PID106	PID105	PID104
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

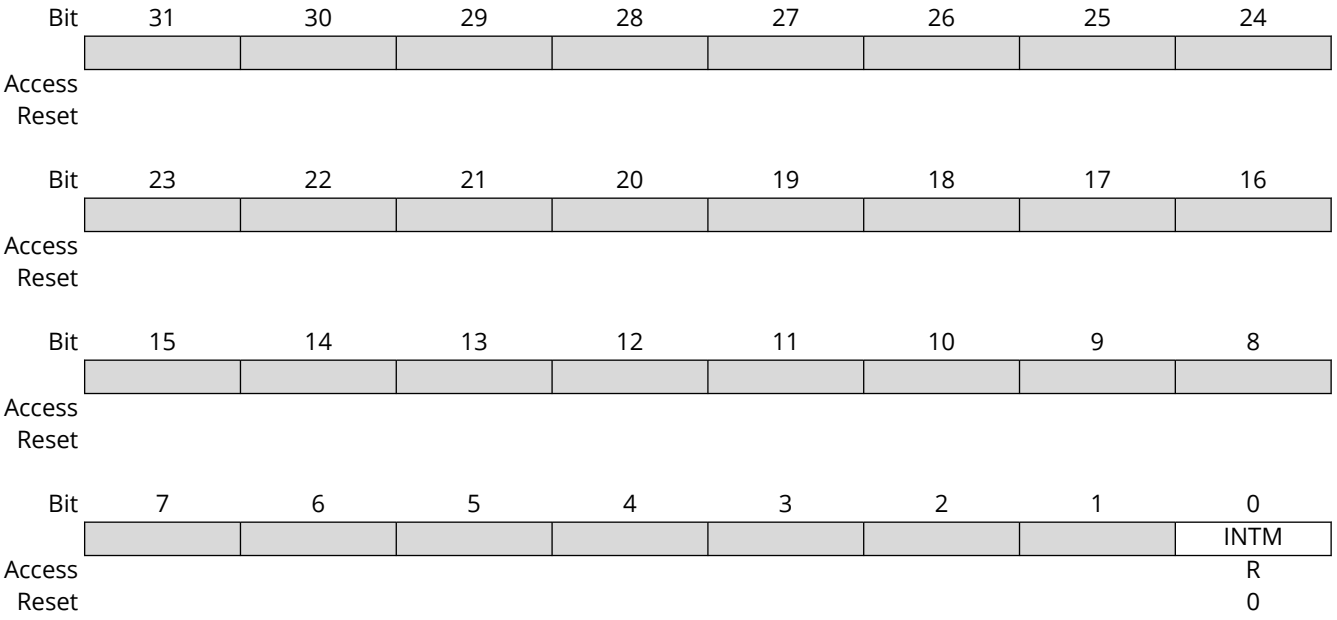
Bit	7	6	5	4	3	2	1	0
	PID103	PID102	PID101	PID100	PID99	PID98	PID97	PID96
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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 – PIDx Interrupt Pending

Value	Description
0	The corresponding interrupt is not pending.
1	The corresponding interrupt is pending.

27.9.11. AIC Interrupt Mask Register

Name: AIC_IMR
Offset: 0x30
Reset: 0x00000000
Property: Read-only



Bit 0 – INTM Interrupt Mask

Value	Description
0	The interrupt source selected by AIC_SSR.INTSEL is disabled.
1	The interrupt source selected by AIC_SSR.INTSEL is enabled.

27.9.12. AIC Core Interrupt Status Register

Name: AIC_CISR
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
Access								
Reset								

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

Bit 1 – NIRQ NIRQ Status

Value	Description
0	nIRQ line is deactivated.
1	nIRQ line is active.

Bit 0 – NFIQ NFIQ Status

Value	Description
0	nFIQ line is deactivated.
1	nFIQ line is active.

27.9.13. AIC End of Interrupt Command Register

Name: AIC_EOICR
Offset: 0x38
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
								ENDIT
Access								W
Reset								–

Bit 0 – ENDIT Interrupt Processing Complete Command
Used by the interrupt routine to indicate that the interrupt treatment is complete. Any value can be written because it is only necessary to make a write to this register location to signal the end of interrupt treatment.

27.9.14. AIC Spurious Interrupt Vector Register

Name: AIC_SPU
Offset: 0x3C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	SIVR[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
	SIVR[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
	SIVR[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
	SIVR[7:0]							
Access	R/W	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 – SIVR[31:0] Spurious Interrupt Vector Register
The user may store the address of a spurious interrupt handler in this register. The written value is returned in AIC_IVR in case of a spurious interrupt, or in AIC_FVR in case of a spurious fast interrupt.

27.9.15. AIC Interrupt Enable Command Register

Name: AIC_IECR
Offset: 0x40
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
								INTEN
Access								W
Reset								–

Bit 0 – INTEN Interrupt Enable

Value	Description
0	No effect.
1	Enables the interrupt source selected by AIC_SSR.INTSEL.

27.9.16. AIC Interrupt Disable Command Register

Name: AIC_IDCR
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								INTD
Access								W
Reset								–

Bit 0 – INTD Interrupt Disable

Value	Description
0	No effect.
1	Disables the interrupt source selected by AIC_SSR.INTSEL.

27.9.17. AIC Interrupt Clear Command Register

Name: AIC_ICCR
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
Access								
Reset								

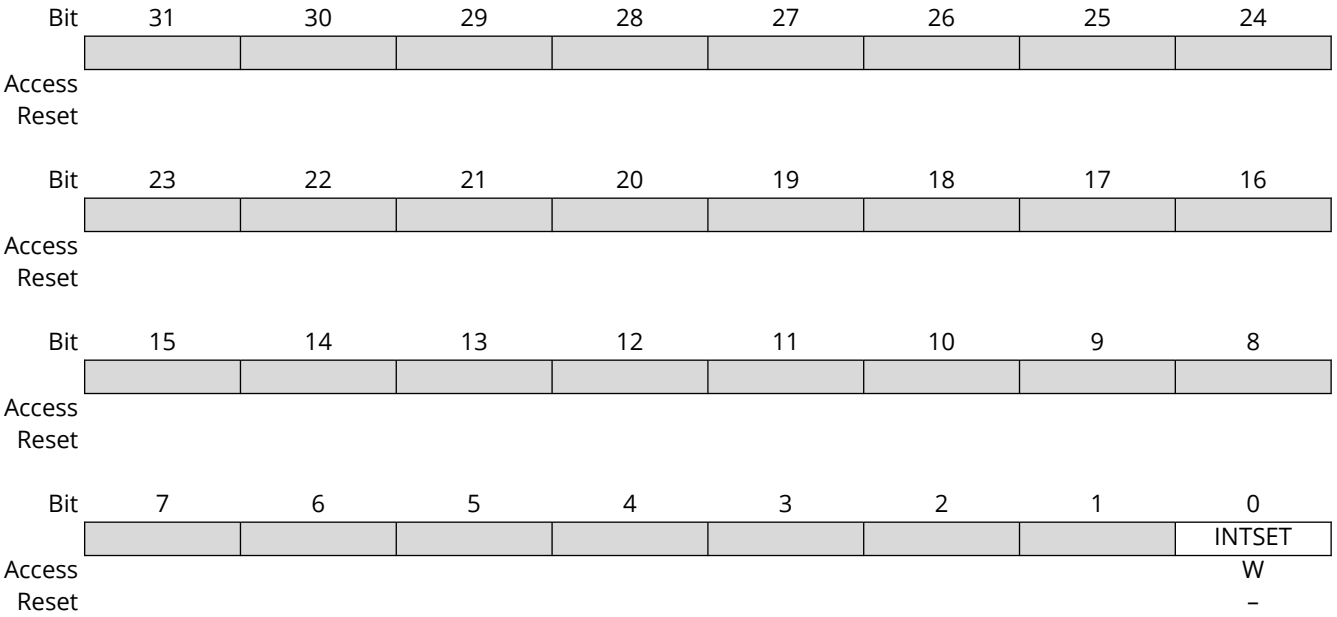
Bit	7	6	5	4	3	2	1	0
								INTCLR
Access								W
Reset								–

Bit 0 – INTCLR Interrupt Clear
Clears one the following depending on the setting of AIC_SSR.INTSEL: FIQ, SYS, PID2-PID60

Value	Description
0	No effect.
1	Clears the interrupt source selected by AIC_SSR.INTSEL.

27.9.18. AIC Interrupt Set Command Register

Name: AIC_ISCR
Offset: 0x4C
Reset: –
Property: Write-only



Bit 0 – INTSET Interrupt Set

Value	Description
0	No effect.
1	Sets the interrupt source selected by INTSEL.

27.9.19. AIC Fast Forcing Enable Register

Name: AIC_FFER
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FFEN
Access								W
Reset								–

Bit 0 – FFEN Fast Forcing Enable

Value	Description
0	No effect.
1	Enables the fast forcing feature on the interrupt source selected by INTSEL.

27.9.20. AIC Fast Forcing Disable Register

Name: AIC_FFDR
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
								FFDIS
Access								W
Reset								–

Bit 0 – FFDIS Fast Forcing Disable

Value	Description
0	No effect.
1	Disables the Fast Forcing feature on the interrupt source selected by INTSEL.

27.9.21. AIC Fast Forcing Status Register

Name: AIC_FFSR
Offset: 0x58
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
								FFS
Access								R
Reset								0

Bit 0 – FFS Fast Forcing Status

Value	Description
0	The Fast Forcing feature is disabled on the interrupt source selected by INTSEL.
1	The Fast Forcing feature is enabled on the interrupt source selected by INTSEL.

27.9.22. AIC SVR Return Enable Register

Name: AIC_SVRRER
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								

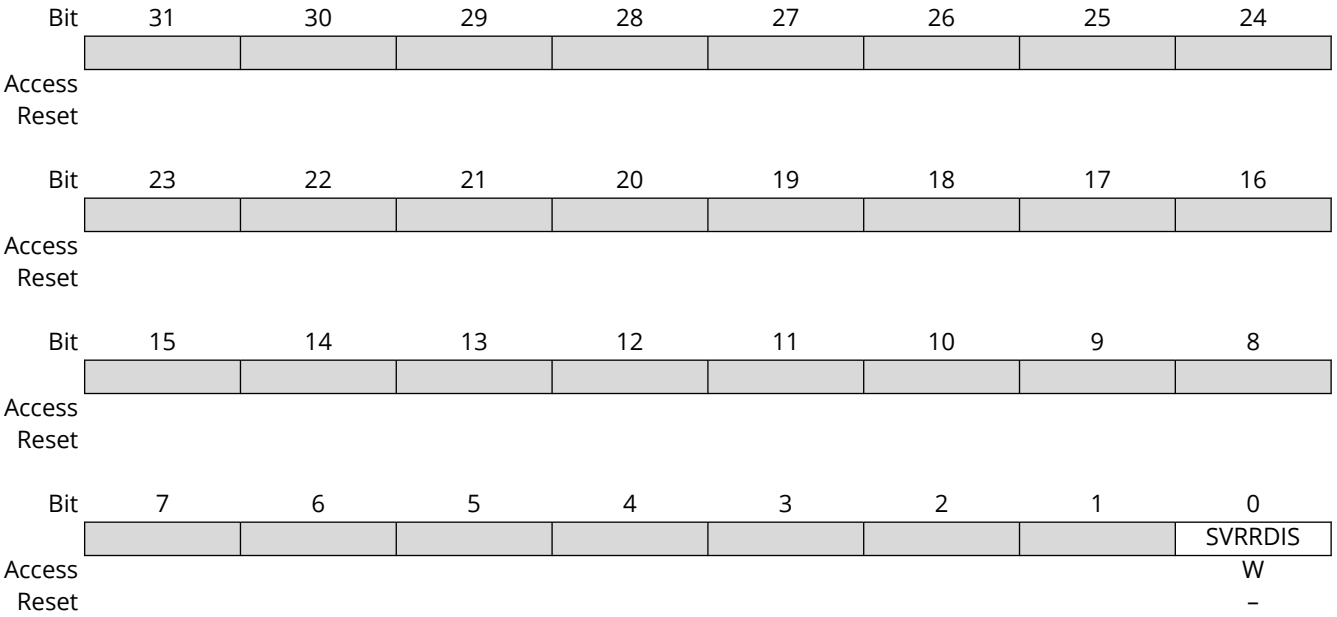
Bit	7	6	5	4	3	2	1	0
								SVRREN
Access								W
Reset								–

Bit 0 – SVRREN SVR Return Enable

Value	Description
0	No effect.
1	IVR register returns the interrupt index for the interrupt source selected by INTSEL.

27.9.23. AIC SVR Return Disable Register

Name: AIC_SVRRDR
Offset: 0x64
Reset: –
Property: Write-only



Bit 0 – SVRRDIS SVR Return Disable

Value	Description
0	No effect.
1	IVR register returns the corresponding vector programmed in AIC_SVR for the interrupt source selected by INTSEL.

27.9.24. AIC SVR Return Status Register

Name: AIC_SVRRSR
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
								SVRRS
Access								R
Reset								0

Bit 0 – SVRRS SVR Return Status

Value	Description
0	IVR register returns the corresponding vector programmed in AIC_SVR for the interrupt source selected by INTSEL.
1	IVR register returns the interrupt index for the interrupt source selected by INTSEL.

27.9.25. AIC Debug Control Register

Name: AIC_DCR
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the AIC 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
							GMSK	PROT
Access							R/W	R/W
Reset							0	0

Bit 1 – GMSK General Interrupt Mask

Value	Description
0	The nIRQ and nFIQ lines are normally controlled by the AIC.
1	The nIRQ and nFIQ lines are tied to their inactive state.

Bit 0 – PROT Protection Mode

Value	Description
0	The Protection mode is disabled.
1	The Protection mode is enabled.

27.9.26. AIC Write Protection Mode Register

Name: AIC_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
0x414943	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 section Register Write Protection for the list of registers that can be protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x414943 (“AIC” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x414943 (“AIC” in ASCII).

27.9.27. AIC Write Protection Status Register

Name: AIC_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 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 AIC_WPSR.
1	A write protection violation has occurred since the last read of AIC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

28. Boot Sequence Controller (BSC)

28.1. Description

The System Controller embeds a Boot Sequence Controller (BSC). The boot sequence is programmable through the Boot Sequence Controller Configuration register (BSC_CR).

The BSC_CR is powered by VDDBU. Any modification of the register value is stored and applied after the next reset. The register defaults to the factory value in case of battery removal.

The BSC_CR is programmable with user programs or SAM-BA and is key-protected.

28.2. Embedded Characteristics

- Boot Sequence Settings
- VDDBU-powered Register

28.3. Product Dependencies

Refer to the “Boot Strategies” section.

28.4. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	BSC_CR	31:24	WPKEY[15:8]							
		23:16	WPKEY[7:0]							
		15:8								
		7:0						BOOT[2:0]		

28.4.1. Boot Sequence Controller Configuration Register

Name: BSC_CR
Offset: 0x0
Reset: –
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	WPKEY[7:0]							
Access	W	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
						BOOT[2:0]		
Access						R/W	R/W	R/W
Reset						–	–	–

Bits 31:16 – WPKEY[15:0] Write Protection Key (Write-only)

Value	Name	Description
0x6683	PASSWD	Writing any other value in this field aborts the write operation of the BOOT field. Always reads as 0.

Bits 2:0 – BOOT[2:0] Boot Media Sequence
This value is defined in the “Boot Strategies” section. It is only written if WPKEY carries the valid value.

29. General Purpose Backup Registers (GPBR)

29.1. Description

The System Controller embeds 256 bits of General Purpose Backup registers organized as 8 32-bit registers.

It is possible to generate an immediate clear of the content of General Purpose Backup registers 0 to 3 if a tamper event is detected on WKUP1 to WKUP8 pins. These pins are internally routed through the VDDCORE area, thus tamper events can be generated only when the VDDCORE is powered. These pins are also used for fast wake-up in Power Management Controller (PMC). Thus, if some WKUP pins are not enabled for fast wake-up in PMC, they can be enabled for tamper event detection. The immediate clear of the GPBR is enabled if RSTC_MR.ENGCLR=1.

The immediate clear on tamper detection can be extended to all General Purpose Backup registers by writing to '1' GPBR_FCLR.FCLR.

If an event has been detected on WKUP pins enabled for event detection in RTC, it is not possible to write to the General Purpose Backup registers (SYS_GPBRx) while the event has not been cleared.

SYS_GPBR0 to SYS_GPBR7 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 VDDBU.

29.2. Embedded Characteristics

- 256 bits of General Purpose Backup Registers
- Immediate Clear on WKUP Event
- Read and Write Protection for SYS_GPBR0 to SYS_GPBR7

29.3. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	GPBR_MR	31:24								
		23:16	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
		15:8								
		7:0	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	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]							
0x10	SYS_GPBR2	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x14	SYS_GPBR3	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x18	SYS_GPBR4	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x1C	SYS_GPBR5	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x20	SYS_GPBR6	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							
0x24	SYS_GPBR7	31:24	GPBR_VALUE[31:24]							
		23:16	GPBR_VALUE[23:16]							
		15:8	GPBR_VALUE[15:8]							
		7:0	GPBR_VALUE[7:0]							

29.3.1. 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 VDDBU.

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
	GPBRRP7	GPBRRP6	GPBRRP5	GPBRRP4	GPBRRP3	GPBRRP2	GPBRRP1	GPBRRP0
Access	R/W	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
	GPBRWP7	GPBRWP6	GPBRWP5	GPBRWP4	GPBRWP3	GPBRWP2	GPBRWP1	GPBRWP0
Access	R/W	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, 17, 18, 19, 20, 21, 22, 23 – 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, 2, 3, 4, 5, 6, 7 – 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.

29.3.2. 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).

Bit	31	30	29	28	27	26	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
GPBR full clear is only possible if the system is not in Backup mode. In Backup mode, FCLR has no effect.

Value	Description
0	SYS_GPBR0 to SYS_GPBR3 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.

29.3.3. General Purpose Backup Register x [x=0..7]

Name: SYS_GPBRx
Offset: 0x08 + x*0x04 [x=0..7]
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 VDDBU.

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.

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

30. Watchdog Timer (WDT)

30.1. Description

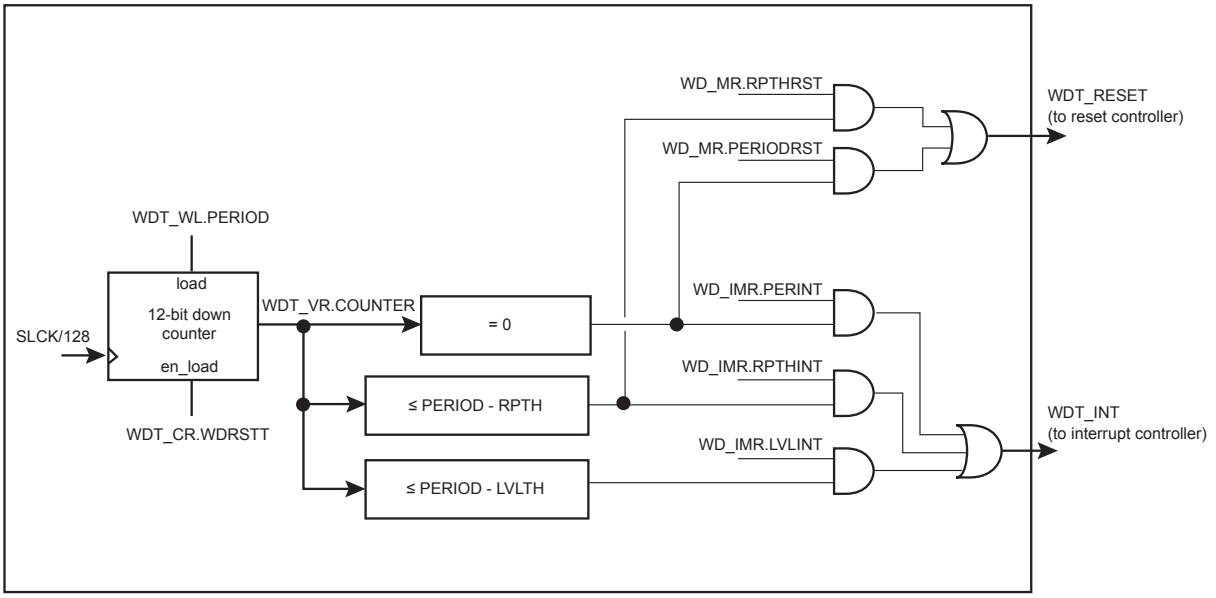
The Watchdog Timer (WDT) is used to prevent system lockup if the software becomes trapped in a deadlock. It features one 12-bit down counter that allows a watchdog period of up to 16 seconds (slow clock around 32 kHz). The WDT can generate a general reset or a processor reset only. In addition, it can be stopped while the processor is in Debug mode or Sleep mode (Idle mode).

30.2. 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

30.3. Block Diagram

Figure 30.1. WDT Block Diagram



30.4. Functional Description

The WDT is used to prevent system lockup if the software becomes trapped in a deadlock. It is supplied with VDDCORE. It restarts with initial values on processor reset.

The WDT is built around a 12-bit down counter loaded with the value defined in field PERIOD of the Window Level Register (WDT_WLR). WDT uses slow clock divided by 128 to establish the maximum watchdog period to 16 seconds (with a typical slow clock of 32.768 kHz).

The following parameters can be configured:

- Watchdog event period: a watchdog event occurs when the 12-bit down counter reaches 0, and leads to either an interrupt (if bit PERINT in the Interrupt Mask register (WDT_IMR) is high) or a reset (if bit PERIODRST in the Mode register (WDT_MR) is high).

- Minimum restart period: if the restart command is performed before this period, WDT creates a repeat violation. A repeat violation leads to either an interrupt (if WDT_IMR.RPTHINT = 1) or a reset (if WDT_MR.RPTHRST = 1).
- Maximum period before single interrupt event: if WDT_IMR.LVLINT = 1, a single interrupt is generated (no reset). The WDT_ILR.LVLTH value must be lower than WDT_WLR.PERIOD.

After a processor reset, the value of PERIOD is 0xFFF, corresponding to the maximum value of the counter with the external reset generation enabled (bit PERIODRST at 1 after a backup reset). This means that the WDT is running at reset, i.e., at powerup. The user can either disable the WDT by setting bit WDT_MR.WDDIS to '1' or reprogram the WDT to meet the maximum WDT period the application requires.

If the WDT is restarted by writing into the corresponding Control register (WDT_CR), the corresponding WDT_MR must not be programmed during a period of time of three slow clock periods following the WDT_CR write access. In any case, programming a new value in WDT_MR automatically initiates a restart instruction.

WDT_MR, WDT_WLR and WDT_ILR (Interrupt Level register) can be written until a WDT_CR.LOCKMR command is issued in the corresponding WDT_CR. Only a peripheral reset can configure the bit LOCKMR to 0.

When the bit WDT_CR.LOCKMR = 0, writing WDT_WLR reloads the corresponding WDT with the newly programmed mode parameters.

In normal operation, the user reloads the WDT at regular intervals before the timer underflow occurs, by setting bit WDT_CR.WDRSTT. The WDT counter is then immediately reloaded from PERIOD and restarted, and the slow clock 128 divider is reset and restarted.

Writing WDR_CR without the correct hard-coded key has no effect (see Watchdog Timer Control Register).

A repeat threshold can be defined for each watchdog in order to protect against dead-locks that would repeatedly restart the watchdog. WDT_WLR.RPTH defines the minimum number of cycles to wait after a watchdog restart before the WDT can be started again. If a watchdog restart occurs before this limit is reached, a repeat threshold failure is asserted and the RPTHINT bit in the Interrupt Status register (WDT_ISR) is set to one.

If WDT_IMR.RPTHINT is high and a repeat threshold violation occurs in the WDT, an interrupt is generated.

If WDT_MR.RPTHRST is high and a repeat threshold violation occurs in the WDT, a watchdog reset is generated.

WDT reload must occur while the WDT counter is within a window between 0 and (PERIOD–RPTH). PERIOD and RPTH are defined in WDT_WLR.

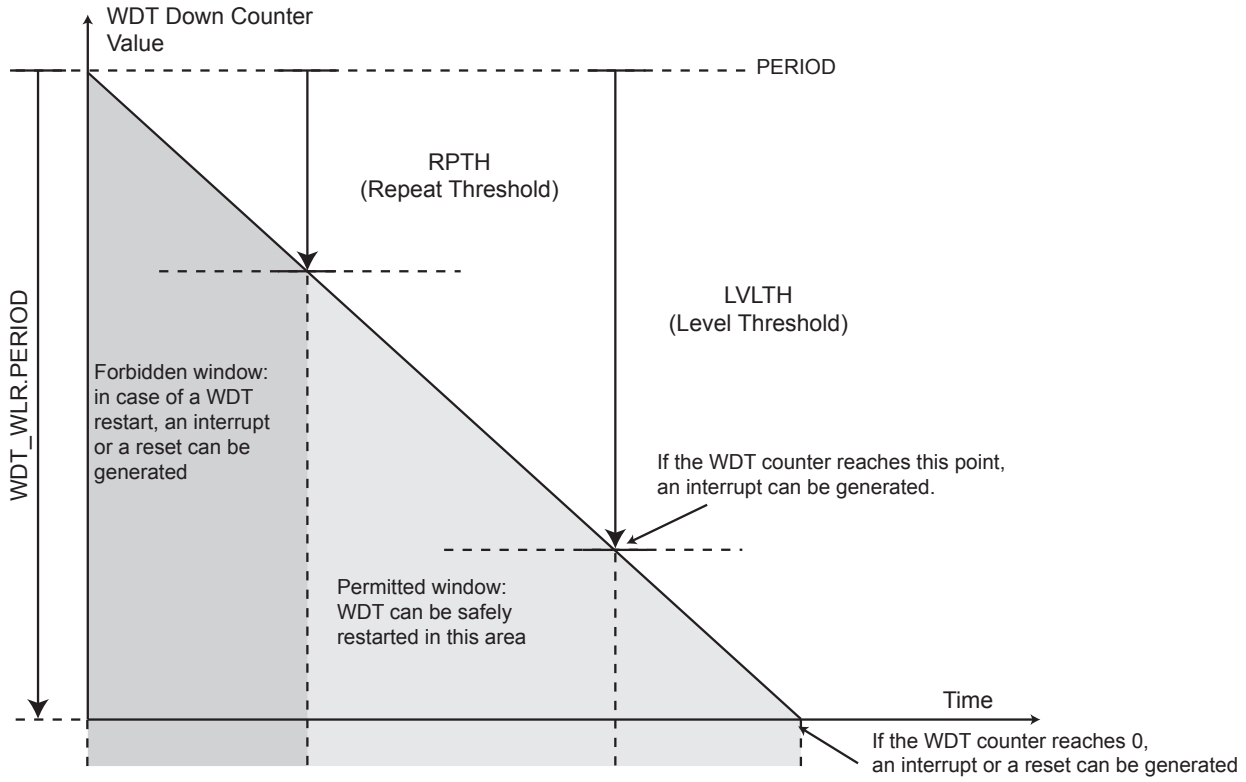
Note that this feature can be disabled by programming a null RPTH value. In such configuration, restarting the WDT is permitted in the whole range [0 up to PERIOD] and does not generate an error. This is the default configuration on reset (RPTH is null).

If a reset is generated or if WDT_SR is read, the status bits are reset and the interrupt is cleared.

Writing WDT_MR reloads and restarts the down counter.

While the processor is in debug state or in Sleep mode, the counter may be stopped depending on the value programmed for the bits WDIDLEHLT and WDDBGHLT in WDT_MR.

Figure 30.2. Watchdog Timing Diagram



30.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	WDT_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0				LOCKMR				WDRSTT
0x04	WDT_MR	31:24			WDDBGHLT	WDIDLEHLT				
		23:16								
		15:8				WDDIS				
		7:0			RPTHRST	PERIODRST				
0x08	WDT_VR	31:24								
		23:16								
		15:8					COUNTER[11:8]			
		7:0	COUNTER[7:0]							
0x0C	WDT_WLR	31:24					RPTH[11:8]			
		23:16	RPTH[7:0]							
		15:8					PERIOD[11:8]			
		7:0	PERIOD[7:0]							
0x10	WDT_ILR	31:24								
		23:16								
		15:8					LVLTH[11:8]			
		7:0	LVLTH[7:0]							
0x14	WDT_IER	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x18	WDT_IDR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x1C	WDT_ISR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT
0x20	WDT_IMR	31:24								
		23:16								
		15:8								
		7:0						LVLINT	RPTHINT	PERINT

30.5.1. Watchdog Timer Control Register

Name: WDT_CR
Offset: 0x00
Reset: –
Property: Write-only

The WDT_CR register values must not be modified within three slow clock periods following a restart of the WDT performed by a write access in WDT_CR. Any modification will cause the WDT 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 WDT_MR, WDT_WLR and WDT_ILR have no effect.

Bit 0 – WDRSTT Watchdog Restart

Value	Description
0	No effect.
1	Restarts the WDT if KEY is written to 0xA5.

30.5.2. Watchdog Timer Mode Register

Name: WDT_MR
Offset: 0x04
Reset: 0x00000030
Property: Read/Write

Write access to this register has no effect if the LOCKMR command is issued in WDT_CR (unlocked on hardware reset).

The WDT_MR register values must not be modified within three slow clock periods following a restart of the WDT performed by a write access in WDT_CR. Any modification will cause the WDT to trigger an end of period earlier than expected.

Bit	31	30	29	28	27	26	25	24
			WDDBGHLT	WDIDLEHLT				
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 – WDDBGHLT Watchdog Debug Halt

Value	Description
0	The WDT runs when the processor is in Debug state.
1	The WDT stops when the processor is in Debug state.

Bit 28 – WDIDLEHLT Watchdog Idle Halt

Value	Description
0	The WDT runs when the system is in Idle state.
1	The WDT stops when the system is in Idle state.

Bit 12 – WDDIS Watchdog Disable

Value	Description
0	Enables the WDT.
1	Disables the WDT.

Bit 5 – RPTHRST Minimum Restart Period

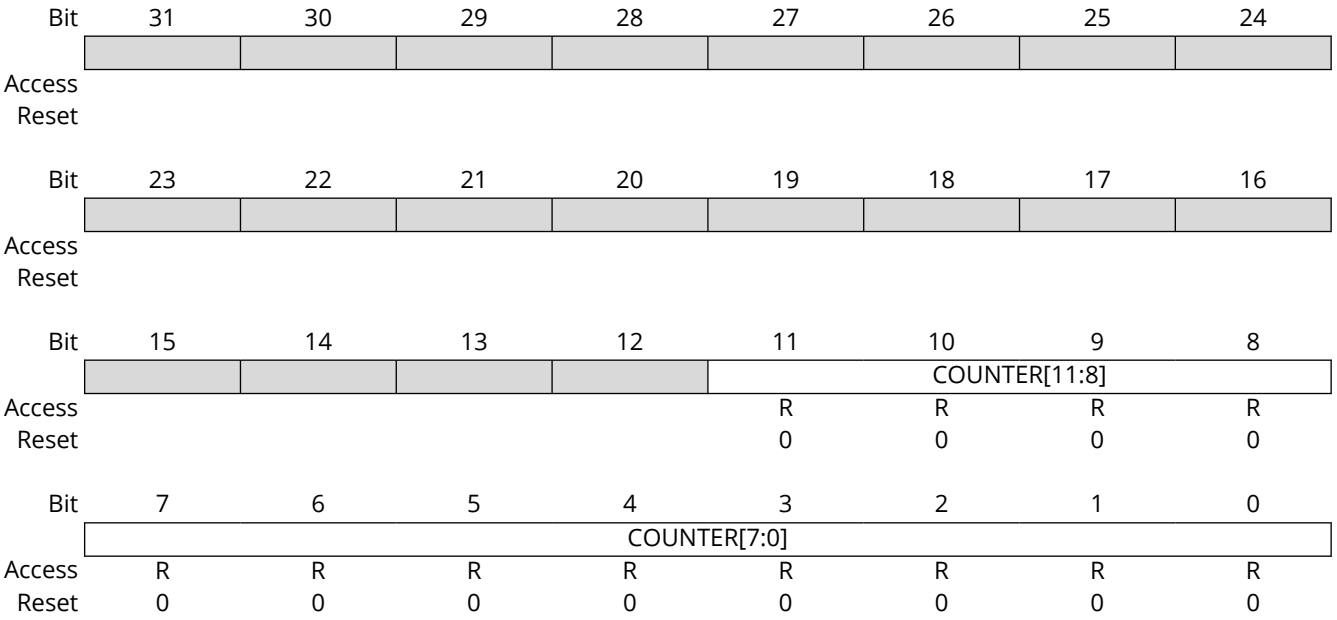
Value	Description
0	No reset is generated if the WDT is restarted before the RPTH threshold.
1	A reset is generated if the WDT is restarted before the RPTH threshold.

Bit 4 – PERIODRST Period Reset

Value	Description
0	No reset is generated if the WDT down counter reaches 0.
1	A reset is generated once the WDT down counter reaches 0.

30.5.3. Watchdog Timer Value Register

Name: WDT_VR
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
					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

Shows the current value of the WDT down counter for debug operation.
Due to the asynchronous operation of the WDT with respect to the rest of the chip, to be certain that the value read in this register is valid and stable, it is necessary to read the 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.

30.5.4. Watchdog Timer Window Level Register

Name: WDT_WLR
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 WDT restart generates an interrupt.

Bits 11:0 – PERIOD[11:0] Watchdog Period
Defines the period after which the WDT generates a reset.

30.5.5. Watchdog Timer Interrupt Level Register

Name: WDT_ILR
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 WDT generates an interrupt.

30.5.6. Watchdog Interrupt Enable Register

Name: WDT_IER
Offset: 0x14
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
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Enable

Bit 1 – RPTHINT Repeat Threshold Interrupt Enable

Bit 0 – PERINT Period Interrupt Enable

30.5.7. Watchdog Interrupt Disable Register

Name: WDT_IDR
Offset: 0x18
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
						LVLINT	RPTHINT	PERINT
Access						W	W	W
Reset						–	–	–

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Disable

Bit 1 – RPTHINT Repeat Threshold Interrupt Disable

Bit 0 – PERINT Period Interrupt Disable

30.5.8. Watchdog Interrupt Status Register

Name: 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 Status (cleared on read)

Value	Description
0	No level threshold failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one level threshold failure has occurred in the WDT since the last read of WDT_ISR.

Bit 1 – RPTHINT Repeat Threshold Interrupt Status (cleared on read)

Value	Description
0	No repeat threshold failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one repeat threshold failure has occurred in the WDT since the last read of WDT_ISR.

Bit 0 – PERINT Period Interrupt Status (cleared on read)

Value	Description
0	No period failure has occurred in the WDT since the last read of WDT_ISR.
1	At least one period failure has occurred in the WDT since the last read of WDT_ISR.

30.5.9. Watchdog Interrupt Mask Register

Name: WDT_IMR
Offset: 0x20
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
						LVLINT	RPTHINT	PERINT
Access						R	R	R
Reset						0	0	0

Bit 2 – LVLINT Interrupt Level Threshold Interrupt Mask

Bit 1 – RPTHINT Repeat Threshold Interrupt Mask

Bit 0 – PERINT Period Interrupt Mask

31. Reset Controller (RSTC)

31.1. 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.

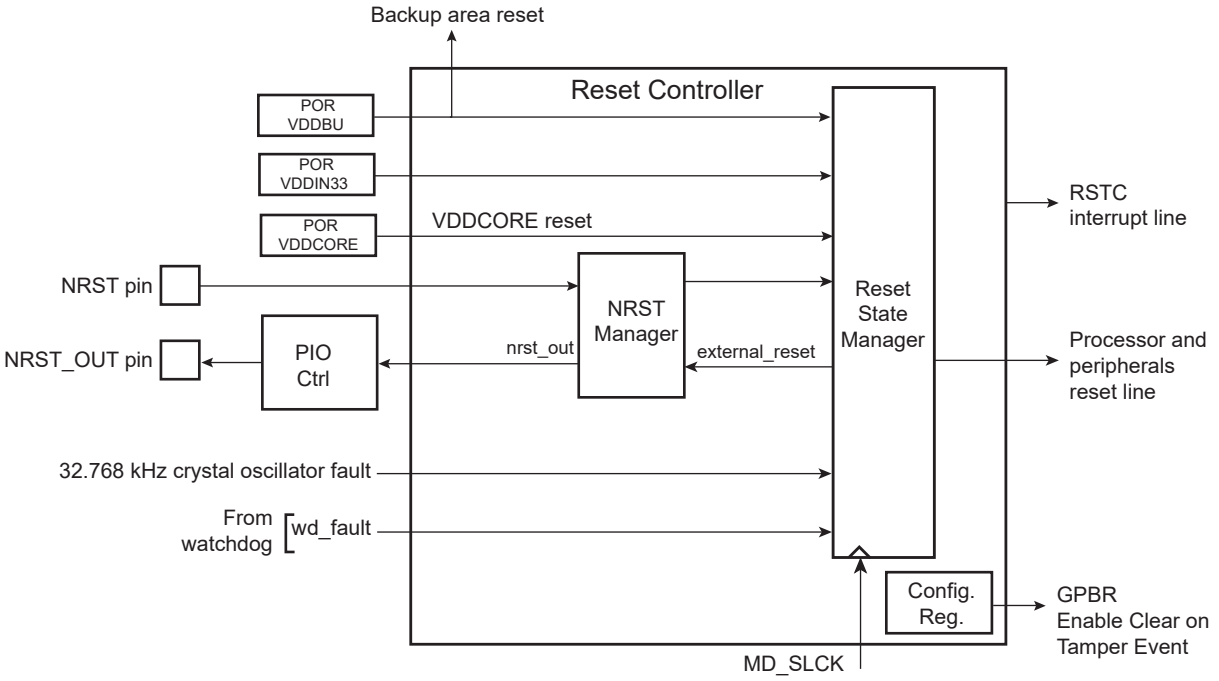
The RSTC drives simultaneously the External reset and the Peripheral and Processor resets.

31.2. 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, VDDIN33 and VDDBU POR reset, Software reset, User reset, Watchdog reset, 32.768 kHz Crystal Oscillator Failure Detection reset

31.3. Block Diagram

Figure 31.1. RSTC Block Diagram



31.4. 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.

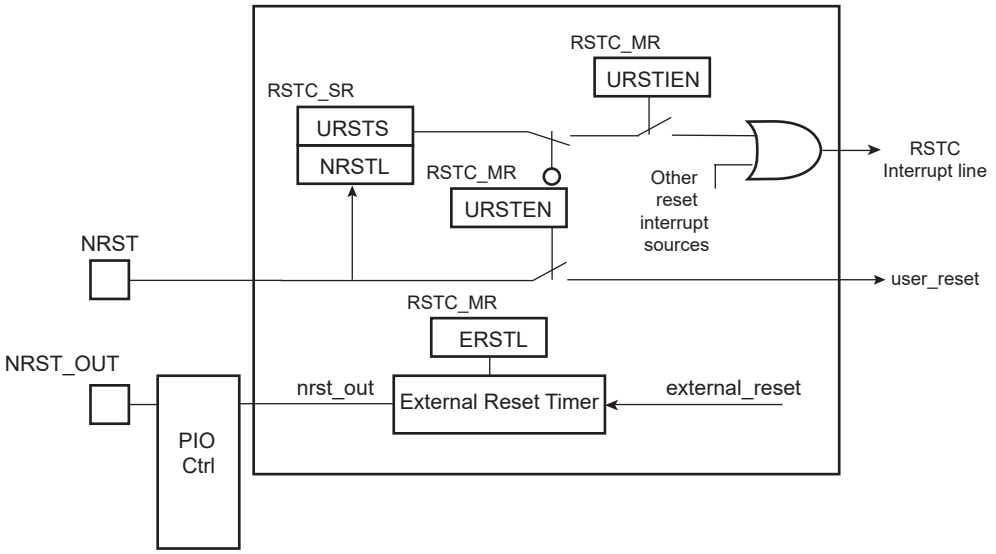
The NRST manager asserts the NRST_OUT pin during a programmable time, thus controlling external device resets.

The Mode register (RSTC_MR), used to configure the RSTC, is powered by VDDBU.

31.4.1. 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 31.2. NRST Pin Management



31.4.1.1. 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.

31.4.1.2. 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).

31.4.2. 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.

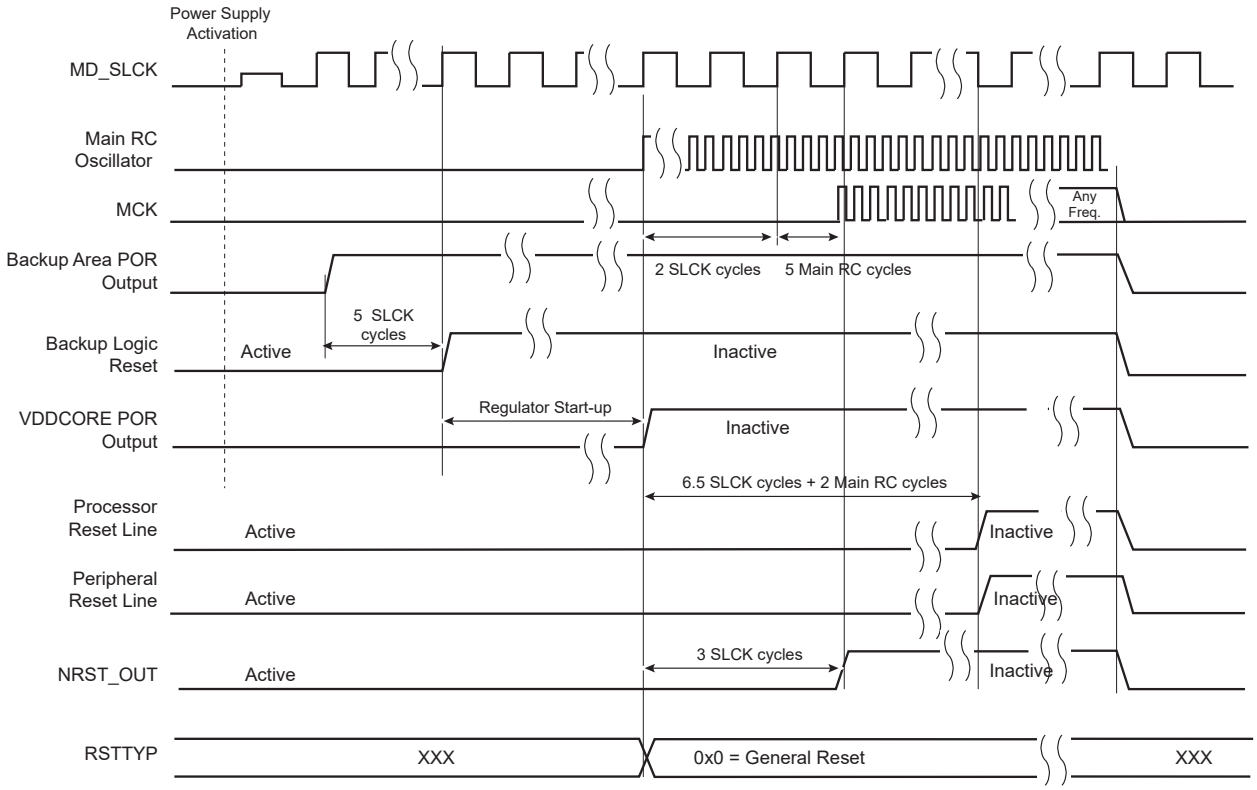
31.4.2.1. General Reset

A general reset occurs when a VDDBU 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 31.3. General Reset Timing Diagram



31.4.2.2.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.

31.4.2.3.32.768 kHz Crystal Oscillator Failure Detection Reset

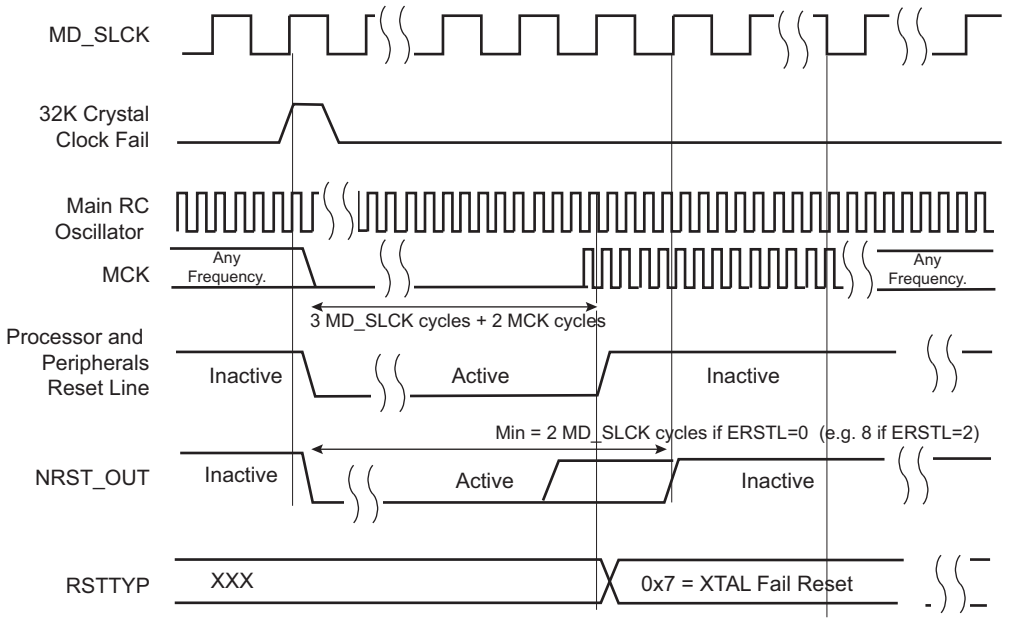
The 32.768 kHz Crystal Oscillator Failure Detection reset is done when the 32.768 kHz crystal oscillator frequency monitoring circuitry in the PMC detects a failure and RSTC_MR.SCKSW is written to '1'. This reset lasts three slow clock cycles.

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 31.4. 32.768 kHz Crystal Oscillator Failure Detection Reset Timing Diagram



31.4.2.4.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.

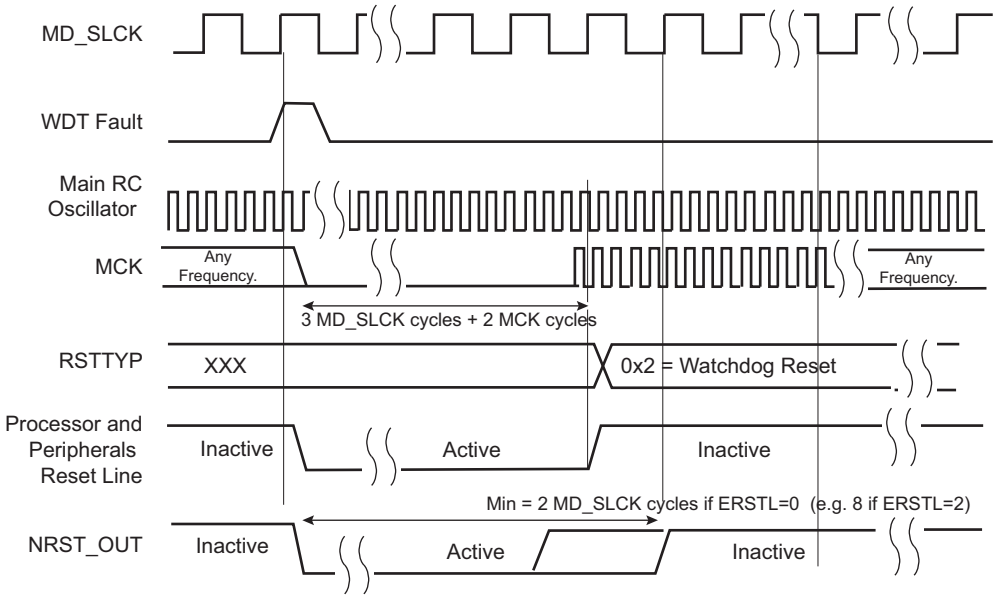
The WDT is reset by the Processor reset signal. As the watchdog fault always causes a Processor reset if WDT_MR.WDRSTEN is written to ‘1’, the WDT is always reset after a Watchdog reset, and the Watchdog is enabled by default and with a period set to a maximum.

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 31.5. Watchdog Reset Timing Diagram



31.4.2.5.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.

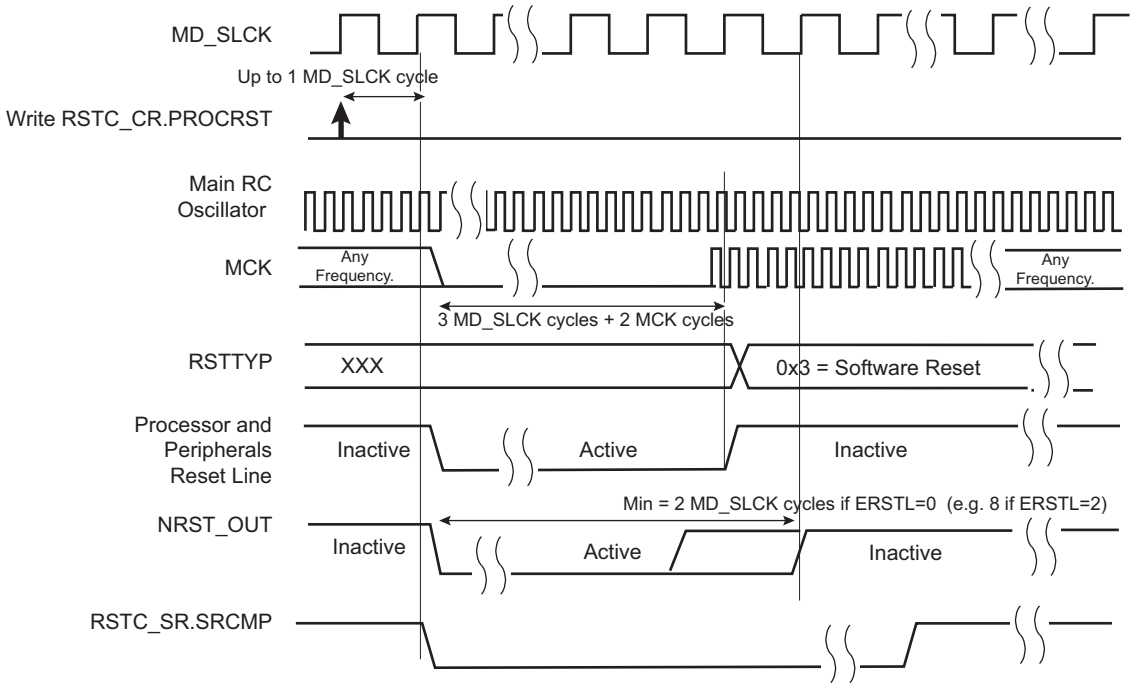
The internal reset signals are asserted as soon as the register write is performed. They are released when the Software reset has ended, i.e., synchronously to MD_SLCK.

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 31.6. Software Reset Timing Diagram (PROCRST)



31.4.2.6. 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 31.7. User Reset State (URSTASYNC = '0')

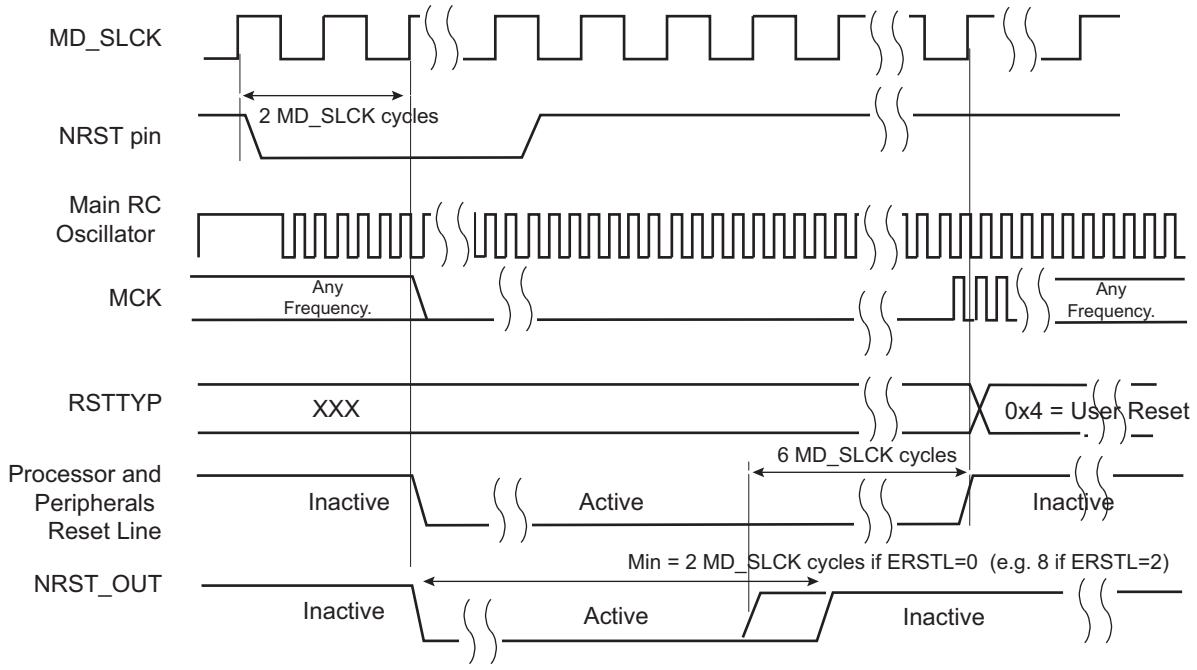
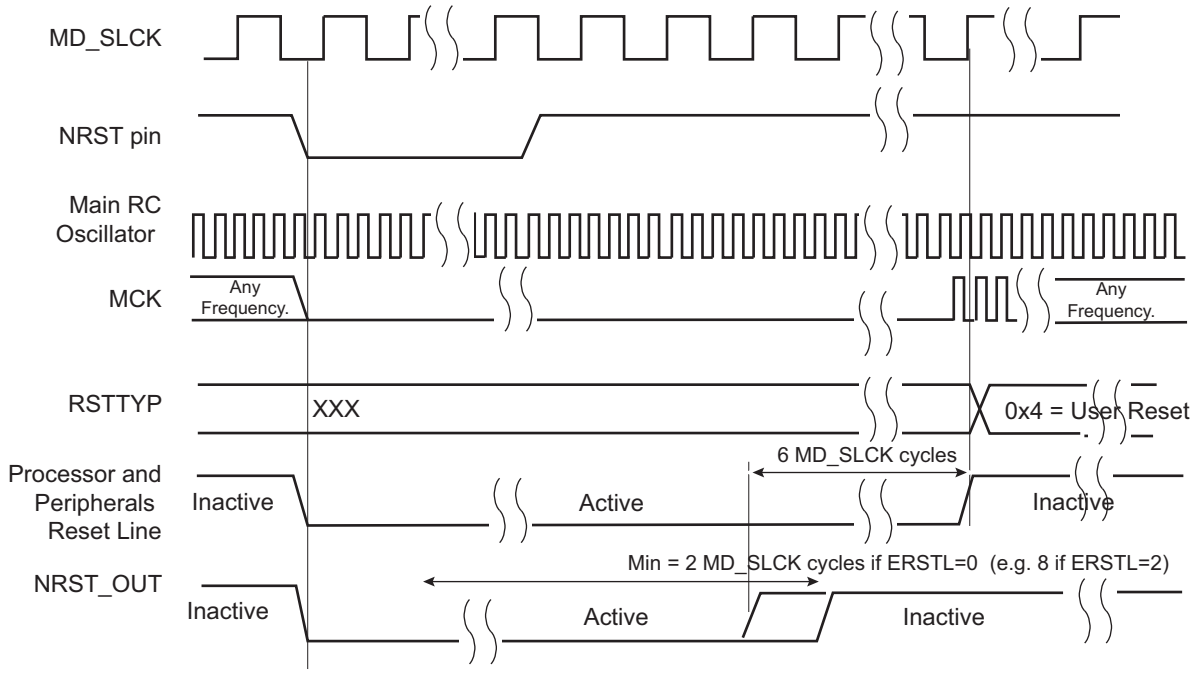


Figure 31.8. User Reset State (URSTASYNC = '1')



31.4.3. 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.

31.5. 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[2:0]		
		7:0								URSTS
0x08	RSTC_MR	31:24	KEY[7:0]							
		23:16				ENGCLR				
		15:8					ERSTL[3:0]			
		7:0				URSTIEN		URSTASYNC	SCKSW	URSTEN

31.5.1. 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.

31.5.2. RSTC Status Register

Name: RSTC_SR
Offset: 0x04
Reset: 0x00000001
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
Access							R	R
Reset							0	0

Bit	15	14	13	12	11	10	9	8
Access						R	R	R
Reset						0	0	0

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

Bit 17 – SRCMP Software Reset Command in Progress
When set, 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 10:8 – RSTTYP[2:0] Reset Type
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

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.

31.5.3. 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
				ENGCLR				
Access				R/W				
Reset				0				

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.

Bit 20 – ENGCLR Enable GPBR Clear on Tamper Event

Value	Description
0	Disables the GPBR immediate clear on tamper detection event.
1	Enables the GPBR immediate clear on tamper detection event

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.URSTS at ‘1’ has no effect on the RSTC interrupt line.
1	RSTC_SR.URSTS 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.

32. Real-Time Timer (RTT)

32.1. 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.

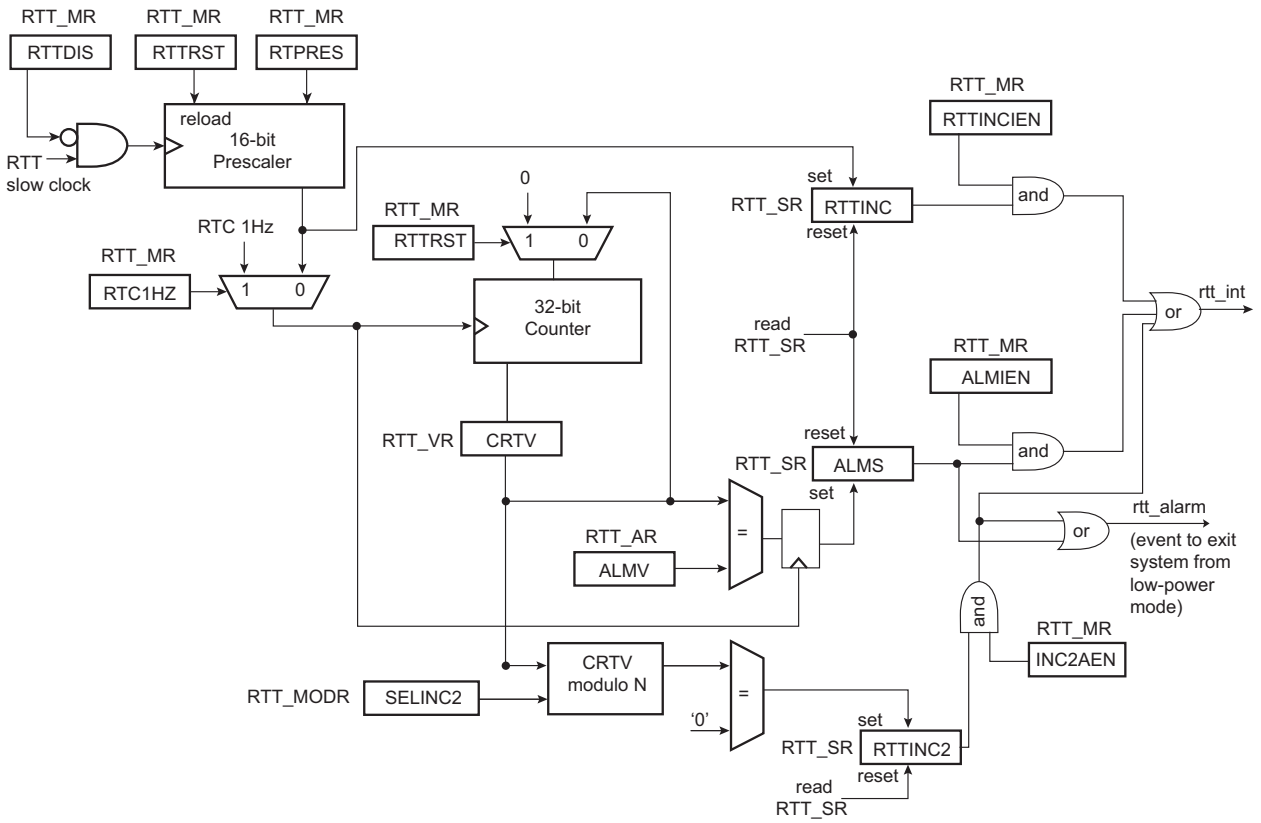
The slow clock source can be fully disabled to reduce power consumption when only an elapsed seconds count is required.

32.2. 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

32.3. Block Diagram

Figure 32.1. RTT Block Diagram



32.4. 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.

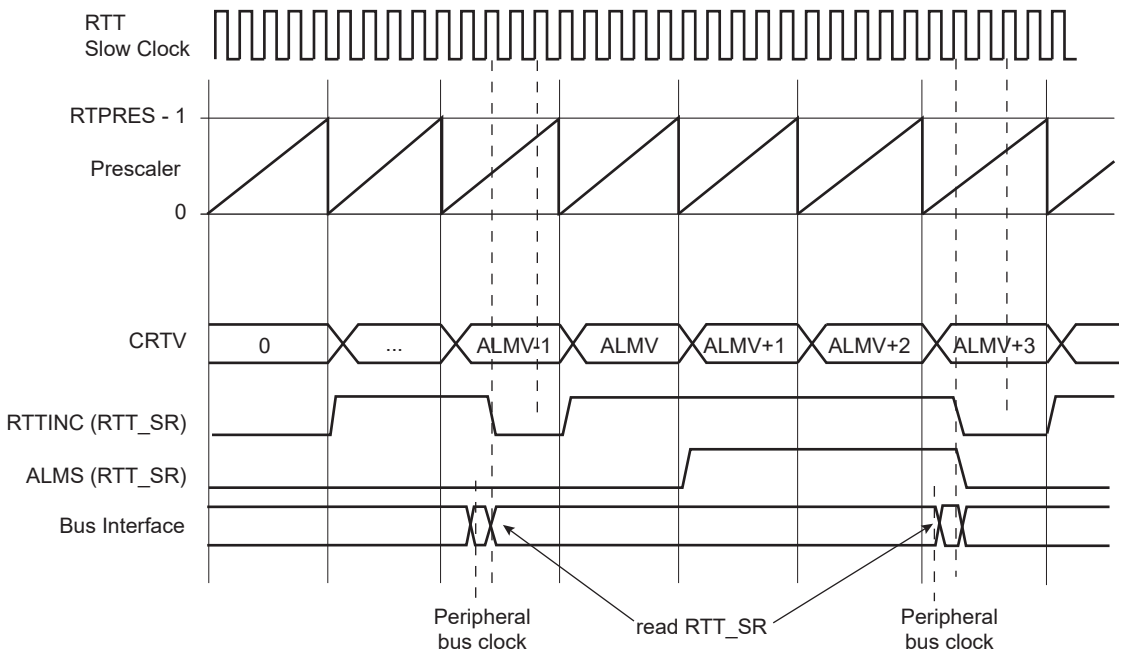
RTT_MR.RTTINCIEN must be cleared prior to writing a new value in RTT_MR. RTPRES.

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 32.2. RTT Counting



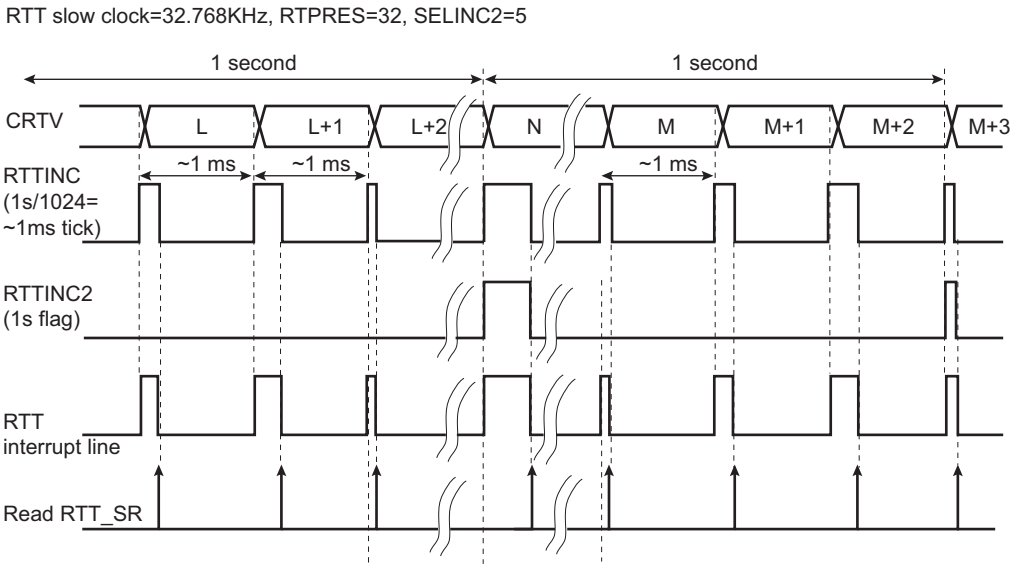
The RTTINC2 flag is set when the number of prescaler roll-overs programmed through the SELINC2 field in the RTT Modulo Selection register (RTT_MODR) has been reached since the last read of RTT_SR.

For example, it is possible to generate two 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 32.3. RTTINC2 Behavior



32.5. 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	TS_OVF	TSTAMP[30:24]						
		23:16	TSTAMP[23:16]							
		15:8	TSTAMP[15:8]							
		7:0	TSTAMP[7:0]							

32.5.1. 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 RTT_SR.RTTINC bit has no effect on interrupt.
1	The RTT_SR.RTTINC bit asserts interrupt.

Bit 16 – ALMIEN Alarm Interrupt Enable

Value	Description
0	The RTT_SR.ALMS bit has no effect on interrupt.
1	The RTT_SR.ALMS bit 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.

32.5.2. 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.

32.5.3. 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.

32.5.4. 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 RTT_SR.
1	The number of prescaler roll-overs programmed through the SELINC2 field has been reached since the last read of 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 RTT_SR.
1	Prescaler roll-over occurred since the last read of 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.

32.5.5. 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.

32.5.6. Real-Time Timer Timestamp Register

Name: RTT_TSR
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TS_OVF	TSTAMP[30:24]						
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
	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

Bit 31 – TS_OVF Timestamp Overflow

Value	Description
0	RTT_ST.RTTINC2 was set but no new event occurred since the last read of RTT_SR.
1	RTT_ST.RTTINC2 was set and a second event occurred. To avoid any overflow, clear RTTINC2 by reading RTT_SR.

Bits 30:0 – TSTAMP[30:0] 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 the next RTTINC2 event and can be used for event timestamping.

33. Real-Time Clock (RTC)

33.1. 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 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.

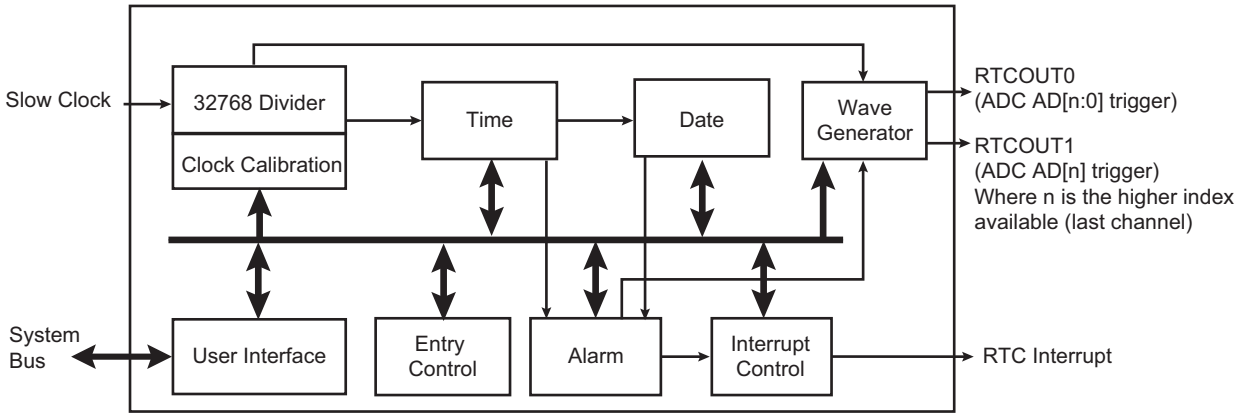
Timestamping capability reports the first and last occurrences of tamper events.

33.2. Embedded Characteristics

- Full Asynchronous Design for Ultra Low-Power Consumption
- Gregorian and UTC 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 Trigger Event
- Tamper Control Registers and Detection Logic
- Tamper Timestamping Registers
- Register Write Protection

33.3. Block Diagram

Figure 33.1. RTC Block Diagram



33.4. Product Dependencies

33.4.1. Power Management

The Real-Time Clock is continuously clocked at 32.768 kHz. The Power Management Controller (PMC) has no effect on RTC behavior.

33.4.2. 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.

33.5. 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.

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 events to trigger ADC measurements.

33.5.1. Reference Clock

The reference clock is the slow clock. It can be driven externally by a 32.768 kHz crystal, or internally.

33.5.2. Timing

In Gregorian mode, 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.

33.5.3. Alarm

In Gregorian mode, 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.

33.5.4. 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)
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.

33.5.5. 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.

33.5.6. Updating Time/Calendar

33.5.6.1. Calendar Mode

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 33.2. Time/Calendar Update Timing Diagram

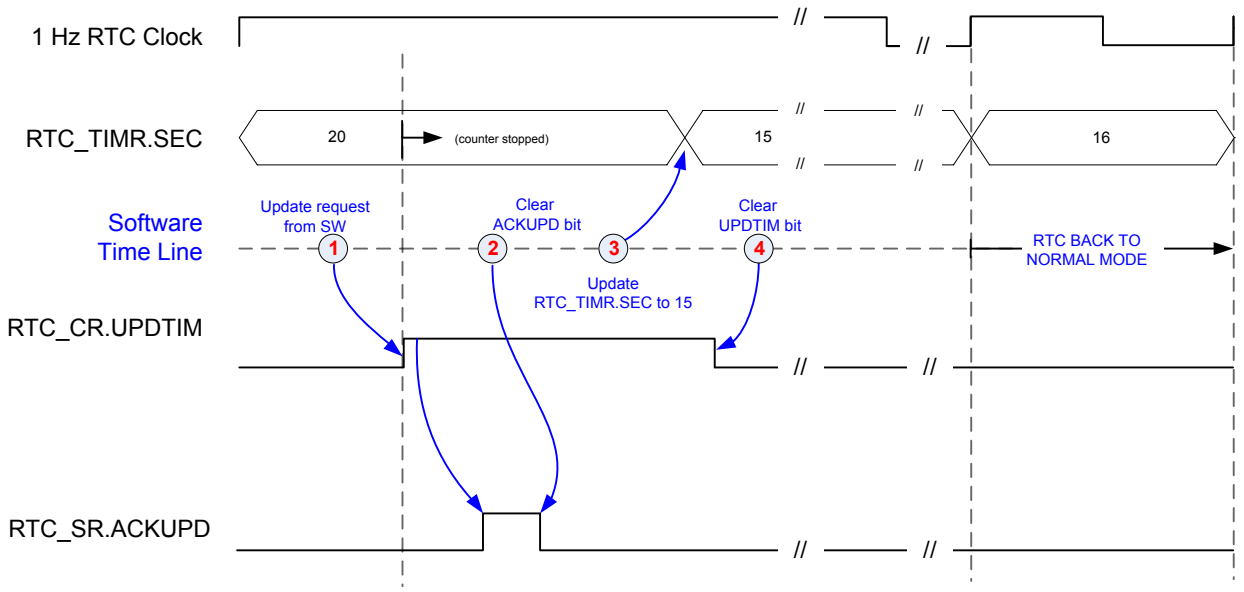
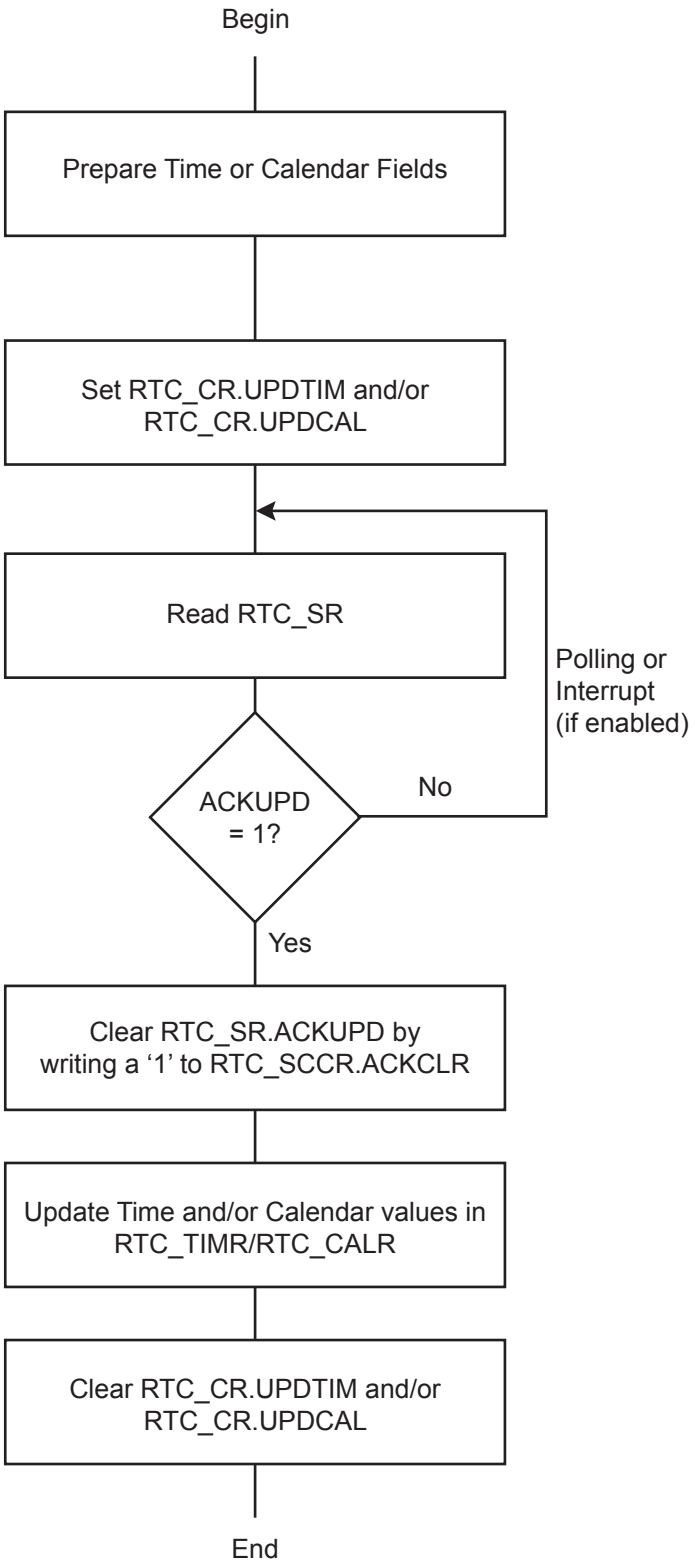


Figure 33.3. Calendar Mode Update Sequence



33.5.6.2.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 33.4. UTC Time Update Timing Diagram

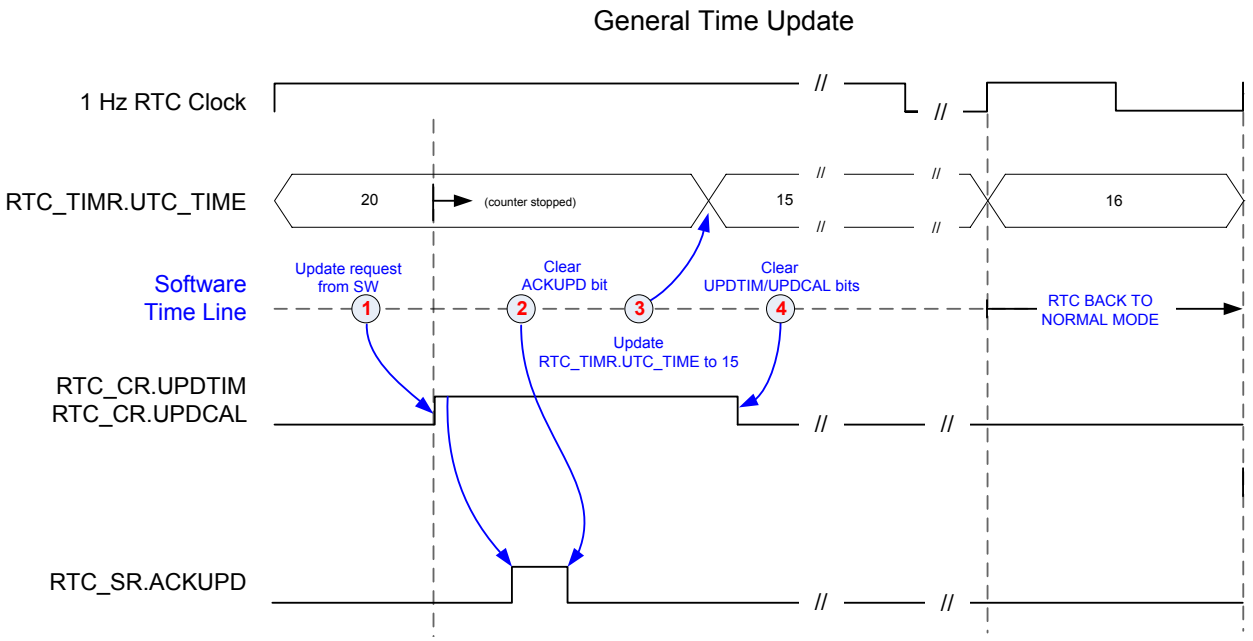
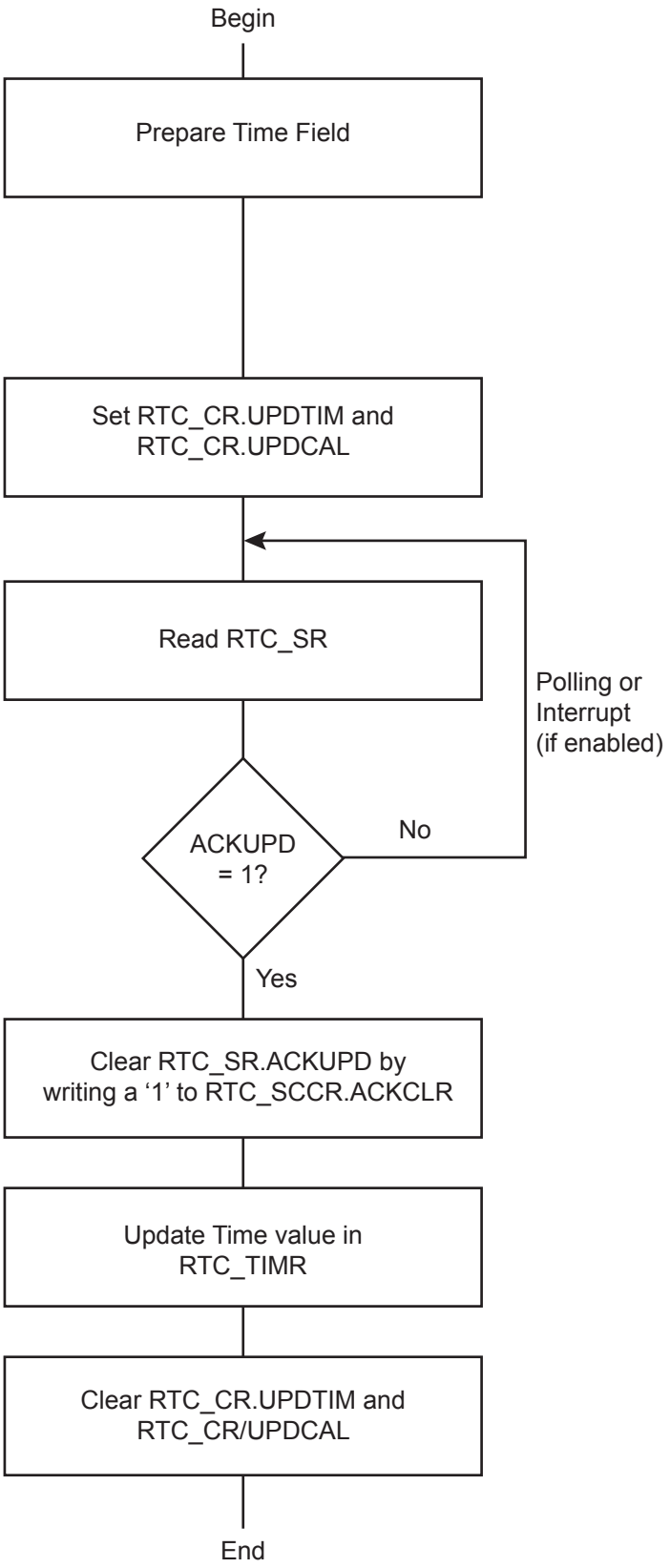


Figure 33.5. UTC Mode Update Sequence



33.5.7. 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 33.6. Calibration Circuitry

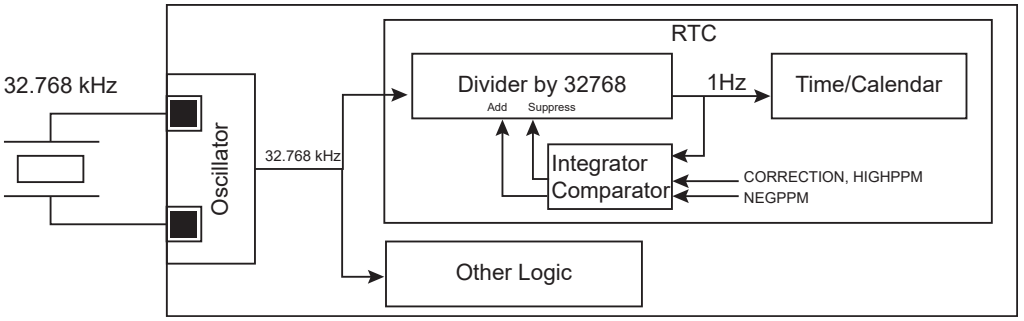
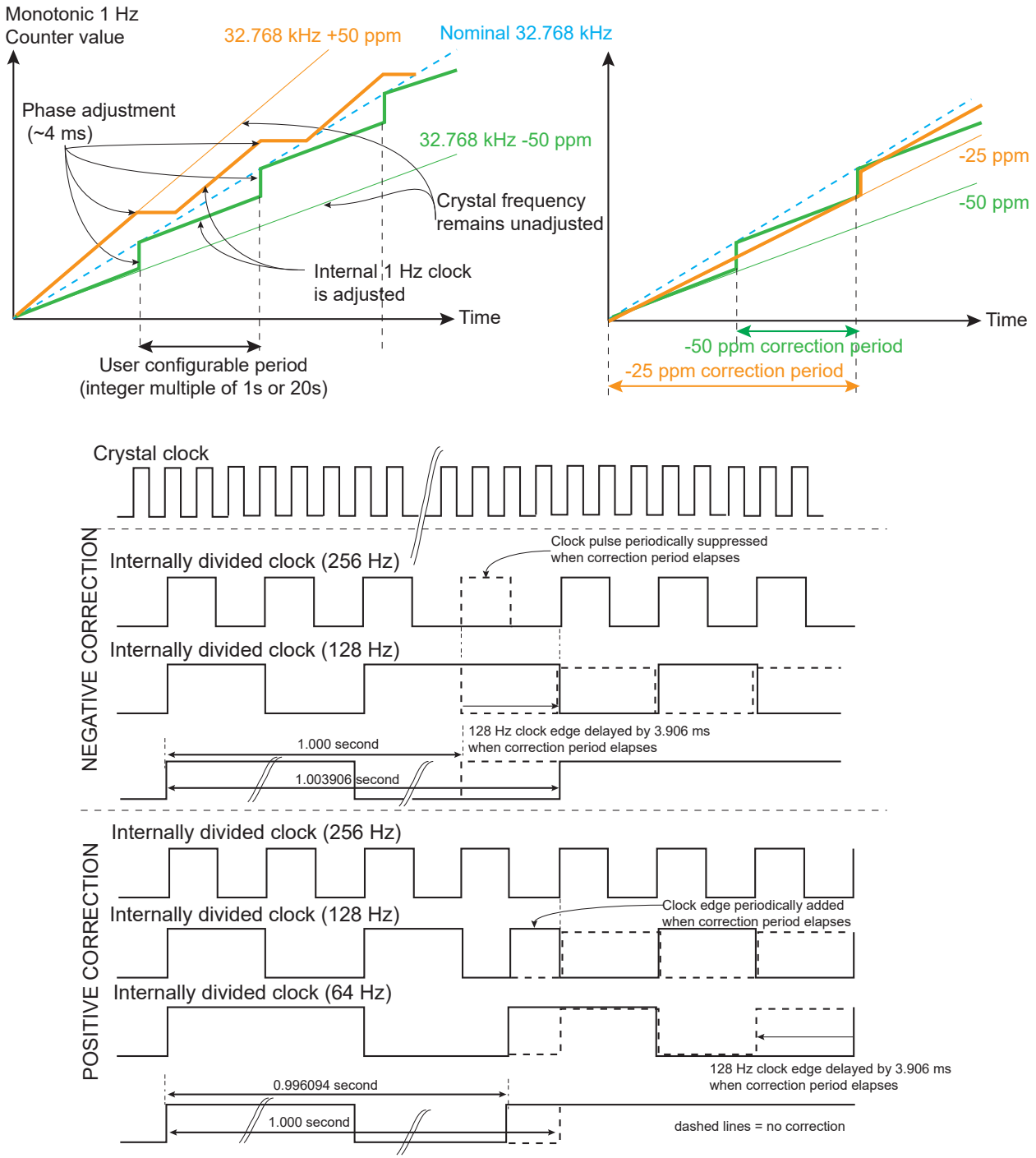


Figure 33.7. Calibration Circuitry Waveforms



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.

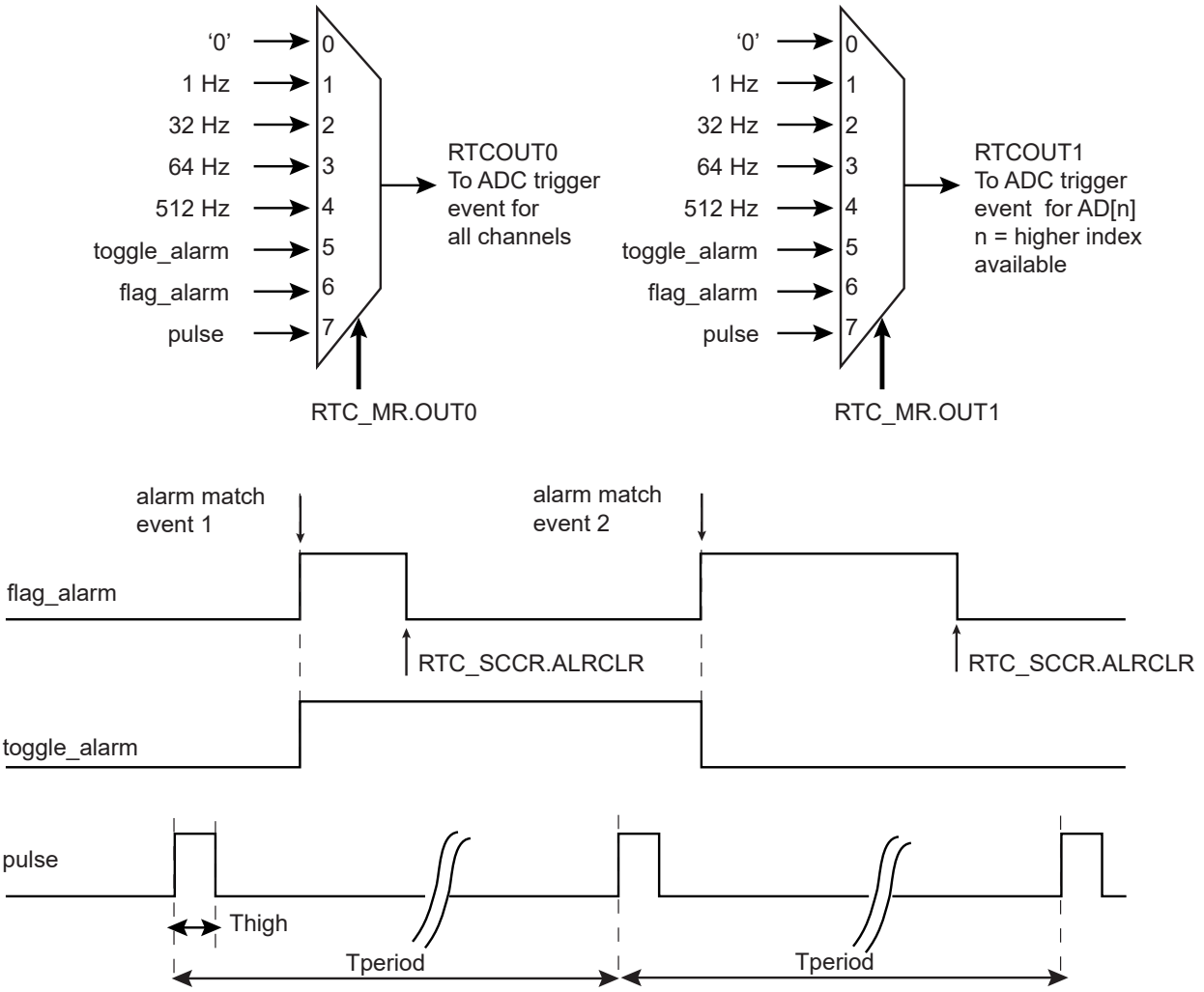
33.5.8. Waveform Generation

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 RTC waveforms are internally routed to ADC trigger events and those events have a source driver selected among five possibilities. Two different triggers can be generated at a time, the first one is configurable in RTC_MR.OUT0, while the second trigger is configurable in RTC_MR.OUT1. RTC_MR.OUT0 manages the trigger for channel AD[n:0] (where n is the higher index available (last channel)), while RTC_MR.OUT1 manages the channel AD[n] only for specific modes. Refer to ADC_MR.TRGSEL in section “Analog-to-Digital Converter (ADC) Controller” for selection of the measurement triggers and associated modes of operation.

When RTC_MR.OUTx is set to 0, the associated output is stuck at 0 (reset value that can be used at any time to disable the waveform generation).

Figure 33.8. Waveform Generation for ADC Trigger Event



33.5.9. Tamper Control Registers and Detection Logic

The WKUP pins used for fast wake-up in PMC are also routed to the tamper detection logic. Any WKUP pin which is not already configured as source of fast wake-up can be configured and selected as a source of a tamper event.

The tamper event can be used to immediately clear (no peripheral clock required) the content of the GPBR, clear the keys stored in AES/TDES, clear the scrambling keys of QSPI/SDR/DDR/SMC memory controllers. Each of these peripherals embeds a configuration bit to allow/disallow the clear on tamper event.

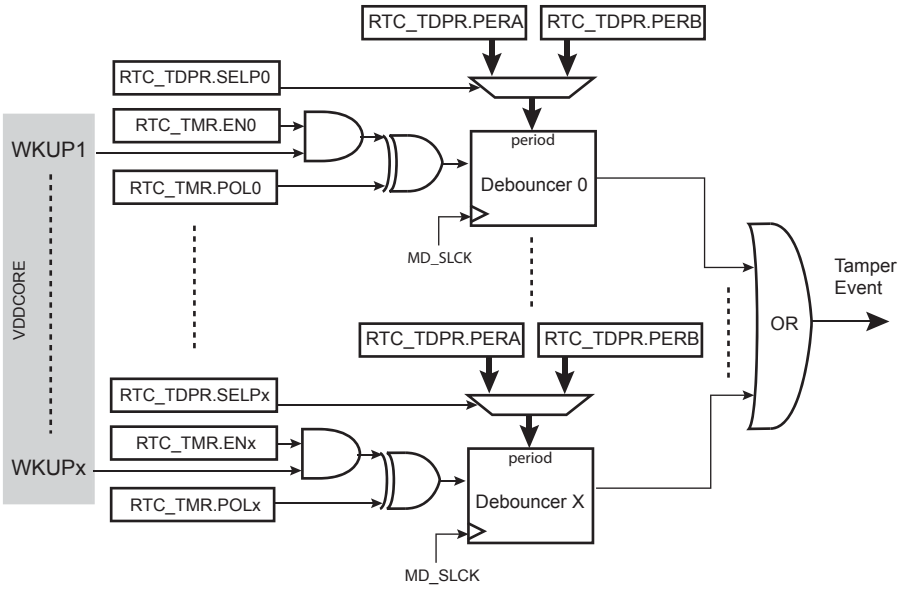
The polarity of each of the WKUP pins can be configured.

Each of the WKUP pins are debounced prior to create the tamper event.

The tamper event is asserted when one of the lines matches the configured polarity after the debouncing period (see the figure below).

The WKUP pins routed to tamper detection logic are all located on VDDCORE domain, thus there is no tamper detection event when the product is in Backup mode.

Figure 33.9. Tamper Detection Circuitry



To enable WKUPx pin to be a source of tamper event, not already configured for a fast wake-up (refer to PMC configuration), the bit RTC_TMR.ENx must be written to 1.

The polarity of the WKUPx pin is configured in RTC_TMR.POLx.

Two debounce periods can be defined by configuring the fields RTC_TDPR.PERA and RTC_TDPR.PERB.

For each WKPUPx pin, the debounce period can be selected from either RTC_TDPR.PERA or RTC_TDPR.PERB by configuring the bit RTC_TDPR.SELPx.

For safety/security reasons, it is possible to lock the tamper configuration registers by writing RTC_TMR.TLOCK=1. Once written to 1, the only way to clear this bit is to perform a VDDCORE reset.

33.5.10. 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.

33.6. 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		HRMOD
0x08	RTC_TIMR (DEFAULT_MODE)	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 (DEFAULT_MODE)	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 (DEFAULT_MODE)	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

Register Summary (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 ... 0x57	Reserved									
0x58	RTC_TMR	31:24	TRLOCK							
		23:16	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
		15:8								
		7:0	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
0x5C	RTC_TDPR	31:24								
		23:16	SELP7	SELP6	SELP5	SELP4	SELP3	SELP2	SELP1	SELP0
		15:8								
		7:0	PERB[3:0]				PERA[3:0]			
0x60 ... 0xAF	Reserved									
0xB0	RTC_TSTR0 (DEFAULT_MODE)	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_TSDRx (DEFAULT_MODE0	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								
		15:8								
		7:0	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0
0xBC	RTC_TSTR1 (DEFAULT_MODE)	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_TSDRx (DEFAULT_MODE1	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								
		15:8								
		7:0	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0

33.6.1. 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	START_UPDATE	Stops the RTC time counting.

33.6.2. 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		HRMOD
Access				R/W		R/W		R/W
Reset				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] 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

Value	Name	Description
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

Bits 18:16 – OUT0[2:0] 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:

$$\text{CORRECTION} = \frac{3906}{20 \times \text{ppm}} - 1$$

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:

$$\text{CORRECTION} = \frac{3906}{\text{ppm}} - 1$$

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

Value	Name	Description
0	DISABLED	Gregorian calendar
1	ENABLED	UTC format

Bit 0 – HRMOD 12-/24-hour Mode

Value	Name	Description
0	24HOURS	24-hour mode is selected.
1	AMPM	12-hour mode is selected.

33.6.3. RTC Time Register (Default Mode)

Name: RTC_TIMR (DEFAULT_MODE)
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

In UTC mode, this register view is not relevant, see 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.

33.6.4. 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.

33.6.5. RTC Calendar Register

Name: RTC_CALR
Offset: 0x0C
Reset: 0x01E11320
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			0	0	0	0	0	1

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	1	1	1	0	0	0	0	1

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	0	0	0	1	0	0	1	1

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		0	1	0	0	0	0	0

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).
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.

33.6.6. RTC Time Alarm Register (Default Mode)

Name: RTC_TIMALR (DEFAULT_MODE)
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.

33.6.7. 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.

33.6.8. RTC Calendar Alarm Register (Default Mode)

Name: RTC_CALALR (DEFAULT_MODE)
Offset: 0x14
Reset: 0x01010000
Property: Read/Write

In UTC mode, this register view is not relevant, see 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.

33.6.9. 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.

33.6.10. RTC Status Register

Name: RTC_SR
Offset: 0x18
Reset: 0x00000004
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	1	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.

33.6.11. 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

33.6.12. 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

33.6.13. 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

33.6.14. 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

33.6.15. 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.

33.6.16. RTC TimeStamp Time Register 0 (Default Mode)

Name: RTC_TSTR0 (DEFAULT_MODE)
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 longer possible to know the exact number of tamper events.
If this field is not null, this implies that at least one tamper event occurred since the 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)

33.6.17. 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.

33.6.18. RTC TimeStamp Time Register 1 (Default Mode)

Name: RTC_TSTR1 (DEFAULT_MODE)
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)

33.6.19. 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.

33.6.20. RTC TimeStamp Date Register (Default Mode)

Name: RTC_TSDRx (DEFAULT_MODE)
Offset: 0xB4 + n*0x0C [n=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)

33.6.21. 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.

33.6.22. 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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DET7	DET6	DET5	DET4	DET3	DET2	DET1	DET0
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 – DETx Tamper Detection on VDDCORE WKUP[8:1] (cleared on read)

33.6.23. RTC Tamper Mode Register

Name: RTC_TMR
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TRLOCK							
Access	W							
Reset	0							

Bit	23	22	21	20	19	18	17	16
	POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Access	R/W	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
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Access	R/W	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 – TRLOCK Tamper Registers Lock (Write-once, cleared by VDDCORE reset)

Value	Name	Description
0	UNLOCKED	RTC_TMR and RTC_TDPR can be written.
1	LOCKED	RTC_TMR and RTC_TDPR cannot be written until the next VDDCORE domain reset.

Bits 16, 17, 18, 19, 20, 21, 22, 23 – POLx WKUPx+1 Polarity

Value	Name	Description
0	LOW	If the source of tamper remains low for a debounce period, a tamper event is generated.
1	HIGH	If the source of tamper remains high for a debounce period, a tamper event is generated.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – ENx WKUPx+1 Tamper Source Enable

Value	Name	Description
0	DISABLE	WKUP pin index x+1 is not enabled as a source of tamper.
1	ENABLE	WKUP pin index x+1 is enabled as a source of tamper.

33.6.24. RTC Tamper Debounce Period Register

Name: RTC_TDPR
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
	SELP7	SELP6	SELP5	SELP4	SELP3	SELP2	SELP1	SELP0
Access	R/W	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
	PERB[3:0]				PERA[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, 17, 18, 19, 20, 21, 22, 23 – SELPx WKUPx+1 Debounce Period Selection

Value	Name	Description
0	SEL_PA	WKUP pin index x+1 is debounced with PERA period.
1	SEL_PB	WKUP pin index x+1 is debounced with PERB period.

Bits 7:4 – PERB[3:0] Debounce Period B

Value	Name	Description
0	MD_SLCK_2	The source of tamper must remain active for at least 2 monitoring domain slow clock cycles to generate a tamper event.
1	MD_SLCK_4	The source of tamper must remain active for at least 4 monitoring domain slow clock cycles to generate a tamper event.
2	MD_SLCK_8	The source of tamper must remain active for at least 8 monitoring domain slow clock cycles to generate a tamper event.
3	MD_SLCK_16	The source of tamper must remain active for at least 16 monitoring domain slow clock cycles to generate a tamper event.
4	MD_SLCK_32	The source of tamper must remain active for at least 32 monitoring domain slow clock cycles to generate a tamper event.
5	MD_SLCK_64	The source of tamper must remain active for at least 64 monitoring domain slow clock cycles to generate a tamper event.
6	MD_SLCK_128	The source of tamper must remain active for at least 128 monitoring domain slow clock cycles to generate a tamper event.
7	MD_SLCK_256	The source of tamper must remain active for at least 256 monitoring domain slow clock cycles to generate a tamper event.

Bits 3:0 – PERA[3:0] Debounce Period A

Value	Name	Description
0	MD_SLCK_2	The source of tamper must remain active for at least 2 monitoring domain slow clock cycles to generate a tamper event.
1	MD_SLCK_4	The source of tamper must remain active for at least 4 monitoring domain slow clock cycles to generate a tamper event.
2	MD_SLCK_8	The source of tamper must remain active for at least 8 monitoring domain slow clock cycles to generate a tamper event.
3	MD_SLCK_16	The source of tamper must remain active for at least 16 monitoring domain slow clock cycles to generate a tamper event.
4	MD_SLCK_32	The source of tamper must remain active for at least 32 monitoring domain slow clock cycles to generate a tamper event.
5	MD_SLCK_64	The source of tamper must remain active for at least 64 monitoring domain slow clock cycles to generate a tamper event.
6	MD_SLCK_128	The source of tamper must remain active for at least 128 monitoring domain slow clock cycles to generate a tamper event.
7	MD_SLCK_256	The source of tamper must remain active for at least 256 monitoring domain slow clock cycles to generate a tamper event.

34. Shutdown Controller (SHDWC)

34.1. Description

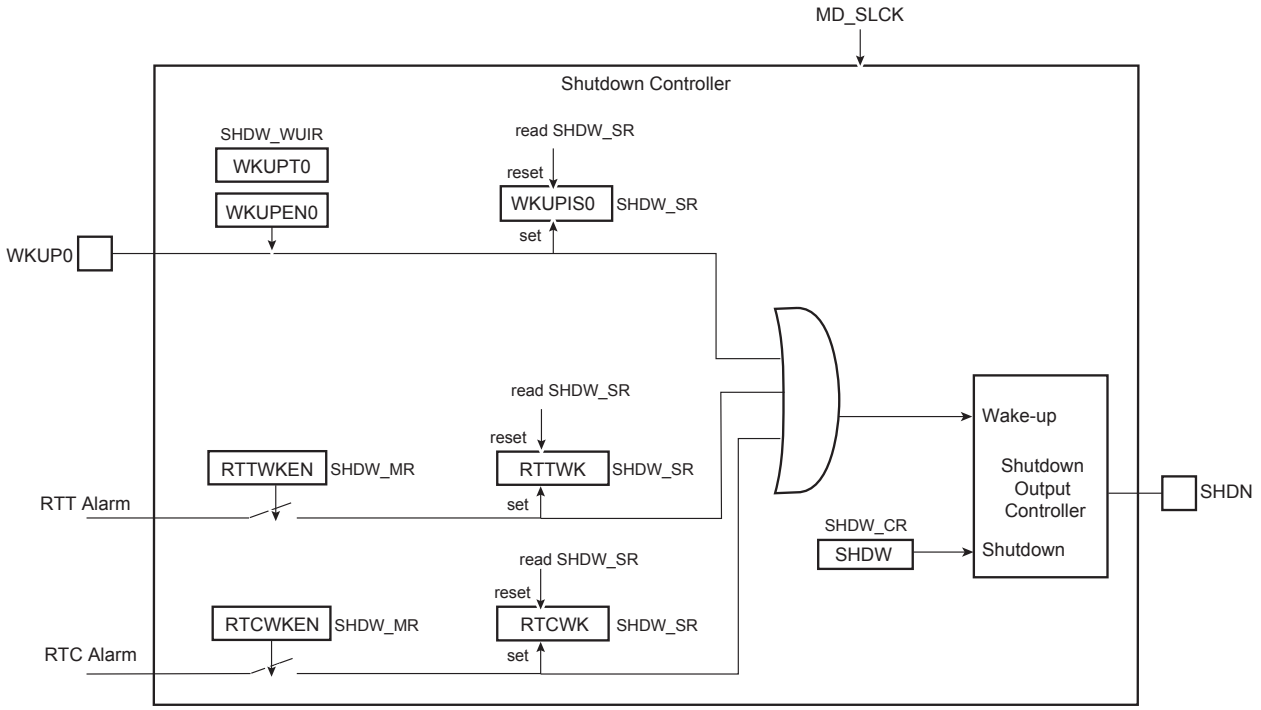
The Shutdown Controller (SHDWC) controls the SHDN output signal (to enable and disable an external power supply circuit), and manages the wake-up events detection.

34.2. 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 input pins, and internal wake-up event from RTC and RTT

34.3. Block Diagram

Figure 34.1. SHDWC Block Diagram



34.4. I/O Lines Description

Table 34.1. I/O Lines Description

Name	Description	Type
WKUP0	Wake-up input	Input
SHDN	Shutdown output	Output

34.5. Product Dependencies

34.5.1. 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.

34.6. 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 (WKUP0) connect to any push-buttons or signal that wake up the system.

The software is able to control the pin SHDN by writing the 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.

34.6.1. Wake-Up Inputs

Any level change on the WKUP pin can trigger a wake-up. Wake-up is configured in the Mode register (SHDW_MR) and Wake-Up Inputs register (SHDW_WUIR). The transition detector can be programmed to detect either a positive or negative transition on the WKUP pin. The detection can also be disabled. Programming is performed by enabling the bit Wake-up Input (WKUPEN0) and defining the bit Wake-up Input Type (WKUPT0) in SHDW_WUIR.

Moreover, a debouncing circuit can be programmed for WKUP. The debouncing circuit filters pulses on 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. WKUPIS0 of the Status register (SHDW_SR) reports the detection of the programmed events on WKUP with a reset after the read of SHDW_SR.

The SHDWC can be programmed to activate the wake-up using the RTC and RTT alarms (detection of the rising edge event is synchronized with SLCK). This is done by writing SHDW_MR.RTCWKEN and RTTWKEN. When enabled, the detection of RTC and RTT alarms is reported in SHDW_SR.RTCWK and RTTWK. They are cleared after reading SHDW_SR. When using the RTC and RTT alarms to wake up the system, the user must ensure that the RTC and RTT alarm status flags are cleared before shutting down the system. Otherwise, no rising edge of the status flags may be detected and the wake-up fails.

34.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	SHDW_CR	31:24	KEY[7:0]							
		23:16								
		15:8								
		7:0								SHDW
0x04	SHDW_MR	31:24						WKUPDBC[2:0]		
		23:16							RTCWKEN	RTTWKEN
		15:8								
		7:0								
0x08	SHDW_SR	31:24								
		23:16								WKUPIS0
		15:8								
		7:0			RTCWK	RTTWK				WKUPS
0x0C	SHDW_WUIR	31:24								
		23:16								WKUPT0
		15:8								
		7:0								WKUPEN0

34.7.1. 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
Access								
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 0 – SHDW Shutdown Command

Value	Description
0	No effect.
1	If KEY value is correct, asserts the SHDN pin.

34.7.2. 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
							RTCWKEN	RTTWKEN
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
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	WKUP shall be in its active state for at least 3 MD_SLCK periods
2	32_SLCK	WKUP shall be in its active state for at least 32 MD_SLCK periods
3	512_SLCK	WKUP shall be in its active state for at least 512 MD_SLCK periods
4	4096_SLCK	WKUP shall be in its active state for at least 4,096 MD_SLCK periods
5	32768_SLCK	WKUP shall be in its active state for at least 32,768 MD_SLCK periods

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.

34.7.3. 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
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 16 – WKUPIS0 Wake-up 0 Input Status

Value	Name	Description
0	DISABLE	The wake-up 0 input is disabled, or was inactive at the time the debouncer triggered a wake-up event.
1	ENABLE	The wake-up 0 input was active at the time the debouncer triggered a wake-up event.

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 Wake-up Status

Value	Name	Description
0	NO	No wake-up due to the assertion of the wake-up pins has occurred since the last read of SHDW_SR.
1	PRESENT	At least one wake-up due to the assertion of the wake-up pins has occurred since the last read of SHDW_SR

34.7.4. 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
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 16 – WKUPT0 Wake-up 0 Input Type

Value	Name	Description
0	LOW	A falling edge followed by a low level on the wake-up 0 input, for a period defined by WKUPDBC, forces wake-up of the core power supply.
1	HIGH	A rising edge followed by a high level on the wake-up 0 input, for a period defined by WKUPDBC, forces wake-up of the core power supply.

Bit 0 – WKUPEN0 Wake-up 0 Input Enable

Value	Name	Description
0	DISABLE	The wake-up 0 input has no wake-up effect.
1	ENABLE	The wake-up 0 input forces wake-up of the core power supply.

35. Periodic Interval Timer (PIT)

35.1. Description

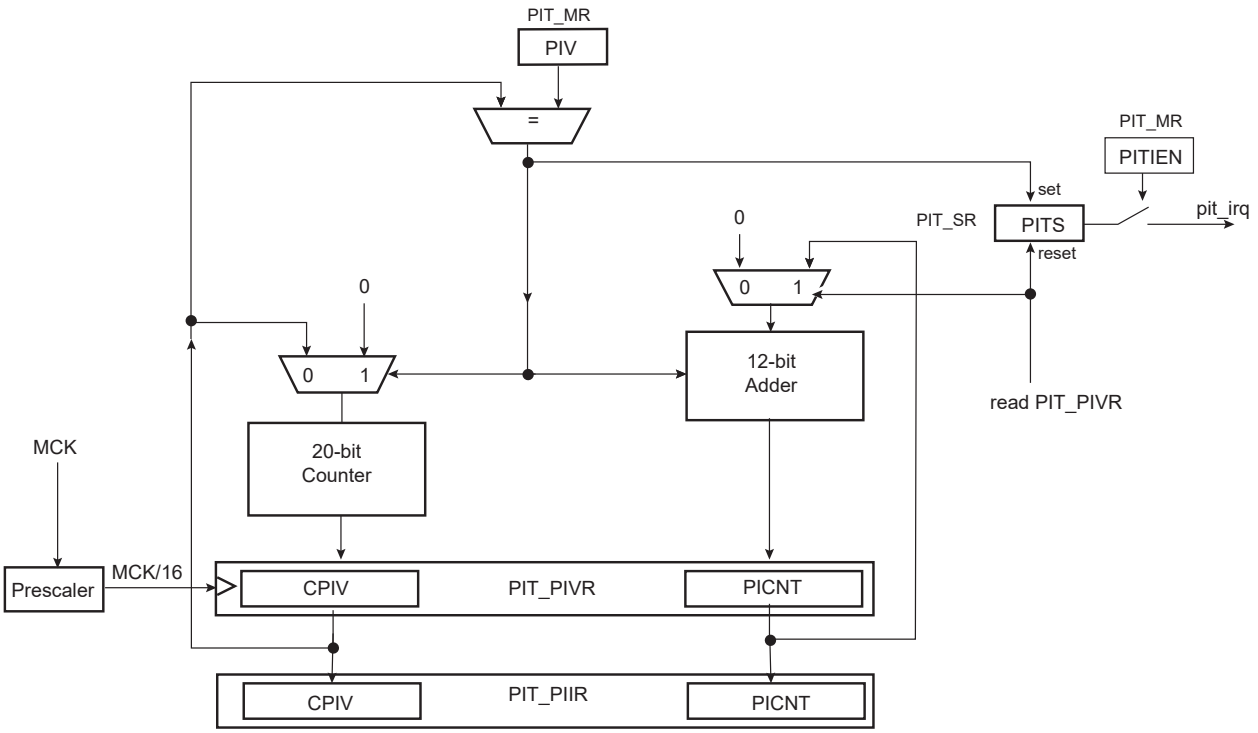
The Periodic Interval Timer (PIT) provides the operating system's scheduler interrupt. It is designed to offer maximum accuracy and efficient management, even for systems with long response time.

35.2. Embedded Characteristics

- 20-bit Programmable Counter plus 12-bit Interval Counter
- Reset-on-read Feature
- Both Counters Work on Main System Bus Clock/16

35.3. Block Diagram

Figure 35.1. Periodic Interval Timer



35.4. Functional Description

The Periodic Interval Timer (PIT) provides periodic interrupts for use by operating systems.

The PIT provides a programmable overflow counter and a reset-on-read feature. It is built around two counters: a 20-bit CPIV counter and a 12-bit PICNT counter. Both counters work at Main System Bus Clock /16.

The 20-bit CPIV counter increments from 0 up to a programmable overflow value set in the field PIV of the Mode register (PIT_MR). When the counter CPIV reaches this value, it resets to 0 and increments PICNT. The status bit PITS in the Status register (PIT_SR) rises and triggers an interrupt, provided the interrupt is enabled (PIT_MR.PITIEN).

Writing a new PIV value in PIT_MR does not reset/restart the counters.

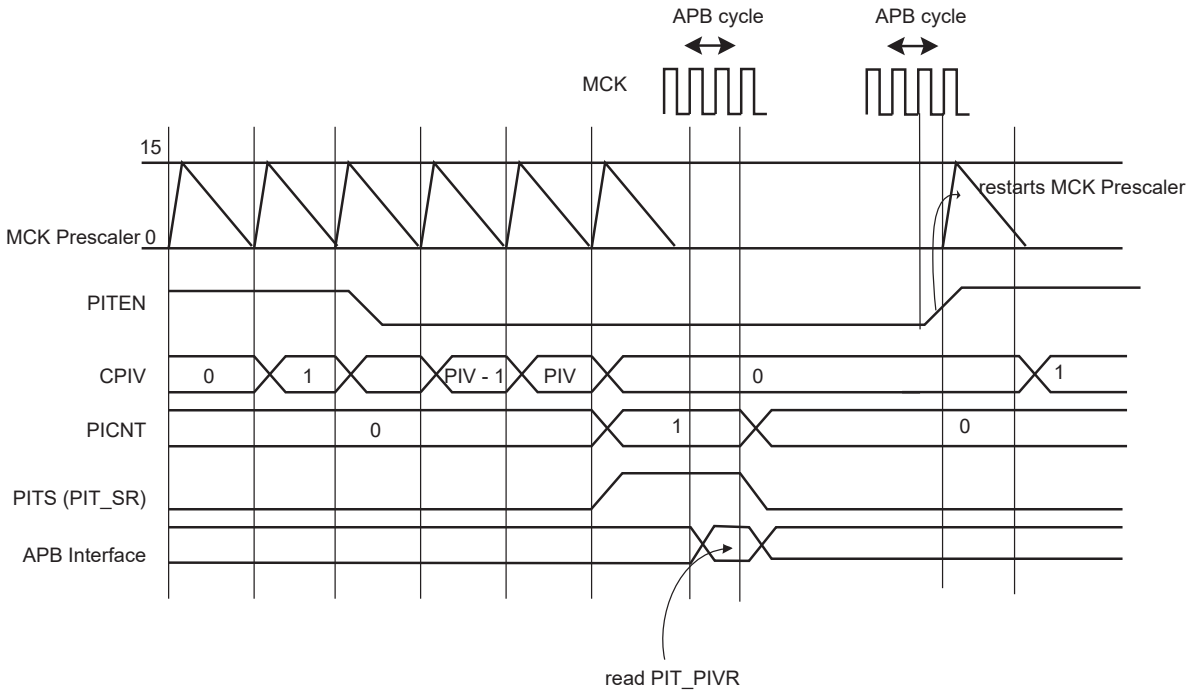
When CPIV and PICNT values are obtained by reading the Periodic Interval Value register (PIT_PIVR), the overflow counter (PICNT) is reset and PIT_SR.PITS is cleared, thus acknowledging the interrupt. The value of PICNT gives the number of periodic intervals elapsed since the last read of PIT_PIVR.

When CPIV and PICNT values are obtained by reading the Periodic Interval Image Register (PIT_PIIR), there is no effect on the counters CPIV and PICNT, nor on the bit PITS. For example, a profiler can read PIT_PIIR without clearing any pending interrupt, whereas a timer interrupt clears the interrupt by reading PIT_PIVR.

The PIT may be enabled/disabled using PIT_MR.PITEN (disabled on reset). PITEN only becomes effective when the CPIV value is 0. The figure below illustrates the PIT counting. After PITEN is reset (PITEN = 0), the CPIV goes on counting until the PIV value is reached, and is then reset. PIT restarts counting, only if PITEN is set again.

The PIT is stopped when the core enters debug state.

Figure 35.2. Enabling/Disabling PIT with PITEN



35.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PIT_MR	31:24							PITIEN	PITEN
		23:16					PIV[19:16]			
		15:8	PIV[15:8]							
		7:0	PIV[7:0]							
0x04	PIT_SR	31:24								
		23:16								
		15:8								
		7:0								PITS
0x08	PIT_PIVR	31:24	PICNT[11:4]							
		23:16	PICNT[3:0]				CPIV[19:16]			
		15:8	CPIV[15:8]							
		7:0	CPIV[7:0]							
0x0C	PIT_PIIR	31:24	PICNT[11:4]							
		23:16	PICNT[3:0]				CPIV[19:16]			
		15:8	CPIV[15:8]							
		7:0	CPIV[7:0]							

35.5.1. Periodic Interval Timer Mode Register

Name: PIT_MR
Offset: 0x00
Reset: 0x000FFFFF
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
							PITIEN	PITEN
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
					PIV[19:16]			
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bit	15	14	13	12	11	10	9	8
	PIV[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
	PIV[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 25 – PITIEN Period Interval Timer Interrupt Enable

Value	Description
0	The bit PITS in PIT_SR has no effect on the interrupt.
1	The bit PITS in PIT_SR asserts an interrupt.

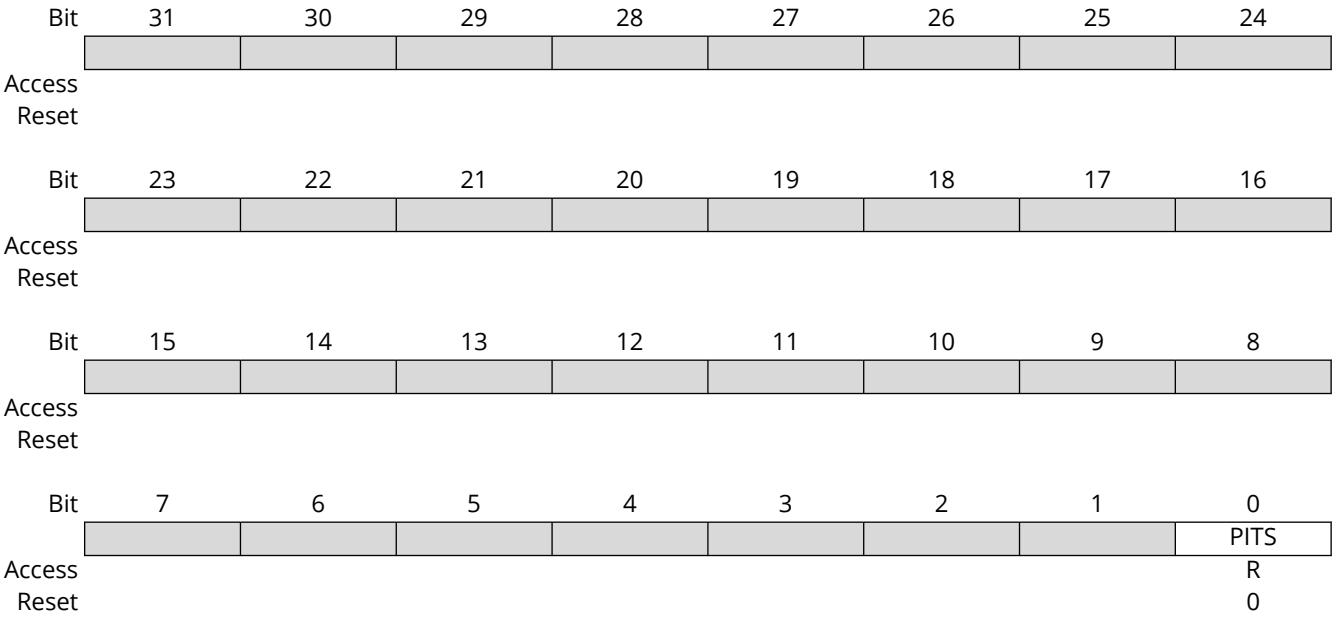
Bit 24 – PITEN Period Interval Timer Enabled

Value	Description
0	The Periodic Interval Timer is disabled when the PIV value is reached.
1	The Periodic Interval Timer is enabled.

Bits 19:0 – PIV[19:0] Periodic Interval Value
Defines the value compared with the primary 20-bit counter of the Periodic Interval Timer (CPIV). The period is equal to (PIV + 1).

35.5.2. Periodic Interval Timer Status Register

Name: PIT_SR
Offset: 0x04
Reset: 0x00000000
Property: Read-only



Bit 0 – PITS Periodic Interval Timer Status

Value	Description
0	The Periodic Interval timer has not reached PIV since the last read of PIT_PIVR.
1	The Periodic Interval timer has reached PIV since the last read of PIT_PIVR.

35.5.3. Periodic Interval Timer Value Register

Name: PIT_PIVR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

Reading this register clears PITS in PIT_SR.

Bit	31	30	29	28	27	26	25	24
	PICNT[11:4]							
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
	PICNT[3:0]				CPIV[19:16]			
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
	CPIV[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
	CPIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:20 – PICNT[11:0] Periodic Interval Counter
Returns the number of occurrences of periodic intervals since the last read of PIT_PIVR.

Bits 19:0 – CPIV[19:0] Current Periodic Interval Value
Returns the current value of the periodic interval timer.

35.5.4. Periodic Interval Timer Image Register

Name: PIT_PIIR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	PICNT[11:4]							
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
	PICNT[3:0]				CPIV[19:16]			
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
	CPIV[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
	CPIV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:20 – PICNT[11:0] Periodic Interval Counter
Returns the number of occurrences of periodic intervals since the last read of PIT_PIVR.

Bits 19:0 – CPIV[19:0] Current Periodic Interval Value
Returns the current value of the periodic interval timer.

36. 64-bit Periodic Interval Timer (PIT64B)

36.1. Description

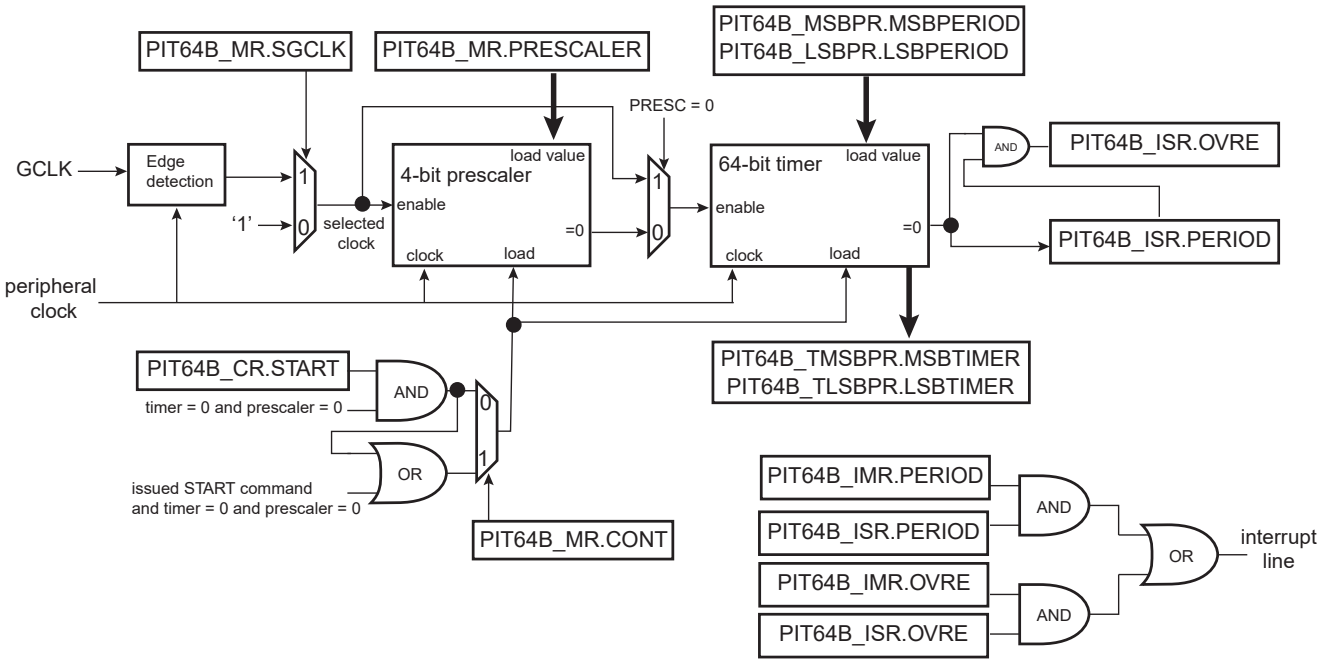
The 64-bit Periodic Interval Timer (PIT64B) provides the operating system scheduler interrupt, as well as any periodic source of interrupt to software. It is designed to offer maximum accuracy and efficient management, even for systems with long response times.

36.2. Embedded Characteristics

- 4-bit Prescaler
- 64-bit Timer
- Single Shot or Continuous Mode
- Safety/Security Access Reports
- Register Write protection

36.3. Block Diagram

Figure 36.1. PIT64B Block Diagram



36.4. Product Dependencies

36.4.1. 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.

36.4.2. 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.

36.5. Functional Description

36.5.1. 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.

36.5.2. 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 36.2)

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 36.3). 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 36.2. Single Waveform in Single Period Mode if bit PIT64B_MR.SMOD=0

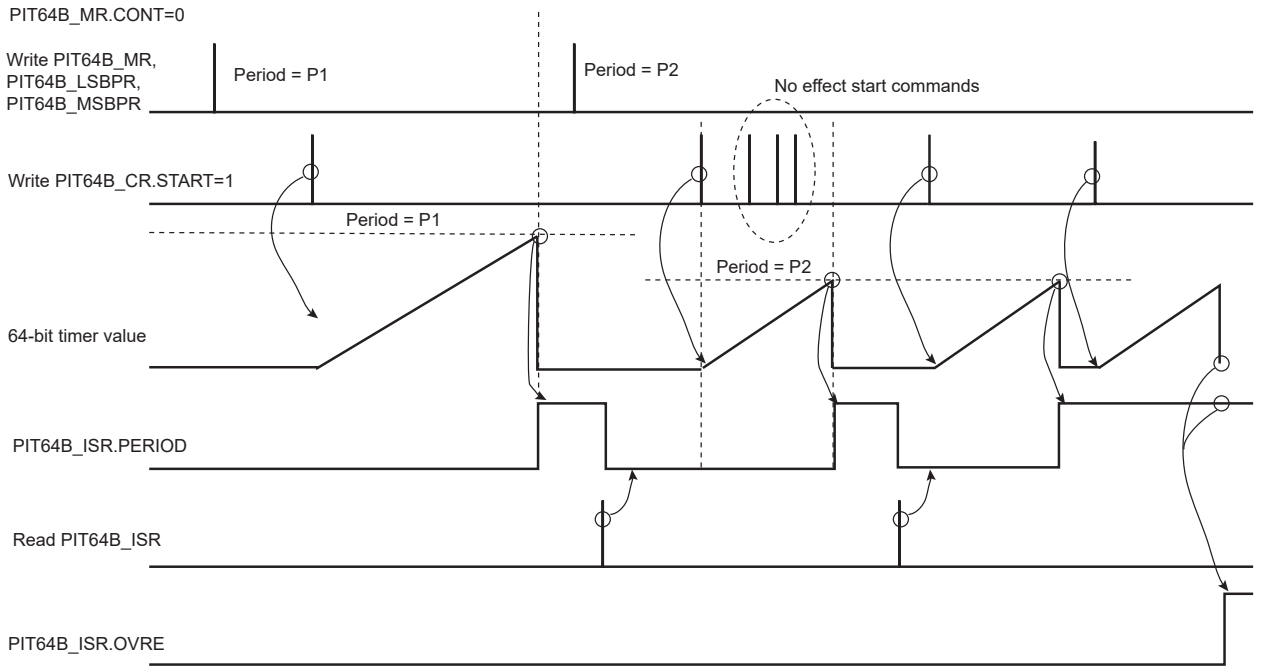
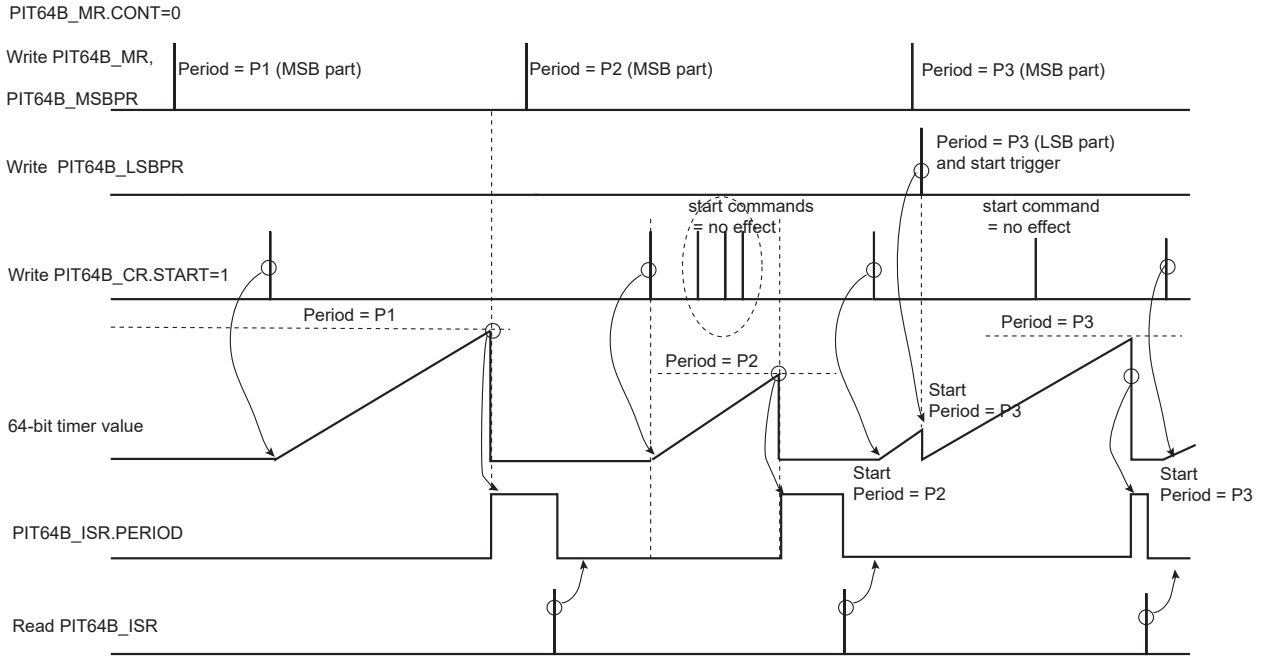


Figure 36.3. Waveform in Single Period Mode if bit PIT64B_MR.SMOD=1



36.5.3. 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 36.4). 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 36.5). 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 36.4. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=0

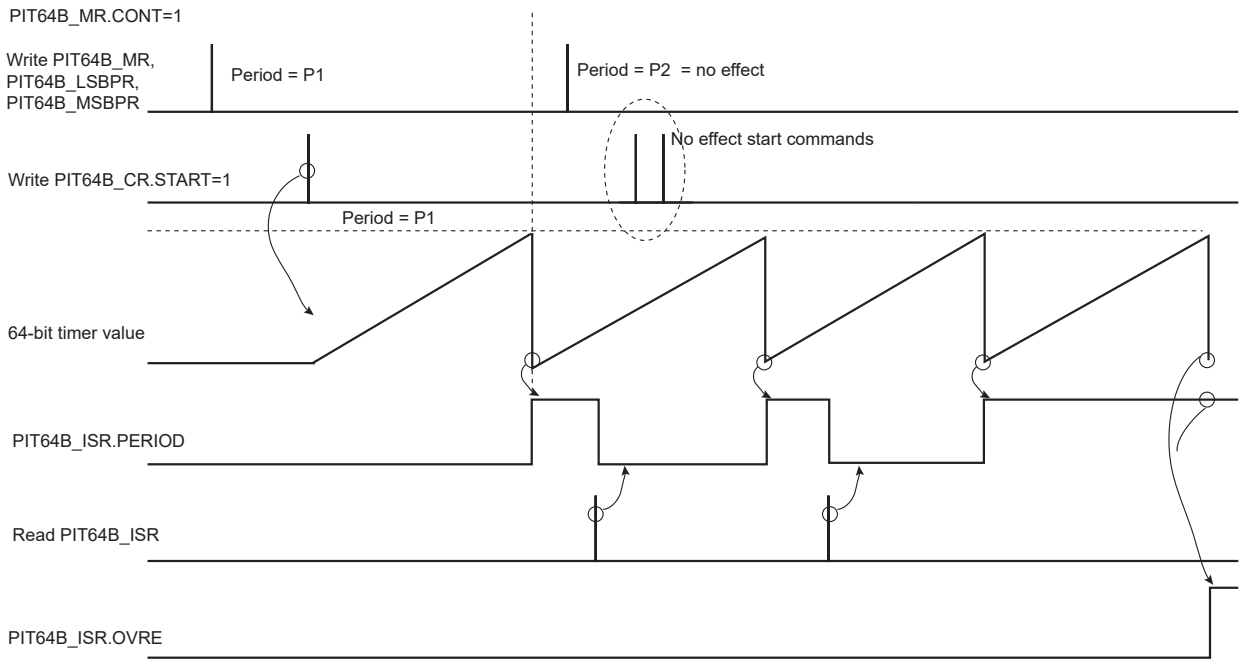
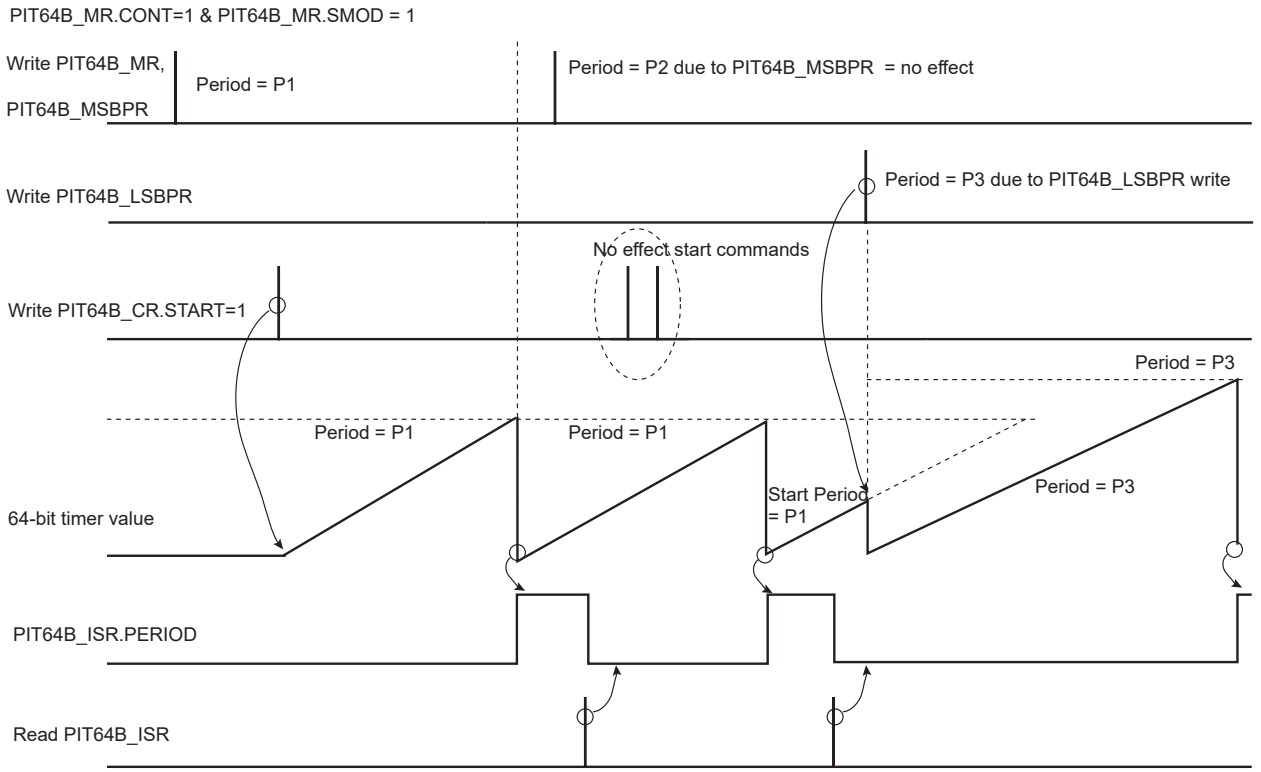


Figure 36.5. Waveform in Continuous Period Mode if bit PIT64B_MR.SMOD=1



36.5.4. 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). For example, 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.

36.5.5. 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

36.6. 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

36.6.1. 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
Access								W
Reset								–

Bit	7	6	5	4	3	2	1	0
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 Single Period Mode and Continuous Period Mode).

36.6.2. 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.

Value	Description
1	Continuous periodic interrupts are generated after a single START command.

36.6.3. 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}.

36.6.4. 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}.

36.6.5. 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

36.6.6. 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

36.6.7. 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

36.6.8. 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.

36.6.9. 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.

36.6.10. 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.

36.6.11. 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 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).

36.6.12. 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.
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.

37. Debug Unit (DBGU)

37.1. Description

The Debug Unit (DBGU) provides a single entry point from the processor for access to all the debug capabilities of the product.

The DBGU features a two-pin UART that can be used for communication and trace purposes and offers an ideal medium for in-situ programming solutions.

Moreover, the association with a DMA controller permits packet handling for these tasks with processor time reduced to a minimum.

The DBGU also makes the Debug Communication Channel (DCC) signals provided by the In-circuit Emulator of the Arm processor visible to the software. These signals indicate the status of the DCC read and write registers and generate an interrupt to the Arm processor, making possible the handling of the DCC under interrupt control.

Chip identifier registers permit recognition of the device and its revision. These registers indicate the sizes and types of the on-chip memories, as well as the set of embedded peripherals.

A Force NTRST capability enables the software to decide whether to prevent access to the system via the In-circuit Emulator (ICE). This disables system access through the processor's ICE, thus protecting the code stored in ROM by asserting the NTRST line of the processor's ICE.

37.2. Embedded Characteristics

- ICE Access Prevention
- Debug Communication Channel (DCC) Support
- Chip ID Registers
- Two-pin UART
  - Independent receiver and transmitter with a common programmable baud rate generator
  - Baud rate can be driven by processor-independent generic source clock
  - Even, odd, mark or space parity generation
  - Parity, framing and overrun error detection
  - Automatic Echo, Local loopback and Remote Loopback Channel modes
  - Digital filter on receive line
  - Interrupt generation
  - Support for two DMA channels with connection to receiver and transmitter
  - Receiver timeout
  - Register write protection

37.3. Block Diagram

Figure 37.1. DBGU Block Diagram

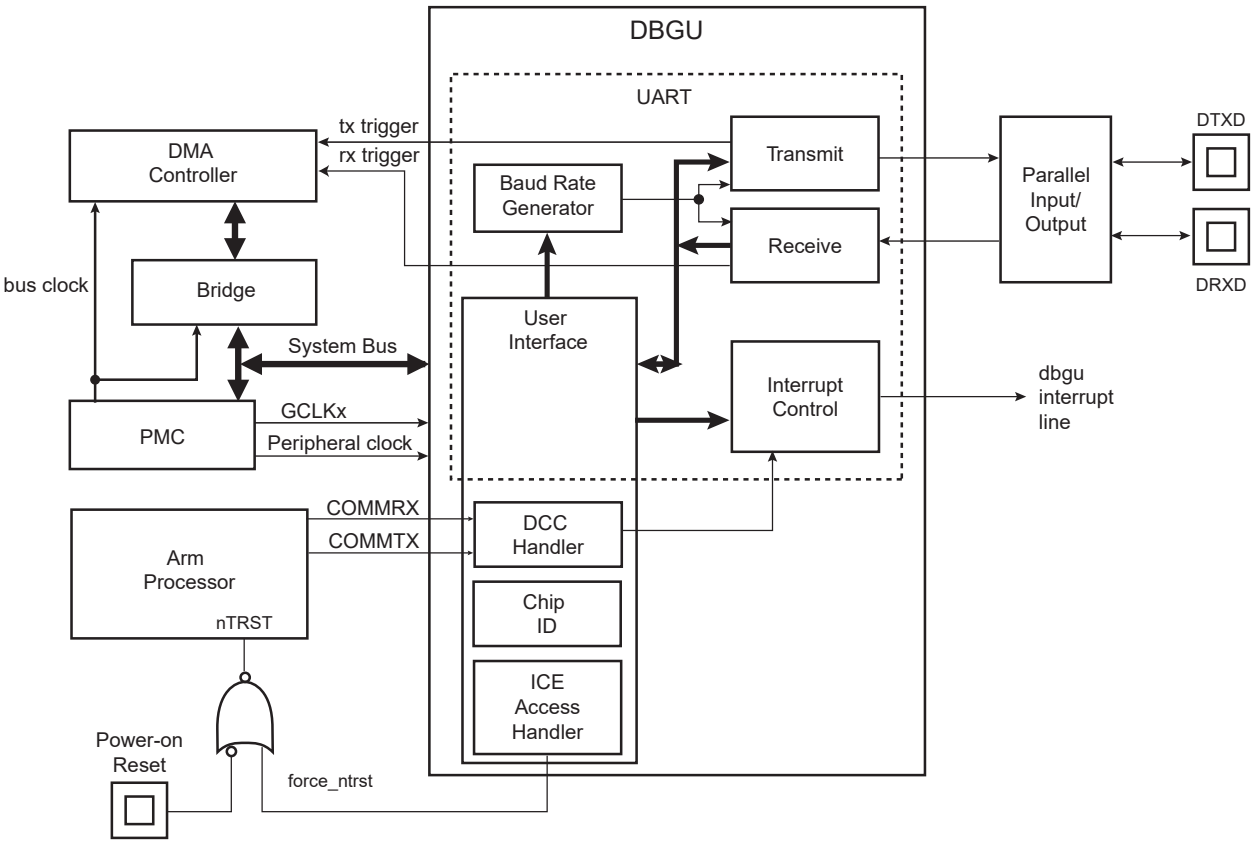


Table 37.1. DBGU Pin Description

Pin Name	Description	Type
DRXD	DBGU Receive Data	Input
DTXD	DBGU Transmit Data	Output

37.4. Product Dependencies

37.4.1. I/O Lines

The DBGU pins are multiplexed with PIO lines. The user must first configure the corresponding PIO Controller to enable I/O line operations of the DBGU.

37.4.2. Power Management

The DBGU clock can be controlled through the Power Management Controller (PMC). In this case, the user must first configure the PMC to enable the DBGU clock.

37.4.3. Interrupt Sources

The DBGU interrupt line is connected to one of the interrupt sources of the Interrupt Controller. Interrupt handling requires programming of the Interrupt Controller before configuring the DBGU.

37.5. Functional Description

The DBGU operates in Asynchronous mode only and supports only 8-bit character handling (with parity). It has no clock pin.

The DBGU is made up of a receiver and a transmitter that operate independently, and a common baud rate generator. Receiver timeout and transmitter time guard are not implemented. However, all the implemented features are compatible with those of a standard USART.

37.5.1. Baud Rate Generator

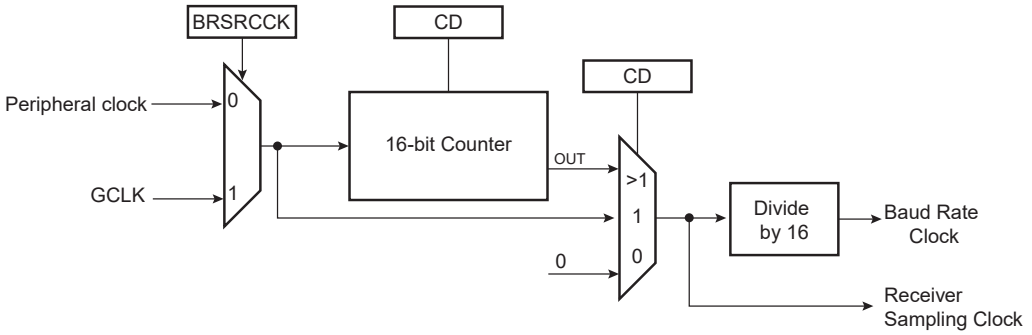
The baud rate generator provides the bit period clock named baud rate clock to both the receiver and the transmitter.

The baud rate clock is the peripheral clock divided by 16 times the clock divisor (CD) value written in the Baud Rate Generator register (DBGU_BRGR). If DBGU_BRGR is set to 0, the baud rate clock is disabled and the DBGU remains inactive. The maximum allowable baud rate is peripheral clock or GCLK divided by 16. The minimum allowable baud rate is peripheral clock divided by (16 x 65536). The clock source driving the baud rate generator (peripheral clock or GCLK) can be selected by writing the bit BRSRCCK in DBGU_MR.

If GCLK is selected, the baud rate is independent of the processor/bus clock. Thus the processor clock can be changed while DBGU is enabled. The processor clock frequency changes must be performed only by programming the field PRES in PMC_MCKR (see PMC section). Other methods to modify the processor/bus clock frequency (PLL multiplier, etc.) are forbidden when DBGU is enabled.

The peripheral clock frequency must be at least three times higher than GCLK.

Figure 37.2. Baud Rate Generator



37.5.2. Receiver

37.5.2.1. Receiver Reset, Enable and Disable

After device reset, the DBGU receiver is disabled and must be enabled before being used. The receiver can be enabled by writing the Control register (DBGU_CR) with the bit RXEN at 1. At this command, the receiver starts looking for a start bit.

The programmer can disable the receiver by writing DBGU_CR with the bit RXDIS at 1. If the receiver is waiting for a start bit, it is immediately stopped. However, if the receiver has already detected a start bit and is receiving the data, it waits for the stop bit before actually stopping its operation.

The receiver can be put in reset state by writing DBGU_CR with the bit RSTRX at 1. In this case, the receiver immediately stops its current operations and is disabled, whatever its current state. If RSTRX is applied when data is being processed, this data is lost.

37.5.2.2. Start Detection and Data Sampling

The DBGU only supports asynchronous operations, and this affects only its receiver. The DBGU receiver detects the start of a received character by sampling the DRXD signal until it detects a valid start bit. A low level (space) on DRXD is interpreted as a valid start bit if it is detected for more than seven cycles of the sampling clock, which is 16 times the baud rate. Hence, a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space which is 7/16 of a bit period or shorter is ignored and the receiver continues to wait for a valid start bit.

When a valid start bit has been detected, the receiver samples the DRXD at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1-bit period) so the bit sampling point is eight cycles (0.5-bit period) after the start of the bit. The first sampling point is therefore 24 cycles (1.5-bit periods) after detecting the falling edge of the start bit.

Each subsequent bit is sampled 16 cycles (1-bit period) after the previous one.

Figure 37.3. Start Bit Detection

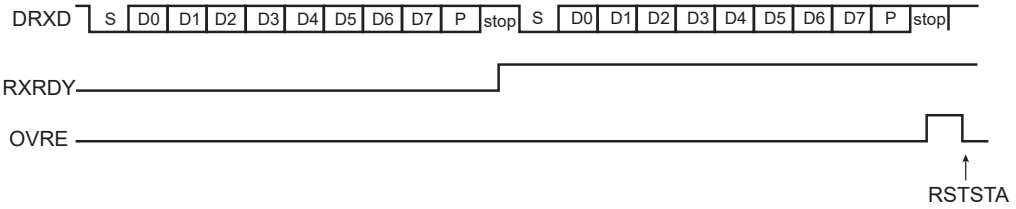
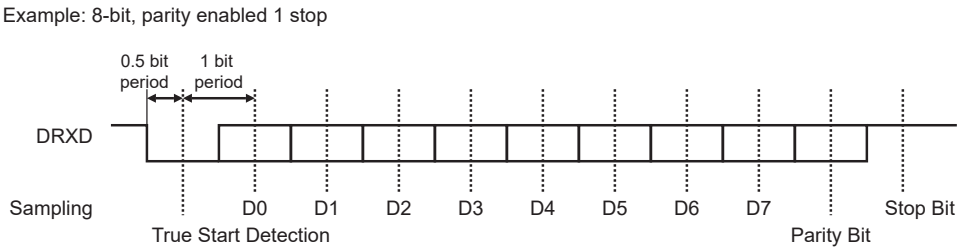


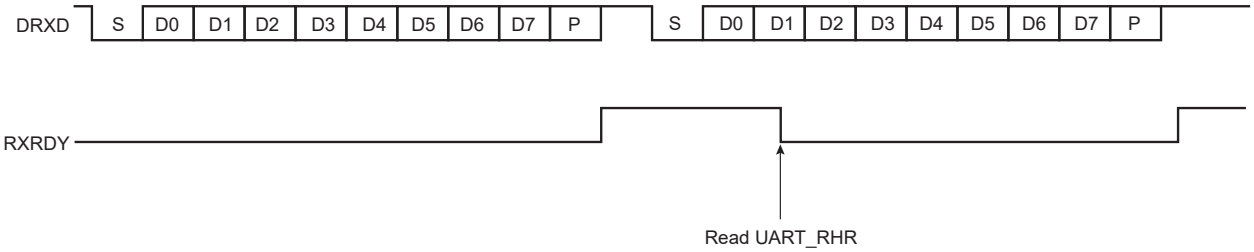
Figure 37.4. Character Reception



37.5.2.3.Receiver Ready

When a complete character is received, it is transferred to the Receive Holding register (DBGU_RHR) and the RXRDY status bit in the Status register (DBGU_SR) is set. The bit RXRDY is automatically cleared when DBGU_RHR is read.

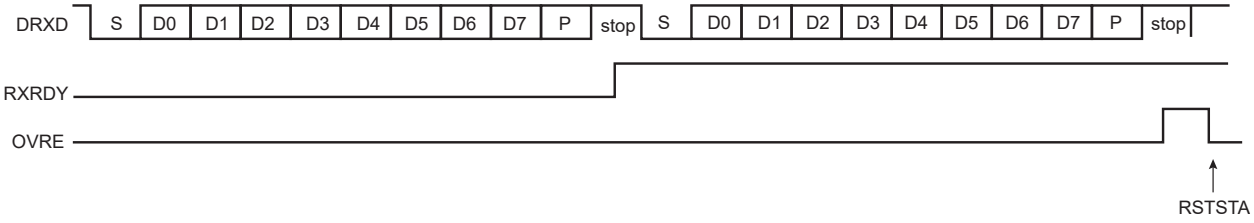
Figure 37.5. Receiver Ready



37.5.2.4.Receiver Overrun

The OVRE status bit in DBGU_SR is set if DBGU_RHR has not been read by the software (or the DMA Controller) since the last transfer, the RXRDY bit is still set and a new character is received. OVRE is cleared when the software writes a 1 to the bit RSTSTA (Reset Status) in DBGU_CR.

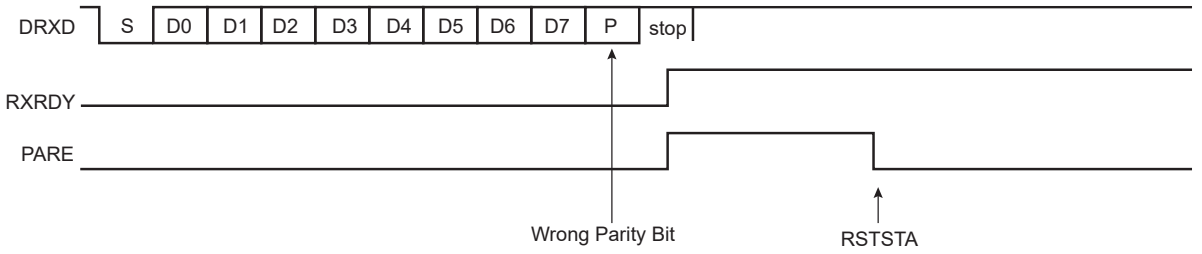
Figure 37.6. Receiver Overrun



37.5.2.5. Parity Error

Each time a character is received, the receiver calculates the parity of the received data bits, in accordance with the field PAR in the Mode register (DBGU_MR). It then compares the result with the received parity bit. If different, the parity error bit PARE in DBGU_SR is set at the same time RXRDY is set. The parity bit is cleared when DBGU_CR is written with the bit RSTSTA (Reset Status) at 1. If a new character is received before the reset status command is written, the PARE bit remains at 1.

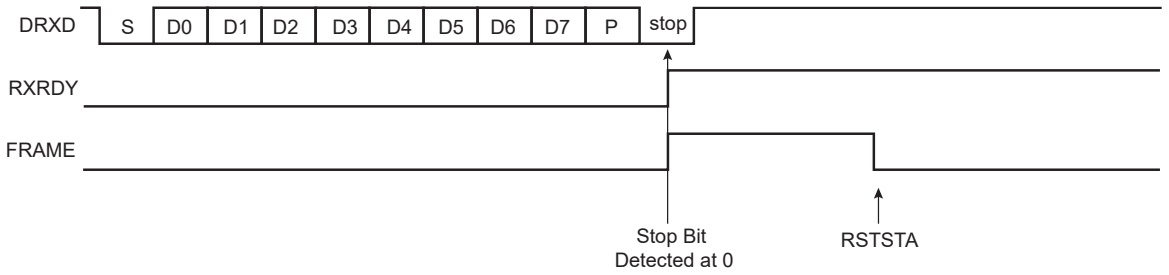
Figure 37.7. Parity Error



37.5.2.6. Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, the FRAME (Framing Error) bit in DBGU_SR is set at the same time the RXRDY bit is set. The FRAME bit remains high until the Control register (DBGU_CR) is written with the bit RSTSTA at 1.

Figure 37.8. Receiver Framing Error



37.5.2.7. Receiver Digital Filter

The DBGU embeds a digital filter on the receive line. It is disabled by default and can be enabled by writing a logical 1 in the FILTER bit of DBGU_MR. When enabled, the receive line is sampled using the 16x bit clock and a three-sample filter (majority 2 over 3) determines the value of the line.

37.5.2.8. Receiver Timeout

The Receiver Timeout provides support in handling variable-length frames. This feature detects an idle condition on the DRXD line. When a timeout is detected, the bit TIMEOUT in DBGU_SR 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 (DBGU_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The TIMEOUT bit remains at 0. Otherwise, the receiver loads an 8-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 TIMEOUT bit rises. Then, the user can either:

- stop the counter clock until a new character is received. This is performed by writing a one to the STTTO (Start timeout) bit in DBGU_CR. In this case, the idle state on DRXD before a new character is received does not provide a timeout. This prevents having to handle an interrupt before a character is received and allows waiting for the next idle state on DRXD after a frame is received, or

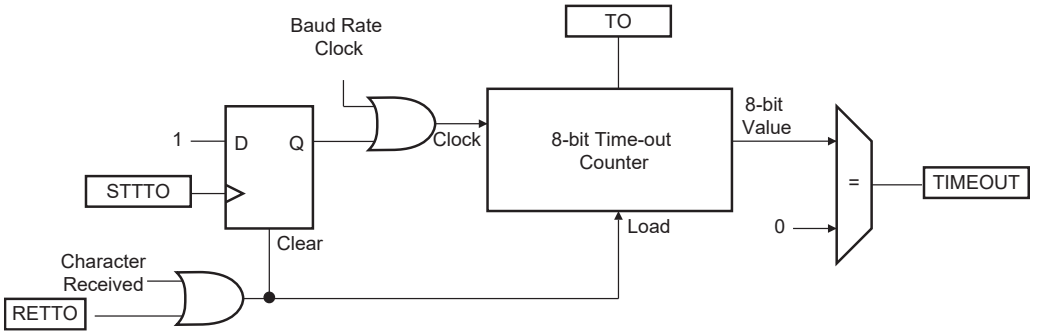
- obtain an interrupt while no character is received. This is performed by writing a one to the RETTO (Reload and start timeout) bit in DBGU_CR. If RETTO is performed, the counter starts counting down immediately from the TO value. This enables generation of a periodic interrupt so that a user timeout can be handled, for example when no key is pressed on a keyboard.

If STTTO is performed, the counter clock is stopped until a first character is received. The idle state on DRXD before the start of the frame does not provide a timeout. This prevents having to obtain a periodic interrupt and enables a wait of the end of frame when the idle state on DRXD is detected.

If RETTO is performed, the counter starts counting down immediately from the TO value. This enables generation of 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 37.9. Receiver Timeout Block Diagram



The following table gives the maximum timeout period for some standard baud rates.

Table 37.2. Maximum Timeout Period

Baud Rate (bit/s)	Bit Time (µs)	Timeout (µs)
600	1,667	425,085
1,200	833	212,415
2,400	417	106,335
4,800	208	53,040
9,600	104	26,520
14,400	69	17,595
19,200	52	13,260
28,800	35	8,925
38,400	26	6,630
56,000	18	4,590
57,600	17	4,335
200,000	5	1,275

37.5.3. Transmitter

37.5.3.1. Transmitter Reset, Enable and Disable

After device reset, the DBGU transmitter is disabled and must be enabled before being used. The transmitter is enabled by writing DBGU_CR with the bit TXEN at 1. From this command, the transmitter waits for a character to be written in the Transmit Holding register (DBGU_THR) before actually starting the transmission.

The programmer can disable the transmitter by writing DBGU_CR with the bit TXDIS at 1. If the transmitter is not operating, it is immediately stopped. However, if a character is being processed

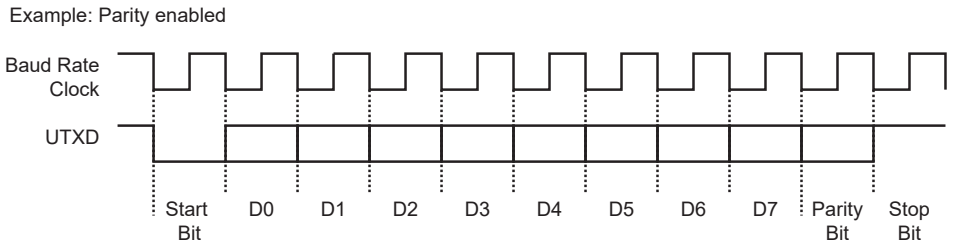
into the internal shift register and/or a character has been written in the DBGU_THR, the characters are completed before the transmitter is actually stopped.

The programmer can also put the transmitter in its reset state by writing the DBGU_CR with the bit RSTTX at 1. This immediately stops the transmitter, whether or not it is processing characters.

37.5.3.2. Transmit Format

The DBGU transmitter drives the pin DTXD at the baud rate clock speed. The line is driven depending on the format defined in DBGU_MR and the data stored in the internal shift register. One start bit at level 0, then the 8 data bits, from the lowest to the highest bit, one optional parity bit and one stop bit at 1 are consecutively shifted out as shown in the following figure. The field PARE in DBGU_MR defines whether or not a parity bit is shifted out. When a parity bit is enabled, it can be selected between an odd parity, an even parity, or a fixed space or mark bit.

Figure 37.10. Character Transmission

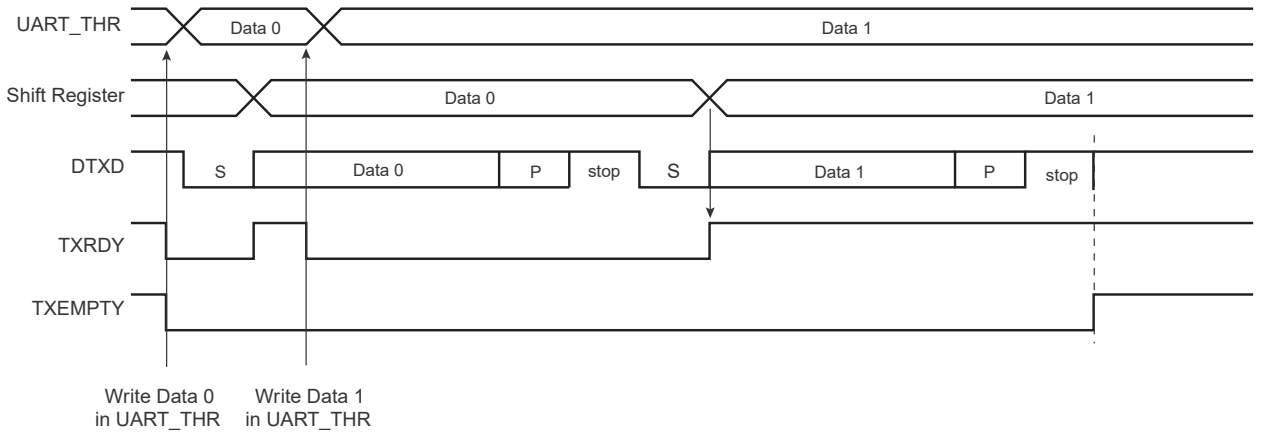


37.5.3.3. Transmitter Control

When the transmitter is enabled, the bit TXRDY (Transmitter Ready) is set in DBGU_SR. The transmission starts when the programmer writes in the DBGU_THR, and after the written character is transferred from DBGU_THR to the internal shift register. The TXRDY bit remains high until a second character is written in DBGU_THR. As soon as the first character is completed, the last character written in DBGU_THR is transferred into the internal shift register and TXRDY rises again, showing that the holding register is empty.

When both the internal shift register and DBGU_THR are empty, i.e., all the characters written in DBGU_THR have been processed, the TXEMPTY bit rises after the last stop bit has been completed.

Figure 37.11. Transmitter Control



37.5.4. DMA Support

Both the receiver and the transmitter of the DBGU are connected to a DMA Controller (DMAC) channel.

The DMA Controller channels are programmed via registers that are mapped within the DMAC user interface.

37.5.5. Register Write Protection

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

The following registers can be write-protected:

- DBGU Mode Register
- DBGU Baud Rate Generator Register
- DBGU Receiver Timeout Register
- Debug Unit Force NTRST Register

37.5.6. Test Modes

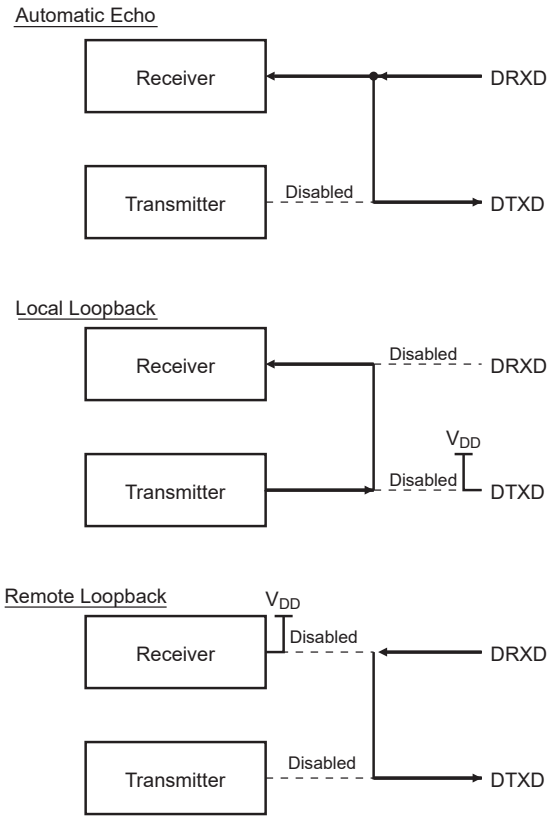
The DBGU supports three test modes. These modes of operation are programmed by using the CHMODE field in DBGU_MR.

The Automatic Echo mode allows a bit-by-bit retransmission. When a bit is received on the DRXD line, it is sent to the DTXD line. The transmitter operates normally, but has no effect on the DTXD line.

The Local loopback mode allows the transmitted characters to be received. DTXD and DRXD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The DRXD pin level has no effect and the DTXD line is held high, as in idle state.

The Remote loopback mode directly connects the DRXD pin to the DTXD line. The transmitter and the receiver are disabled and have no effect. This mode allows a bit-by-bit retransmission.

Figure 37.12. Test Modes



37.5.7. Debug Communication Channel Support

The DBGU handles the COMMRX and COMMTX signals that come from the Debug Communication Channel of the Arm processor and are driven by the In-circuit Emulator.

The Debug Communication Channel contains two registers that are accessible through the ICE Breaker on the JTAG side and through the coprocessor 0 on the Arm processor side.

As a reminder, the following instructions are used to read and write the Debug Communication Channel:

```
MRC p14, 0, Rd, c1, c0, 0
```

Returns the debug communication data read register into Rd.

```
MCR p14, 0, Rd, c1, c0, 0
```

Writes the value in Rd to the debug communication data write register.

The COMMRX and COMMTX bits which indicate, respectively, that the read register has been written by the debugger but not yet read by the processor, and that the write register has been written by the processor and not yet read by the debugger, are wired on the two highest bits of DBGU_SR. These bits can generate an interrupt. This feature can be used to handle under interrupt a debug link between a debug monitor running on the target system and a debugger.

37.5.8. Chip Identifier

The DBGU features two chip identifier registers, DBGU Chip ID register (DBGU_CIDR) and DBGU Extension ID register (DBGU_EXID). Both registers contain a hard-wired value that is read-only.

Table 37.3. DBGU_CIDR and DBGU_EXID Reset Values

Device Name	DBGU_CIDR	DBGU_EXID
SAM9X70	0x8975003X	0x00000080
		0x000000C4 (SL1)
		0x000000C5 (SL2)
		0x000000C6 (SL3)
SAM9X72		0x00000082
		0x000000D8 (SL1)
		0x000000D9 (SL2)
		0x000000DA (SL3)
SAM9X75		0x00000085
		0x000000DC (SL1)
		0x000000DD (SL2)
		0x000000DE (SL3)

37.5.9. ICE Access Prevention

The DBGU allows blockage of access to the system through the Arm processor's ICE interface. This feature is implemented via the Force NTRST register (DBGU_FNR), that allows assertion of the NTRST signal of the ICE interface. Writing the bit FNTRST (Force NTRST) to 1 in this register prevents any activity on the TAP controller.

Notes:

1. During ROM code execution, this bit is set (JTAG access is disabled).
2. This bit has no more effect once the JTAG access is disabled in the OTPC.

37.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DBGU_CR	31:24								
		23:16								
		15:8					STTTO	RETTO		RSTSTA
		7:0	TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX		
0x04	DBGU_MR	31:24								
		23:16								
		15:8	CHMODE[1:0]			BRSRCCK	PAR[2:0]			
		7:0				FILTER				
0x08	DBGU_IER	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x0C	DBGU_IDR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x10	DBGU_IMR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x14	DBGU_SR	31:24	COMMRX	COMMTX						
		23:16								
		15:8							TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE				TXRDY	RXRDY
0x18	DBGU_RHR	31:24								
		23:16								
		15:8								
		7:0	RXCHR[7:0]							
0x1C	DBGU_THR	31:24								
		23:16								
		15:8								
		7:0	TXCHR[7:0]							
0x20	DBGU_BRGR	31:24								
		23:16								
		15:8	CD[15:8]							
		7:0	CD[7:0]							
0x24 ... 0x27	Reserved									
0x28	DBGU_RTOR	31:24								
		23:16								
		15:8								
		7:0	TO[7:0]							
0x2C ... 0x3F	Reserved									
0x40	DBGU_CIDR	31:24	EXT	CHID[30:24]						
		23:16	CHID[23:16]							
		15:8	CHID[15:8]							
		7:0	CHID[7:0]							
0x44	DBGU_EXID	31:24	EXID[31:24]							
		23:16	EXID[23:16]							
		15:8	EXID[15:8]							
		7:0	EXID[7:0]							
0x48	DBGU_FNR	31:24								
		23:16								
		15:8								
		7:0								FNTRST

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x4C ... 0xE3	Reserved									
0xE4	DBGU_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

37.6.1. DBGU Control Register

Name: DBGU_CR
Offset: 0x0000
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
					STTTO	RETTO		RSTSTA
Access					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 11 – STTTO Start Timeout

Value	Description
0	No effect.
1	Starts waiting for a character before clocking the timeout counter. Resets status bit DBGU_SR.TIMEOUT.

Bit 10 – RETTO Rearm Timeout

Value	Description
0	No effect.
1	Restarts timeout.

Bit 8 – RSTSTA Reset Status

Value	Description
0	No effect.
1	Resets the status bits PARE, FRAME and OVRE in DBGU_SR.

Bit 7 – TXDIS Transmitter Disable

Value	Description
0	No effect.
1	The transmitter is disabled. If a character is being processed and a character has been written in DBGU_THR and RSTTX is not set, both characters are completed before the transmitter is stopped.

Bit 6 – TXEN Transmitter Enable

Value	Description
0	No effect.

Value	Description
1	The transmitter is enabled if TXDIS is 0.

Bit 5 – RXDIS Receiver Disable

Value	Description
0	No effect.
1	The receiver is disabled. If a character is being processed and RSTRX is not set, the character is completed before the receiver is stopped.

Bit 4 – RXEN Receiver Enable

Value	Description
0	No effect.
1	The receiver is enabled if RXDIS is 0.

Bit 3 – RSTTX Reset Transmitter

Value	Description
0	No effect.
1	The transmitter logic is reset and disabled. If a character is being transmitted, the transmission is aborted.

Bit 2 – RSTRX Reset Receiver

Value	Description
0	No effect.
1	The receiver logic is reset and disabled. If a character is being received, the reception is aborted.

37.6.2. DBGU Mode Register

Name: DBGU_MR
Offset: 0x0004
Reset: 0x0000000
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
	CHMODE[1:0]			BRSRCCK	PAR[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
				FILTER				
Access				R/W				
Reset				0				

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

Bit 12 – BRSRCCK Baud Rate Source Clock
0 (PERIPH_CLK): The baud rate is driven by the peripheral clock
1 (GCLK): The baud rate is driven by a PMC-programmable clock GCLK (see section Power Management Controller (PMC)).

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even Parity
1	ODD	Odd Parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

Bit 4 – FILTER Receiver Digital Filter
0 (DISABLED): DBGU does not filter the receive line.
1 (ENABLED): DBGU filters the receive line using a three-sample filter (16x-bit clock) (2 over 3 majority).

37.6.3. DBGU Interrupt Enable Register

Name: DBGU_IER
Offset: 0x0008
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
	COMMRX	COMMTX						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							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 – COMMRX Enable COMMRX (from Arm) Interrupt

Bit 30 – COMMTX Enable COMMTX (from Arm) Interrupt

Bit 9 – TXEMPTY Enable TXEMPTY Interrupt

Bit 8 – TIMEOUT Enable Timeout Interrupt

Bit 7 – PARE Enable Parity Error Interrupt

Bit 6 – FRAME Enable Framing Error Interrupt

Bit 5 – OVRE Enable Overrun Error Interrupt

Bit 1 – TXRDY Enable TXRDY Interrupt

Bit 0 – RXRDY Enable RXRDY Interrupt

37.6.4. DBGU Interrupt Disable Register

Name: DBGU_IDR
Offset: 0x000C
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
	COMMRX	COMMTX						
Access	W	W						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							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 – COMMRX Disable COMMRX (from Arm) Interrupt

Bit 30 – COMMTX Disable COMMTX (from Arm) Interrupt

Bit 9 – TXEMPTY Disable TXEMPTY Interrupt

Bit 8 – TIMEOUT Disable Timeout Interrupt

Bit 7 – PARE Disable Parity Error Interrupt

Bit 6 – FRAME Disable Framing Error Interrupt

Bit 5 – OVRE Disable Overrun Error Interrupt

Bit 1 – TXRDY Disable TXRDY Interrupt

Bit 0 – RXRDY Disable RXRDY Interrupt

37.6.5. DBGU Interrupt Mask Register

Name: DBGU_IMR
Offset: 0x0010
Reset: 0x0000000
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
	COMMRX	COMMTX						
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
							TXEMPTY	TIMEOUT
Access							R	R
Reset							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 – COMMRX Mask COMMRX (from Arm) Interrupt

Bit 30 – COMMTX Mask COMMTX (from Arm) Interrupt

Bit 9 – TXEMPTY Mask TXEMPTY Interrupt

Bit 8 – TIMEOUT Mask Timeout Interrupt

Bit 7 – PARE Mask Parity Error Interrupt

Bit 6 – FRAME Mask Framing Error Interrupt

Bit 5 – OVRE Mask Overrun Error Interrupt

Bit 1 – TXRDY Disable TXRDY Interrupt

Bit 0 – RXRDY Mask RXRDY Interrupt

37.6.6. DBGU Status Register

Name: DBGU_SR
Offset: 0x0014
Reset: –
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COMMRX	COMMTX						
Access	R	R						
Reset	–	–						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							TXEMPTY	TIMEOUT
Access							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 – COMMRX Debug Communication Channel Read Status

Value	Description
0	COMMRX from the Arm processor is inactive.
1	COMMRX from the Arm processor is active.

Bit 30 – COMMTX Debug Communication Channel Write Status

Value	Description
0	COMMTX from the Arm processor is inactive.
1	COMMTX from the Arm processor is active.

Bit 9 – TXEMPTY Transmitter Empty

Value	Description
0	There are characters in DBGU_THR, or characters are being processed by the transmitter, or the transmitter is disabled.
1	There are no characters in DBGU_THR and there are no characters being processed by the transmitter.

Bit 8 – TIMEOUT Receiver Timeout

Value	Description
0	There has not been any timeout since the last Start timeout command (DBGU_CR.STTTO), or the Timeout register is 0.
1	There has been a timeout since the last Start timeout command (DBGU_CR.STTTO).

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has occurred since the last RSTSTA.
1	At least one parity error has occurred since the last RSTSTA.

Bit 6 – FRAME Framing Error

Value	Description
0	No framing error has occurred since the last RSTSTA.
1	At least one framing error has occurred since the last RSTSTA.

Bit 5 – OVRE Overrun Error

Value	Description
0	No overrun error has occurred since the last RSTSTA.
1	At least one overrun error has occurred since the last RSTSTA.

Bit 1 – TXRDY Transmitter Ready

Value	Description
0	A character has been written to DBGU_THR and not yet transferred to the internal shift register, or the transmitter is disabled.
1	There is no character written to DBGU_THR that is not yet transferred to the internal shift register.

Bit 0 – RXRDY Receiver Ready

Value	Description
0	No character has been received since the last read of DBGU_RHR, or the receiver is disabled.
1	At least one complete character has been received, transferred to DBGU_RHR, and not yet read.

37.6.7. DBGU Receiver Holding Register

Name: DBGU_RHR
Offset: 0x0018
Reset: 0x0000000
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
	RXCHR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXCHR[7:0] Received Character
Last received character if RXRDY is set.

37.6.8. DBGU Transmit Holding Register

Name: DBGU_THR
Offset: 0x001C
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
	TXCHR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 7:0 – TXCHR[7:0] Character to be Transmitted
Next character to be transmitted after the current character if TXRDY is not set.

37.6.9. DBGU Baud Rate Generator Register

Name: DBGU_BRGR
Offset: 0x0020
Reset: 0x0000000
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
	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 15:0 – CD[15:0] Clock Divisor

Value	Description
0	Baud rate clock is disabled.
1	1 to 65,535: If BRSRCCK = 0: $CD = \frac{f_{peripheral\ clock}}{16 \times Baud\ Rate}$ If BRSRCCK = 1: $CD = \frac{f_{GCLKx}}{16 \times Baud\ Rate}$

37.6.10. DBGU Receiver Timeout Register

Name: DBGU_RTOR
Offset: 0x0028
Reset: 0x0000000
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
	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 7:0 – TO[7:0] Timeout Value

Value	Description
0	The receiver timeout is disabled.
1–255	The receiver timeout is enabled and the timeout delay is TO x bit period.

37.6.11. DBGU Chip ID Register

Name: DBGU_CIDR
Offset: 0x0040
Reset: –
Property: Read-only

For detailed reset values, refer to Chip Identifier.

Bit	31	30	29	28	27	26	25	24
	EXT	CHID[30:24]						
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	CHID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CHID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

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

Bit 31 – EXT Extension Flag

Value	Description
0	Chip ID has a single register definition without extension.
1	An extended Chip ID exists.

Bits 30:0 – CHID[30:0] Chip ID Value

37.6.12. DBGU Chip ID Extension Register

Name: DBGU_EXID
Offset: 0x0044
Reset: –
Property: Read-only

For detailed reset values, refer to Chip Identifier.

Bit	31	30	29	28	27	26	25	24
	EXID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	EXID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

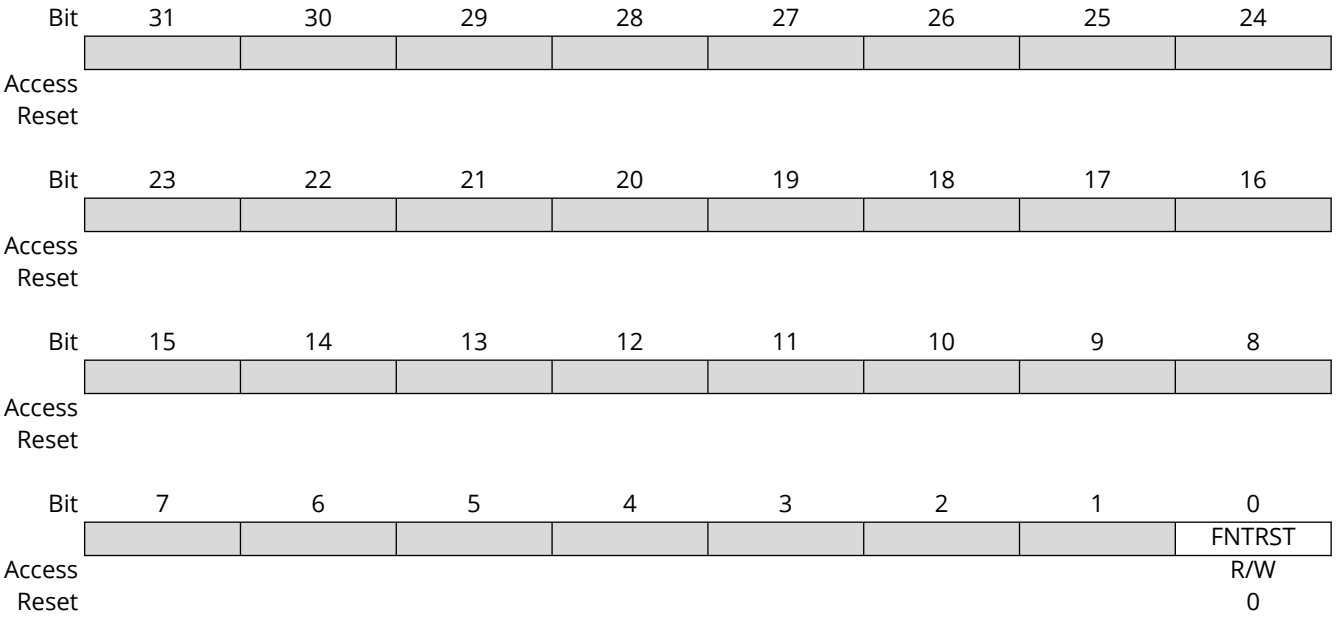
Bit	15	14	13	12	11	10	9	8
	EXID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

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

Bits 31:0 – EXID[31:0] Chip ID Extension
Read as 0 if DBGU_CIDR.EXT=0.

37.6.13. Debug Unit Force NTRST Register

Name: DBGU_FNR
Offset: 0x0048
Reset: 0x00000000
Property: Read/Write



Bit 0 – FNTRST Force NTRST

Value	Description
0	NTRST of the Arm processor's TAP controller is driven by the power_on_reset signal.
1	NTRST of the Arm processor's TAP controller is held low.

37.6.14. DBGU Write Protection Mode Register

Name: DBGU_WPMR
Offset: 0x00E4
Reset: 0x0000000
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
0x554152	PASSWD	Writing any other value in this field aborts the write operation. 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 0x554152 (DBGU in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x554152 (DBGU in ASCII).

38. Special Function Registers (SFR)

38.1. Description

Special Function Registers (SFR) manage specific aspects of the integrated memory, bridge implementations, processor and other functionality not controlled elsewhere.

38.2. Embedded Characteristics

- 32-bit Special Function Registers Control Specific Behavior of the Product

38.3. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	SFR_CCFG_EBICSA	31:24							DDR_MP_EN	NFD0_ON_D16
		23:16				DQIEN_F				
		15:8							EBI_DBPDC	EBI_DBPUC
		7:0						EBI_CS2A	EBI_CS1A	
0x08 ... 0x0F	Reserved									
0x10	SFR_OHCIICR	31:24								
		23:16	UDPPUDIS							
		15:8						SUSP2	SUSP1	SUSP0
		7:0			APPSTART	ARIE		RES2	RES1	RES0
0x14	SFR_OHCIISR	31:24								
		23:16								
		15:8								
		7:0						RIS2	RIS1	RIS0
0x18 ... 0x33	Reserved									
0x34	SFR_UTMIHSTRIM	31:24								
		23:16							SLOPE2[2:0]	
		15:8		SLOPE1[2:0]					SLOPE0[2:0]	
		7:0								
0x38	SFR_UTMIFSTRIM	31:24		ZP_CAL[2:0]					ZN_CAL[2:0]	
		23:16		ZP[2:0]					ZN[2:0]	
		15:8								
		7:0								
0x3C	SFR_UTMISWAP	31:24								
		23:16								
		15:8								
		7:0						PORT2	PORT1	PORT0
0x40 ... 0x9F	Reserved									
0xA0	SFR_LS	31:24								
		23:16								MEM_POWER_GATING_ULP1_EN
		15:8			LS13	LS12	LS11	LS10	LS9	LS8
		7:0	LS7	LS6	LS5	LS4	LS3	LS2	LS1	LS0
0xA4 ... 0xB3	Reserved									
0xB4	SFR_CAL1	31:24								
		23:16								
		15:8								TEST_M
		7:0		CALP_M[3:0]					CALN_M[3:0]	
0xB8 ... 0xE3	Reserved									
0xE4	SFR_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8 ... 0x01FF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0200	SFR_PUFCTL	31:24								
		23:16								
		15:8							PHG	PLG
		7:0		ALWAYSONE	PUFLTM	PUFDIS	PUFRST	PONOFFLZ	PONOFFHZ	PONOFFM
0x0204 ... 0x0207	Reserved									
0x0208	SFR_PUFDIS	31:24								
		23:16								
		15:8				RESEED	LAB_TEST_MODE	TEST_MEMORY	RECONSTRUCT	WRAP_GENERATED_RANDOM
		7:0	WRAP	UNWRAP	TEST_PUF	STOP	START	GET_KEY	GENERATE_RANDOM	ENROLL
0x020C	SFR_PUFRUCR0	31:24	RESTRICT_USER_CONTEXT_0[31:24]							
		23:16	RESTRICT_USER_CONTEXT_0[23:16]							
		15:8	RESTRICT_USER_CONTEXT_0[15:8]							
		7:0	RESTRICT_USER_CONTEXT_0[7:0]							
0x0210	SFR_PUFRUCR1	31:24	RESTRICT_USER_CONTEXT_1[31:24]							
		23:16	RESTRICT_USER_CONTEXT_1[23:16]							
		15:8	RESTRICT_USER_CONTEXT_1[15:8]							
		7:0	RESTRICT_USER_CONTEXT_1[7:0]							
0x0214	SFR_PUFWORUCR0	31:24	RESTRICT_USER_CONTEXT_0_WO[31:24]							
		23:16	RESTRICT_USER_CONTEXT_0_WO[23:16]							
		15:8	RESTRICT_USER_CONTEXT_0_WO[15:8]							
		7:0	RESTRICT_USER_CONTEXT_0_WO[7:0]							
0x0218	SFR_PUFWORUCR1	31:24	RESTRICT_USER_CONTEXT_1_WO[31:24]							
		23:16	RESTRICT_USER_CONTEXT_1_WO[23:16]							
		15:8	RESTRICT_USER_CONTEXT_1_WO[15:8]							
		7:0	RESTRICT_USER_CONTEXT_1_WO[7:0]							
0x021C ... 0x021F	Reserved									
0x0220	SFR_FLEXRAMS_CLKG_DIS	31:24								
		23:16								
		15:8								
		7:0							FLEX1_CLKG_DIS	FLEX0_CLKG_DIS
0x0224 ... 0x023F	Reserved									
0x0240	SFR_ISS_CFG	31:24								
		23:16								
		15:8								
		7:0								MODE
0x0244 ... 0x024F	Reserved									
0x0250	SFR_TSU_CFG	31:24								
		23:16								
		15:8								
		7:0	WIDTH[7:0]							
0x0254 ... 0x025F	Reserved									
0x0260	SFR_REMAP_MP_DDR	31:24								
		23:16								
		15:8			Host_13	Host_12	Host_11	Host_10	Host_9	Host_8
		7:0	Host_7	Host_6	Host_5	Host_4	Host_3	Host_2	Host_1	Host_0

38.3.1. EBI Chip Select Register

Name: SFR_CCFG_EBICSA
Offset: 0x04
Reset: 0x00000300
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
							DDR_MP_EN	NFD0_ON_D16
Access							R/W	R/W
Reset							0	0

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

Bit	15	14	13	12	11	10	9	8
							EBI_DBPDC	EBI_DBPUC
Access							R/W	R/W
Reset							1	1

Bit	7	6	5	4	3	2	1	0
						EBI_CS2A	EBI_CS1A	
Access						R/W	R/W	
Reset						0	0	

Bit 25 – DDR_MP_EN DDR Multiport Enable

Value	Description
0	DDR Multiport is disabled (default).
1	DDR Multiport is enabled, performance is increased.

Bit 24 – NFD0_ON_D16 NAND Flash Databus Selection

Value	Description
0	NAND Flash I/Os are connected to D0–D7 (default).
1	NAND Flash I/Os are connected to D16–D23.

Bit 20 – DQIEN_F Force Analog Input Comparator Configuration

Value	Description
0	No effect
1	Enables the input comparator in the VDDIOM I/O data lines. Must be set to one in an initialization phase whenever an MPDDRC external component (DDR2 or LPDDR) and an SMC external component (e.g., NAND Flash) are multiplexed on the D0-D15 bus.

Bit 9 – EBI_DBPDC EBI Data Bus Pulldown Configuration

Value	Description
0	EBI D0–D15 Data Bus bits are not internally pulled down.
1	EBI D0–D15 Data Bus bits are internally pulled down to the ground.

Bit 8 – EBI_DBPUC EBI Data Bus Pullup Configuration

Value	Description
0	EBI D0–D15 Data Bus bits are internally pulled up to the VDDIOM power supply.
1	EBI D0–D15 Data Bus bits are not internally pulled up.

Bit 2 – EBI_CS2A EBI Chip Select 2 Assignment

Value	Description
0	EBI Chip Select 2 is only assigned to the SMC and EBI_NCS2 behaves as defined by the SMC.
1	EBI Chip Select 2 is assigned to the SMC and the NAND Flash Logic is activated.

Bit 1 – EBI_CS1A EBI Chip Select 1 Assignment

Value	Description
0	EBI Chip Select 1 is assigned to the Static Memory Controller (SMC).
1	EBI Chip Select 1 is assigned to the AHB Multiport DDR-SDRAM Controller (MPDDRC).

38.3.2. OHCI Interrupt Configuration Register

Name: SFR_OHCIICR
Offset: 0x10
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
	UDPPUDIS							
Access	R/W							
Reset	0							

Bit	15	14	13	12	11	10	9	8
						SUSP2	SUSP1	SUSP0
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
			APPSTART	ARIE		RES2	RES1	RES0
Access			R/W	R/W		R/W	R/W	R/W
Reset			0	0		0	0	0

Bit 23 – UDPPUDIS Reserved

Value	Description
0	Must write 0.

Bits 8, 9, 10 – SUSPx USB PORTx

Value	Description
0	Does not suspend USB PORTx.
1	Forces PORTx suspend.

Bit 5 – APPSTART Reserved

Value	Description
0	Must write 0.

Bit 4 – ARIE OHCI Asynchronous Resume Interrupt Enable

Value	Description
0	Disables interrupt.
1	Enables interrupt.

Bits 0, 1, 2 – RESx USB PORTx Reset

Value	Description
0	No effect (USB PORTx reset released, default value)
1	Resets USB PORTx.

38.3.3. OHCI Interrupt Status Register

Name: SFR_OHCIISR
Offset: 0x14
Reset: 0x00000000
Property: Read-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
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 is not detected.
1	OHCI port resume is detected.

38.3.4. UTMI High-Speed Trimming Register

Name: SFR_UTMIHSTRIM
Offset: 0x34
Reset: 0x00044433
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
						SLOPE2[2:0]		
Access						R/W	R/W	R/W
Reset						1	0	0

Bit	15	14	13	12	11	10	9	8
		SLOPE1[2:0]				SLOPE0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	0	0		1	0	0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 8:10, 12:14, 16:18 – SLOPEx UTMI HS PORTx Transceiver Slope Trimming
Adjusts HS transceiver output slope for PORTx.
These bits are duplicated for each port because the HS slope depends on the PCB line length. Short lines appear as lumped capacitances and exhibit a slower rise/fall time, while longer lines appear as transmission lines with sharper edges.

Value	Name
0	Faster
1	–
2	–
3	–
4	Default
5	–
6	–
7	Slower

38.3.5. UTMI Full-Speed Trimming Register

Name: SFR_UTMIFSTRIM
Offset: 0x38
Reset: 0x00430211
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
		ZP_CAL[2:0]				ZN_CAL[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
		ZP[2:0]				ZN[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		1	0	0		0	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 30:28 – ZP_CAL[2:0] FS Transceiver PMOS Impedance Calibration
Adjusts the FS transceiver PMOS output impedance calibration.

Value	Name
0	Default
1	–
2	–
3	Higher
4	Lower
5	–
6	–
7	–

Bits 26:24 – ZN_CAL[2:0] FS Transceiver NMOS Impedance Calibration
Adjusts the FS transceiver NMOS output impedance calibration.

Value	Name
0	Default
1	–
2	–
3	Higher
4	Lower
5	–
6	–
7	–

Bits 22:20 – ZP[2:0] FS Transceiver PMOS Impedance Trimming
Adjusts the FS transceiver PMOS output impedance.

Value	Name
0	Lower
1	–
2	–
3	–
4	Default
5	–
6	–
7	Higher

Bits 18:16 – ZN[2:0] FS Transceiver NMOS Impedance Trimming
Adjusts the FS transceiver NMOS output impedance.

Value	Name
0	Higher
1	–
2	–
3	–
4	Default
5	–
6	–
7	Lower

38.3.6. UTMI DP/DM Pin Swapping Register

Name: SFR_UTMISWAP
Offset: 0x3C
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						PORT2	PORT1	PORT0
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 0, 1, 2 – PORTx Port x DP/DM Pin Swapping

Value	Name	Description
0	NORMAL	DP/DM normal pinout
1	SWAPPED	DP/DM swapped pinout

38.3.7. SFR Light Sleep Register

Name: SFR_LS
Offset: 0xA0
Reset: 0x00000000
Property: Read/Write

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

The following configuration values are valid for all listed LSx bit names of this register:

0: Disables Light Sleep mode.

1: Enables Light Sleep mode.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								MEM_POWER_GATING_ULP1_EN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
			LS13	LS12	LS11	LS10	LS9	LS8
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
	LS7	LS6	LS5	LS4	LS3	LS2	LS1	LS0
Access	R/W	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 – MEM_POWER_GATING_ULP1_EN Light Sleep Value for ULP1 Power-Gated Memories
The memory power gating can be automatically enabled when entering ULP1 Low-power mode. Refer to the section “Electrical Characteristics”.

Value	Description
0	Light Sleep mode is not activated by the MEM_POWER_GATING_ULP1 output signal from the PMC.
1	Light Sleep mode is activated when the MEM_POWER_GATING_ULP1 output signal from the PMC is activated.

Bit 13 – LS13 Light Sleep Value (GMAC)

Bit 12 – LS12 Light Sleep Value (DSI)

Bit 11 – LS11 Light Sleep Value (ISC)

Bit 10 – LS10 Light Sleep Value (CSI2DC)

Bit 9 – LS9 Light Sleep Value (ARM926)

Bit 8 – LS8 Light Sleep Value (ROM + OTPC)

Bit 7 – LS7 Light Sleep Value (FLEXRAM1 (OTPC))

Bit 6 – LS6 Light Sleep Value (FLEXRAM0)

Bit 5 – LS5 Light Sleep Value (UHPHS)

Bit 4 – LS4 Light Sleep Value (XDMAC)

Bit 3 – LS3 Light Sleep Value (UDPHS)

Bit 2 – LS2 Light Sleep Value (SDMMC)

Bit 1 – LS1 Light Sleep Value (LCDC)

Bit 0 – LS0 Light Sleep Value (GFX2D)

38.3.8. SFR I/O Calibration 1 Register

Name: SFR_CAL1
Offset: 0xB4
Reset: 0x00000084
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								R/W
Reset								0

Bit	7	6	5	4	3	2	1	0
	CALP_M[3:0]				CALN_M[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 8 – TEST_M Enable Calibration of Low/High Level Output Impedance of Pads with VDDIOM Supply
Enables calibration bits to adjust the low/high level output impedance of pads with power supply input VDDIOM.

Bits 7:4 – CALP_M[3:0] Calibration of High Level Output Impedance of Pads with VDDIOM Supply
Adjusts the high level output impedance of pads with power supply input VDDIOM.

Value	Name
0000	Minimum output impedance value
1111	Maximum output impedance value

Bits 3:0 – CALN_M[3:0] Calibration of Low Level Output Impedance of Pads with VDDIOM Supply
Adjusts the low level output impedance of pads with power supply input VDDIOM.

Value	Name
0000	Maximum output impedance value
1111	Minimum output impedance value

38.3.9. SFR Write Protection Mode Register

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

Note: All registers are write-protected.

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
								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

Value	Description
0	Disables write protection if WPKEY corresponds to 0x534652 (“SFR” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x534652 (“SFR” in ASCII).

38.3.10. PUFSRAM Domain Control Register

Name: SFR_PUFCTL
Offset: 0x200
Reset: 0x00000148
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
							PHG	PLG
Access							R	R
Reset							0	1

Bit	7	6	5	4	3	2	1	0
		ALWAYSONE	PUFLTM	PUFDIS	PUFRST	PONOFFLZ	PONOFFHZ	PONOFFM
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	0	0	0

Bit 9 – PHG PUFSRAM Power High Status
CAUTION: PHG is only 1 when the two power switches (HiZ and LoZ) are on. In other case PHG=0 and PLG=1.

Value	Description
0	PUFSRAM domain is not powered.
1	PUFSRAM domain is powered.

Bit 8 – PLG PUFSRAM Power Low Status

Value	Description
0	PUFSRAM domain is powered.
1	PUFSRAM domain is not powered.

Bit 6 – ALWAYSONE Must always be programmed to 1

Bit 5 – PUFLTM PUF Lab Test Mode

Value	Description
0	No effect
1	Enters Test mode when PUFRST=0.

Bit 4 – PUFDIS Disable the PUF
CAUTION: This bit is write-once until the next chip reset.

Value	Description
0	No effect

Value	Description
1	PUF clock and psel signals are disabled by HW and PUFSRAM domain is off.

Bit 3 – PUFRST Reset for PUF IP

Value	Description
0	PUF is active.
1	PUF is in Reset mode.

Bit 2 – PONOFFLZ Controls Power Switch Low Z if PONOFFM=1

Value	Description
0	Power switch Low Z is off.
1	Power switch Low Z is on.

Bit 1 – PONOFFHZ Controls Power Switch High Z if PONOFFM=1

Value	Description
0	Power switch High Z is off.
1	Power switch High Z is on.

Bit 0 – PONOFFM PUFSRAM Power Switches Controlled Manually

CAUTION: When controlling power switches manually, apply the following sequence:

1. Write PONOFFHZ to 1.
2. Wait the time defined in the fuses (from 20 µs to 80 µs).
3. Write PONOFFLZ to 1.

Value	Description
0	Power switches are automatically controlled by SYSC at startup.
1	Power switches are manually controlled by the PONOFFLZ and PONOFFHZ bits.

38.3.11. PUF Disable Functions Register

Name: SFR_PUFDIS
Offset: 0x208
Reset: 0x00000000
Property: Read/Write

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

CAUTION: This register only allows to write 'ones' and forbids to write 'zeros'. A chip reset is needed to clear this register.

The following configuration values are valid for all listed bit names of this register:

0: Operation is permitted.

1: Operation is forbidden.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				RESEED	LAB_TEST_MODE	TEST_MEMORY	RECONSTRUCT	WRAP_GENERATED_RANDOM
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
	WRAP	UNWRAP	TEST_PUF	STOP	START	GET_KEY	GENERATE_RANDOM	ENROLL
Access	R/W	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 – RESEED Connected to qk_disable_reseed input for QK

Bit 11 – LAB_TEST_MODE Connected to qk_disable_lab_test_mode input for QK

Bit 10 – TEST_MEMORY Connected to qk_disable_test_memory input for QK

Bit 9 – RECONSTRUCT Connected to qk_disable_reconstruct input for QK

Bit 8 – WRAP_GENERATED_RANDOM Connected to qk_disable_wrap_generated_random input for QK

Bit 7 – WRAP Connected to qk_disable_wrap input for QK

Bit 6 – UNWRAP Connected to qk_disable_unwrap input for QK

Bit 5 – TEST_PUF Connected to qk_disable_test_puf input for QK

Bit 4 – STOP Connected to qk_disable_stop input for QK

Bit 3 – START Connected to qk_disable_start input for QK

Bit 2 – GET_KEY Connected to qk_disable_get_key input for QK

Bit 1 – GENERATE_RANDOM Connected to qk_disable_generate_random input for QK

Bit 0 – ENROLL Connected to qk_disable_enroll input for QK

38.3.12. PUF Restrict User Context 0 Register

Name: SFR_PUFRUCR0
Offset: 0x20C
Reset: 0x00000000
Property: Read/Write

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

CAUTION SFR_PUFRUCRx and SFR_PUFWORUCRx registers are connected to the same input by an OR gate.

Bit	31	30	29	28	27	26	25	24
	RESTRICT_USER_CONTEXT_0[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
	RESTRICT_USER_CONTEXT_0[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
	RESTRICT_USER_CONTEXT_0[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
	RESTRICT_USER_CONTEXT_0[7:0]							
Access	R/W	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 – RESTRICT_USER_CONTEXT_0[31:0] Value Connected to qk_restrict_user_context_0 input for QK

38.3.13. PUF Restrict User Context 1 Register

Name: SFR_PUFRUCR1
Offset: 0x210
Reset: 0x00000000
Property: Read/Write

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

CAUTION SFR_PUFRUCRx and SFR_PUFWORUCRx registers are connected to the same input by an OR gate.

Bit	31	30	29	28	27	26	25	24
	RESTRICT_USER_CONTEXT_1[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
	RESTRICT_USER_CONTEXT_1[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
	RESTRICT_USER_CONTEXT_1[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
	RESTRICT_USER_CONTEXT_1[7:0]							
Access	R/W	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 – RESTRICT_USER_CONTEXT_1[31:0] Value Connected to qk_restrict_user_context_1 input for QK

38.3.14. PUF Restrict User Context 0 Write Ones Register

Name: SFR_PUFWORUCR0
Offset: 0x214
Reset: 0x00000000
Property: Read/Write

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

CAUTION

1. This register only allows to write "ones". "Zeros" are forbidden. A chip reset is needed to clear this register.
2. SFR_PUFRUCRx and SFR_PUFWORUCRx registers are connected to the same input by an OR gate.

Bit	31	30	29	28	27	26	25	24
	RESTRICT_USER_CONTEXT_0_WO[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
	RESTRICT_USER_CONTEXT_0_WO[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
	RESTRICT_USER_CONTEXT_0_WO[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
	RESTRICT_USER_CONTEXT_0_WO[7:0]							
Access	R/W	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 – RESTRICT_USER_CONTEXT_0_WO[31:0] Value Connected to qk_restrict_user_context_0 input for QK

38.3.15. PUF Restrict User Context 1 Write Ones Register

Name: SFR_PUFWORUCR1
Offset: 0x218
Reset: 0x00000000
Property: Read/Write

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

CAUTION

1. This register only allows to write "ones". "Zeros" are forbidden. A chip reset is needed to clear this register.
2. SFR_PUFRUCRx and SFR_PUFWORUCRx registers are connected to the same input by an OR gate.

Bit	31	30	29	28	27	26	25	24
	RESTRICT_USER_CONTEXT_1_WO[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
	RESTRICT_USER_CONTEXT_1_WO[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
	RESTRICT_USER_CONTEXT_1_WO[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
	RESTRICT_USER_CONTEXT_1_WO[7:0]							
Access	R/W	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 – RESTRICT_USER_CONTEXT_1_WO[31:0] Value Connected to qk_restrict_user_context_1 input for QK

38.3.16. FLEXRAMS Clock Gating Disable Register

Name: SFR_FLEXRAMS_CLKG_DIS
Offset: 0x220
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
							FLEX1_CLKG_DIS	FLEX0_CLKG_DIS
Access							R/W	R/W
Reset							0	0

Bit 1 – FLEX1_CLKG_DIS Clock Gating Disable for FLEXRAM1

Value	Description
0	Clock Gating is enabled.
1	Clock Gating is disabled.

Bit 0 – FLEX0_CLKG_DIS Clock Gating Disable for FLEXRAM0

Value	Description
0	Clock Gating is enabled.
1	Clock Gating is disabled.

38.3.17. ISS Configuration Register

Name: SFR_ISS_CFG
Offset: 0x240
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
Access								
Reset								

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

Bit 0 – MODE DSI/CSI Selection

Value	Name	Description
0	CSI	CSI mode
1	DSI	DSI mode

38.3.18. TSU Configuration Register

Name: SFR_TSU_CFG
Offset: 0x250
Reset: 0x00000043
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
	WIDTH[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	1	1

Bits 7:0 – WIDTH[7:0] Number of TSU Cycles to Increase GTSUCOMP Width
GTSUCOMP_cycles = WIDTH +1

38.3.19. Remap Multiport DDR Register

Name: SFR_REMAP_MP_DDR
Offset: 0x260
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
			Host_13	Host_12	Host_11	Host_10	Host_9	Host_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
	Host_7	Host_6	Host_5	Host_4	Host_3	Host_2	Host_1	Host_0
Access	R/W	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 – Host_x Enable DDR Multiport Only for Host_x
Use instead of DDR_MP_EN in SFR_CCFG_EBICSA.

39. Slow Clock Controller (SCKC)

39.1. 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:

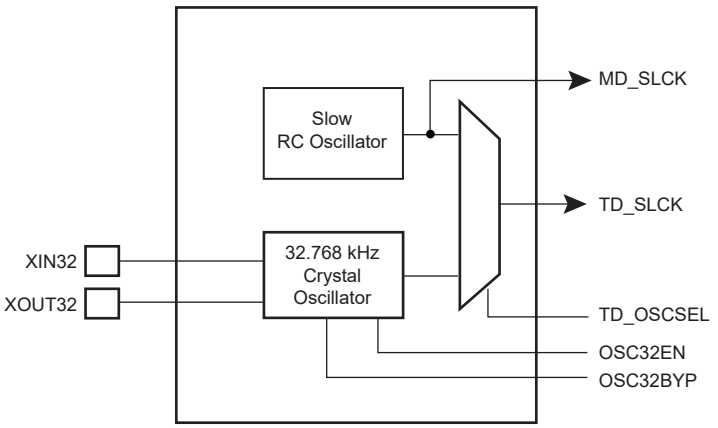
- External 32.768 kHz crystal oscillator
- Embedded 32 kHz (typical) slow RC oscillator

39.2. Embedded Characteristics

- Embedded 32 kHz (Typical) Slow RC Oscillator or 32.768 kHz Crystal Oscillator Selector
- VDDBU-Powered

39.3. Block Diagram

Figure 39.1. SCKC Block Diagram



39.4. 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 VDDBU is present.

The embedded 32 kHz (typical) slow RC oscillator is always enabled as soon as VDDBU is established. The Slow Clock Selector command TD_OSCSEL bit selects the slow clock source of the RTT and the RTC.

After the VDDBU 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.

39.4.1. Switching from Embedded 32 kHz RC Oscillator to 32.768 kHz Crystal Oscillator

The sequence to switch from the embedded 32 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 32 kHz RC oscillator to the 32.768 kHz crystal oscillator by writing a 1 to the TD_OSCSEL bit.
3. Wait 39000 slow clock cycles for internal resynchronization.

39.4.2. Bypassing the 32.768 kHz Crystal Oscillator

The sequence to bypass the 32.768 kHz crystal oscillator is the following:

1. An external clock must be connected on XIN32.
2. Enable the bypass path by writing a 1 to the OSC32BYP bit.
3. Disable the 32.768 kHz crystal oscillator by writing a 0 to the OSC32EN bit.

39.4.3. Switching from 32.768 kHz Crystal Oscillator to Embedded 32 kHz RC Oscillator

The sequence to switch from the 32.768 kHz crystal oscillator to the embedded 32 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.
4. Disable the 32.768 kHz crystal oscillator by writing a 0 to the OSC32EN bit.

39.5. 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						OSC32BYP	OSC32EN	

39.5.1. 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
						OSC32BYP	OSC32EN	
Access						R/W	R/W	
Reset						0	0	

Bit 24 – TD_OSCSEL Timing Domain Slow Clock Selector

Value	Description
0 (RC)	Slow clock of the timing domain is driven by the embedded 32 kHz (typical) RC oscillator.
1 (XTAL)	Slow clock of the timing domain is driven by the 32.768 kHz crystal oscillator.

Bit 2 – OSC32BYP 32.768 kHz Crystal Oscillator Bypass

Value	Description
0	32.768 kHz crystal oscillator is not bypassed.
1	32.768 kHz crystal oscillator is bypassed and accepts an external slow clock on XIN32.

Bit 1 – OSC32EN 32.768 kHz Crystal Oscillator

Value	Description
0	32.768 kHz crystal oscillator is disabled.
1	32.768 kHz crystal oscillator is enabled.

40. Clock Generator

40.1. 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_.

40.2. Embedded Characteristics

The Clock Generator is made up of:

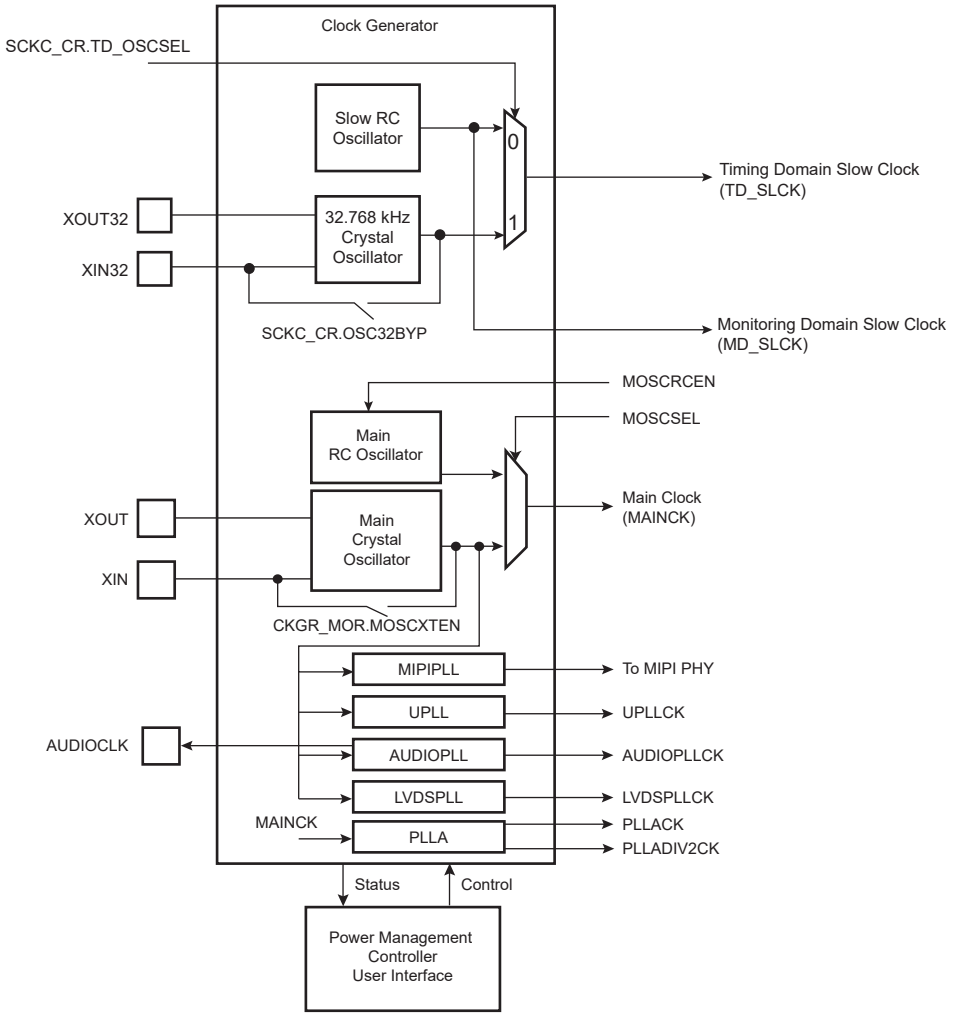
- Oscillators
  - A low-power 32.768 kHz oscillator supporting crystals, resonators and Bypass mode (referred to as “32.768 KHz crystal oscillator” throughout the document)
  - An embedded always-on, slow RC oscillator generating a typical 32 kHz clock
  - A 20 to 50 MHz oscillator supporting crystals, resonators and Bypass mode (referred to as “main crystal oscillator” throughout the document)
  - A main RC oscillator generating a typical 12 MHz clock
- Five fractional-N PLLs with an input range of 20 to 50 MHz and different internal frequency ranges
  - System PLL (PLLA) - PLL ID0
  - USB High-speed PLL (UPLL) - PLL ID1
  - Audio PLL (AUDIOPLL) - PLL ID2
  - LVDS PLL (LVDSPLL) - PLL ID3
  - System PLL divided by 2 (PLLADIV2) - PLL ID4

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

40.3. Block Diagram

Figure 40.1. Clock Generator Block Diagram



40.4. 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).

40.4.1. 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”.

40.4.2. 32.768 kHz Crystal Oscillator

By default, the 32.768 kHz oscillator is disabled. 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 to 1. The switch of TD_SLCK source is glitch-free.

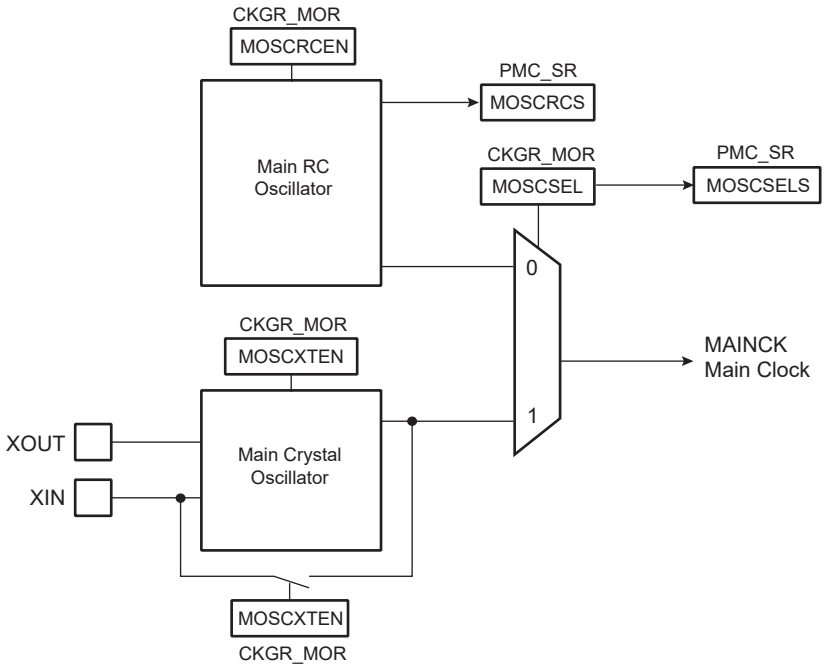
The user can also set the 32.768 kHz crystal oscillator in Bypass mode instead of connecting a crystal. In this case, the user must provide the external clock signal on XIN32. For input characteristics of the XIN32 pin, refer to the section “Electrical Characteristics”. To enter Bypass mode, the OSC32BYP bit of the Slow Clock Controller Configuration register (SCKC_CR) must be set to 1 prior to setting SCKC_CR.TD_OSCSEL.

40.5. 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 40.2. Main Clock (MAINCK) Block Diagram



40.5.1. 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.

40.5.2. 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.

40.5.3. 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 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 MCK is fed by MAINCK. PMC_SR.MOSCSELS indicates when the switch sequence is done.

Setting PMC_IMR.MOSCSELS triggers an interrupt to the processor.

40.5.4. 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.

40.5.5. 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

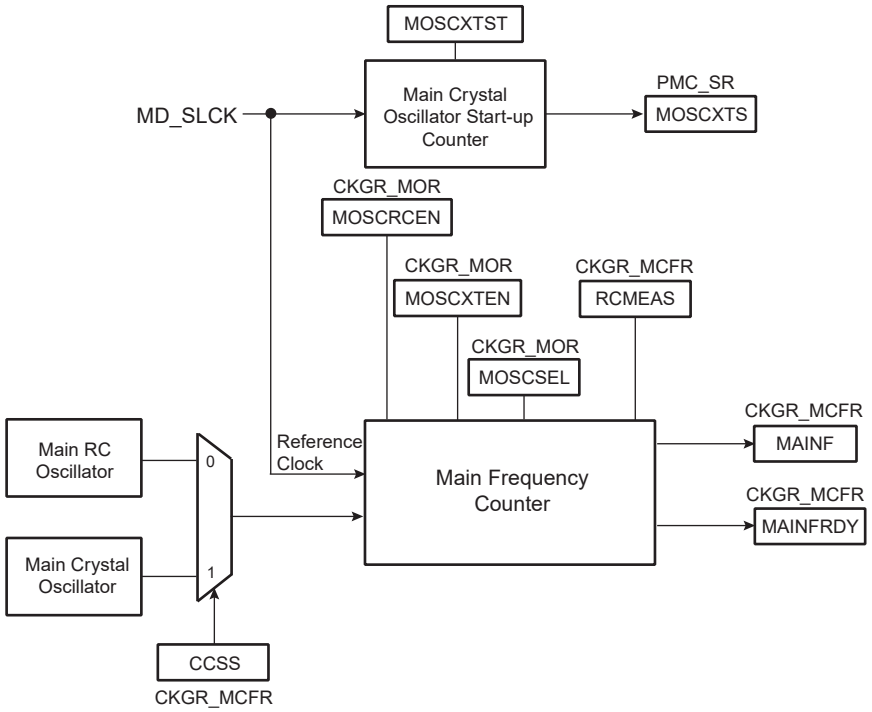
The measurement period ends at the 16th falling edge of MD_SLCK, the MAINFRDY bit in CKGR_MCFR is set and the counter stops counting. Its value can be read in the MAINF field of CKGR_MCFR and gives the number of clock cycles during 16 periods of MD_SLCK, so that the frequency of the main RC oscillator 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 40.3. Main Frequency Counter Block Diagram



40.6. PLL Controls

The PMC embeds 4 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.

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 40.1. PLL IDs

Index	PLL Name	Clock Name	PLL Clock Source	Usage Example
0	PLLA	PLLACK	MAINCK	CPU_CLK and MCK clock sources
1	UPLL	ULLCK	MAIN XTAL OSC	UTMI clock source
2	AUDIOPLL	AUDIOPLLCK	MAIN XTAL OSC	AUDIOCLK output clock source
3	LVDSPLL	LVDSPLLCK	MAIN XTAL OSC	LVDS clock source
4	PLLADIV2	PLLADIV2CK	MAINCK	Generic clocks source

40.6.1. 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 by the following formula:

$$f_{\mathrm{COREPLLCK}} = f_{\mathrm{ref}}\left(\mathrm{MUL} + 1 + \frac{\mathrm{FRACR}}{2^{22}}\right)$$

The PLLA (ID0) clock frequency is defined by the following formula:

$$f_{\mathrm{PLLA\ Clock}} = \frac{f_{\mathrm{COREPLLACK}}}{2 \times (\mathrm{DIVPMC} + 1)}$$

Note: The PLL general frequency is not applied to PLLA.

The UPLL clock frequency is defined by the following formula:

$$f_{\mathrm{UPLL\ Clock}} = \frac{f_{\mathrm{COREUPLLCK}}}{2}$$

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 IO 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.

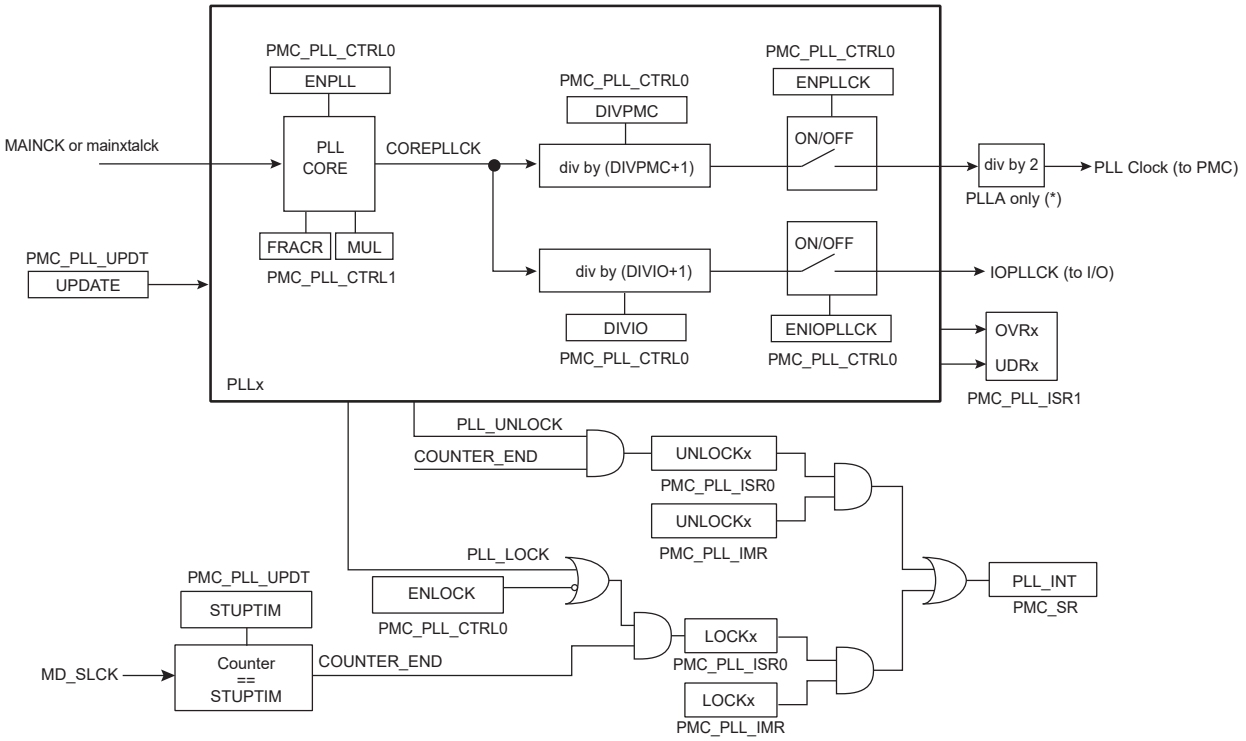
If neither the start-up time nor the lock are enabled, there is no way to know the lock status of the PLL.

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 40.4. PLL Controls



(*): This divider is only implemented at the output of PLLA (ID0).

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.
   If UPLL is being configured, follow Step 4. to Step 7., otherwise jump to Step 8.
4. Write PMC_PLL_ACR.UTMIBG to ‘1’ to enable the UTMI internal bandgap.
5. Wait 10 µs.
6. Write PMC_PLL_ACR.UTMIVR to ‘1’ to enable the UTMI internal regulator.
7. Wait 10 µs.
8. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1., otherwise the update is cancelled.
9. In PMC_PLL_CTRL0, write a ‘1’ to ENLOCK and to ENPLL and configure DIVPMC (for PLLA only, as UPLL has a fixed divider value), DIVIO, ENPLLCK and ENIOPLLCK.
10. Set PMC_PLL_UPDT.UPDATE to ‘1’. PMC_PLL_UPDT.ID must equal the one written during Step 1. otherwise the update is cancelled.
11. Wait for the lock bit to rise by polling the PMC_PLL_ISR0 or by enabling the corresponding interrupt in PMC_PLL_IER.
12. Disable the interrupt (if enabled).
13. 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'.

The clock generated by the AUDIOPLL is sent to the IO only 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.

40.6.2. UTMI PLL

The UTMI PLL is not controllable by PMC_PLL_CTRL0/1. It can be enabled by PMC_PLL_ACR. To specify the currently used main crystal frequency to the UTMI PHY, set the applicable value in PMC_XTALF.XTALF.

40.6.3. 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.

40.6.4. Spread Spectrum

Spread spectrum is obtained by slightly modifying the PLL target frequency. Two parameters are used to configure the spread spectrum:

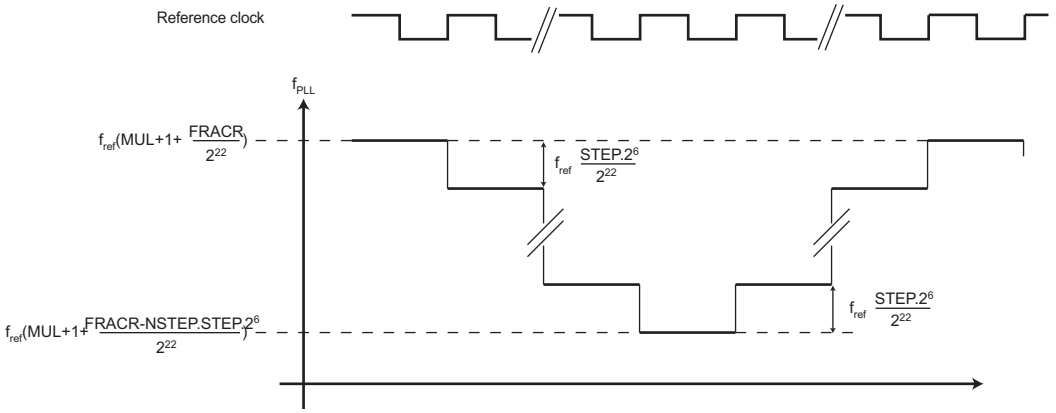
- STEP—the frequency step
- NSTEP—the number of times the STEP will be applied

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 40.5. Spread Spectrum Mechanism



41. Power Management Controller (PMC)

41.1. 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.

By default, at start-up, the chip runs out of MCK using the main RC oscillator.

41.2. Embedded Characteristics

The Power Management Controller provides the following clocks:

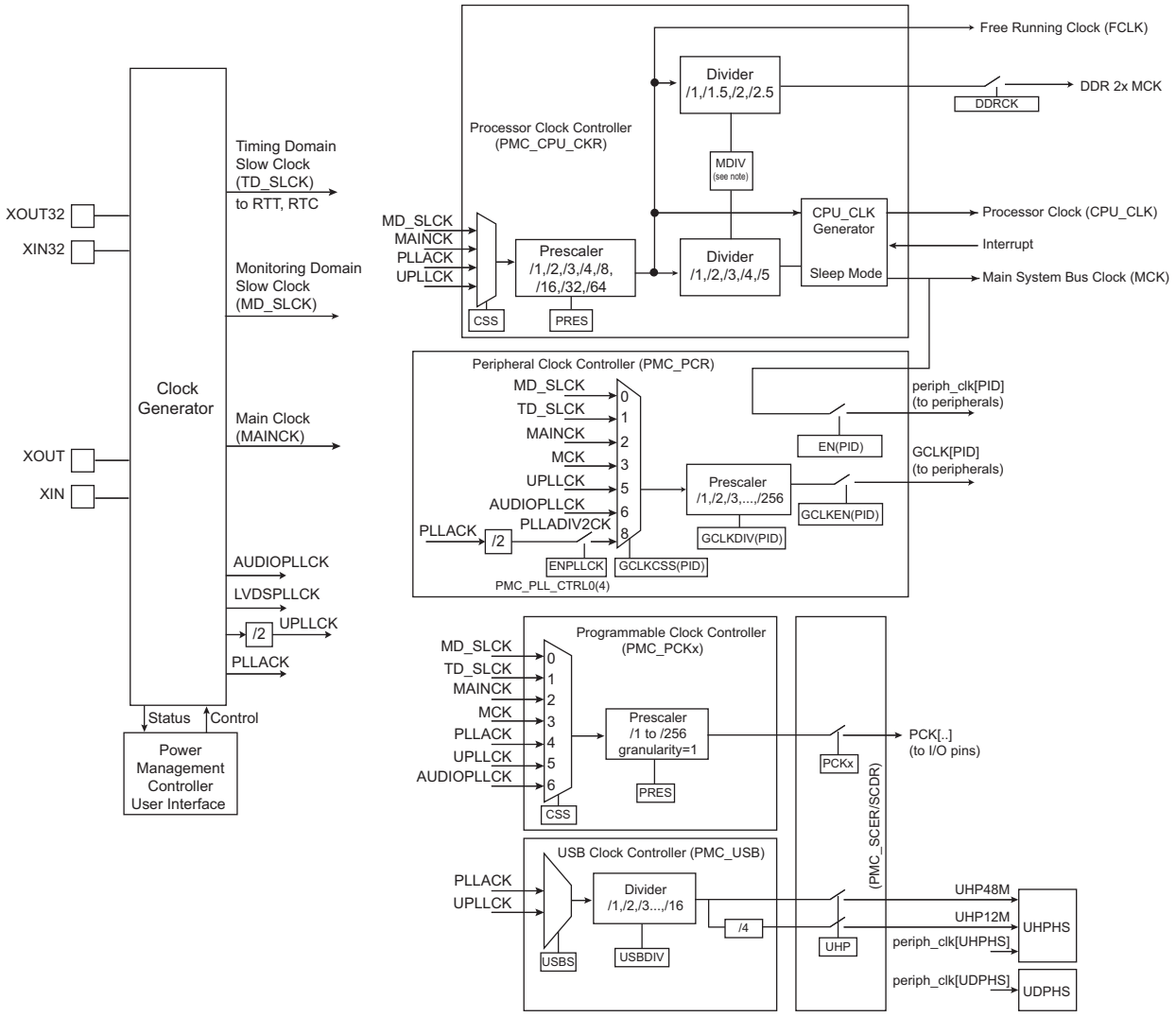
- Main System Bus Clock (MCK)—programmable from a few hundred Hz to the maximum operating frequency of the device. It is available to the modules running permanently.
- Processor Clock (CPU_CLK)—can be tuned through a frequency scaler module and automatically switched off when entering the processor in Sleep mode.
- Free-Running Processor Clock (FCLK)—the source clock of CPU_CLK. Is not affected when Sleep mode is activated.
- UHDP Clocks (UHP48M and UHP12M)—required by USB Host Device Port operations.
- Peripheral Clocks with independent on/off control, provided to the peripherals. Each peripheral clock is inherited from MCK.
- 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 MCK 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 MCK failure detector

41.3. Block Diagram

Figure 41.1. General Clock Distribution Block Diagram



Note: MDIV should always be different from 0 when using DDR memories. If MDIV must be set to 0, first switch the DDR memories to Self-refresh mode and disable DDRCLK.

41.4. 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) 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, MCK, which is a subdivision of the CPU_CLK.

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.

Note: Among all possible sources of GCLK, PLL(0) cannot be selected. PLL(4) (which is PLL(0) divided by 2) can be selected instead.

41.5. USB Clock Controller

The user can select the PLLA or the UPLL output as the USB source clock by writing PMC_USB.USBS. If using the USB, the user must program the PLL to generate an appropriate frequency depending on PMC_USB.USBDIV.

When the PLL output is stable, i.e., the LOCK bit is set, the USB device and host clocks can be enabled by setting the UHP bits in the System Clock Enable register (PMC_SCER). To save power on this peripheral when it is not used, the user can set the UHP bits in the System Clock Disable register (PMC_SCDR). The UHP bits in the System Clock Status register (PMC_SCSR) gives the activity of this clock. The USB device and host ports requires both the 48 MHz signal and the peripheral clock. The USB peripheral clock may be controlled by means of the Peripheral Clock Controller.

41.6. 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.

41.7. 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 MCK. 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, MCK 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.

41.8. Programmable Clock Output Controller

The PMC controls two 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, MCK and any PLLCK 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.

41.9. Ultra-Low Power Mode and Fast Start-Up

The following sections provide a brief description of the Ultra-Low Power mode features of the device as seen from the PMC. A more detailed description, including power consumption and wake-up time figures, can be found in the section "Electrical Characteristics".

41.9.1. ULP1 Mode

When the system is in Ultra-Low Power 1 (ULP1) mode, all clocks of the system except MD_SLCK are stopped. The source clock of all MCKx must be set to the main clock, and the source of the main clock must be set to main RC oscillator.

Prior to instructing the device to enter ULP1 mode:

1. Select Main RC as the source of MAINCK by configuring CKGR_MOR.MOSCSEL to '0'.
2. Select MAINCK as the source of MCK by configuring PMC_CPU_CKR.CSS to '1'.
3. Disable the PLL if enabled and disable the main crystal oscillator by setting CKGR_MOR.MOSCXTEN to '0'.
4. Wait for two SLCK clock cycles.
5. Clear the internal wake-up sources.
6. Verify that none of the enabled external wake-up inputs (WKUP) hold an active polarity.

The system enters ULP1 mode by setting CKGR_MOR.ULP1. The PMC registers must not be accessed immediately after this access.

41.9.2. Fast Start-Up

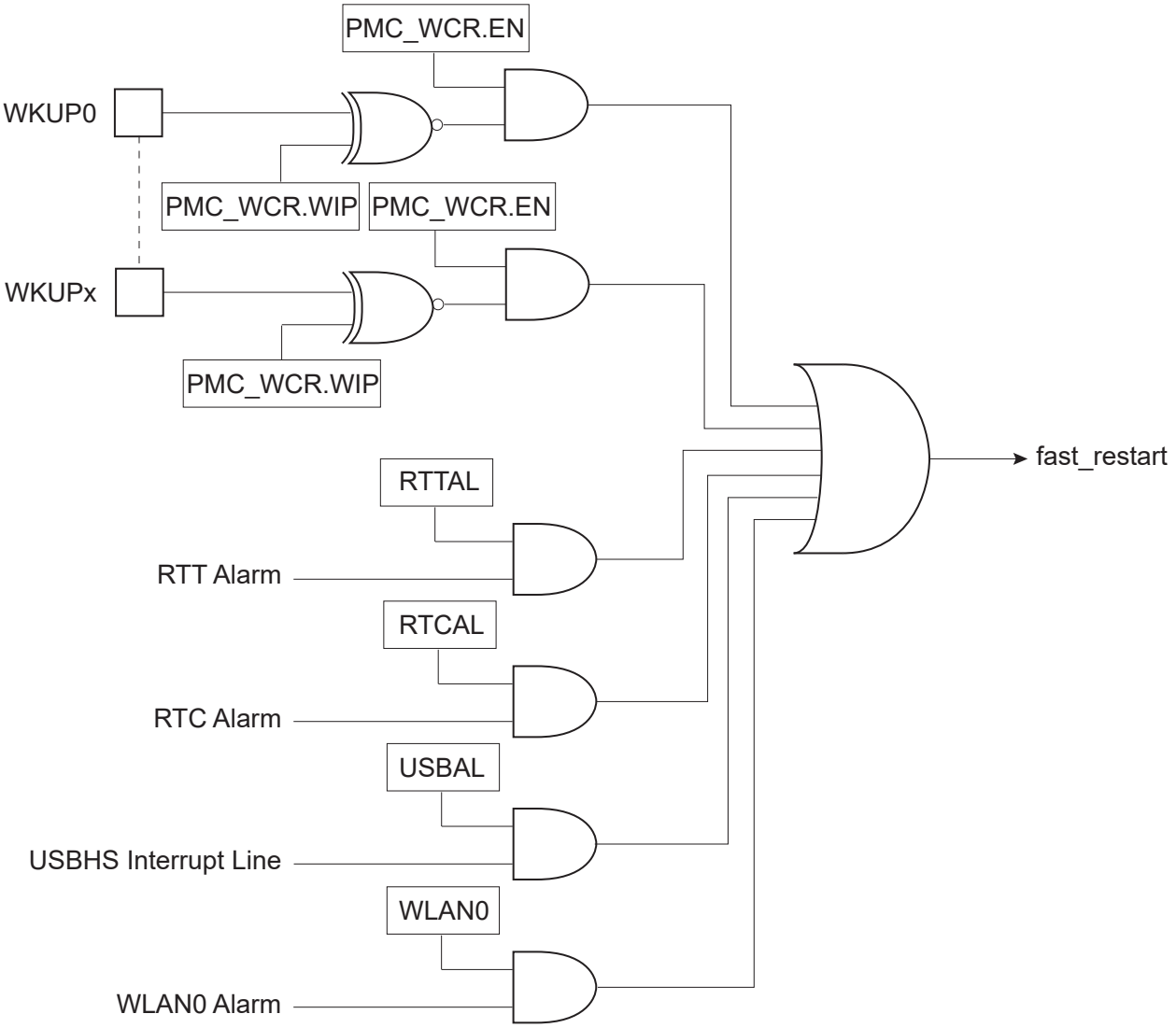
At exit from ULP1 mode, the device allows the processor to restart in several microseconds only if the C-code function that manages the ULP1 mode entry and exit is linked to and executed from on-chip SRAM.

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, USB Controller or Wake-up on LAN 0 alarm. 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 41.2. 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, WIP 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.

41.10. 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)

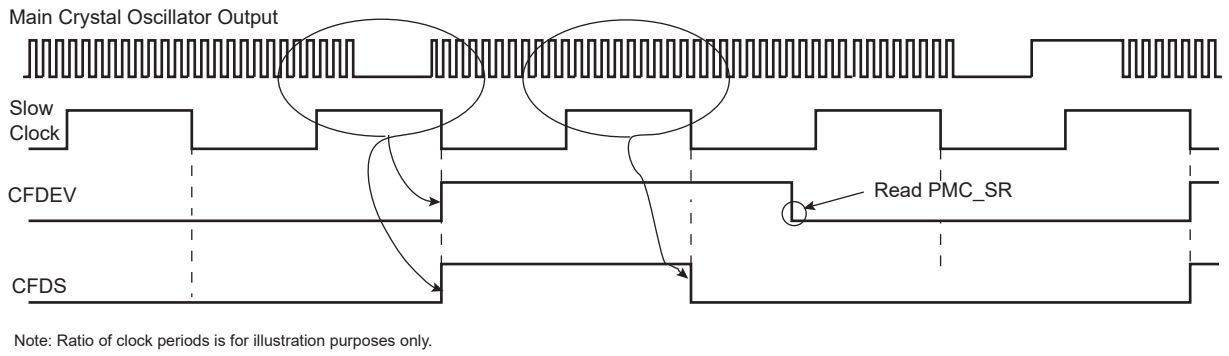
A failure is detected by means of a counter incrementing on the main crystal oscillator output and detection logic is triggered by the slow RC oscillator which is automatically enabled when CFDEN = 1.

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 41.3. Clock Failure Detection Example



Note: Ratio of clock periods is for illustration purposes only.

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 MCK 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'.

Six slow RC oscillator clock cycles are necessary to detect and switch from the main crystal oscillator to the main RC oscillator.

41.11. 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 permitted drift of the crystal is 10000 ppm (1%), which allows any standard crystal to be used.

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.

41.12. MCK Frequency Monitor

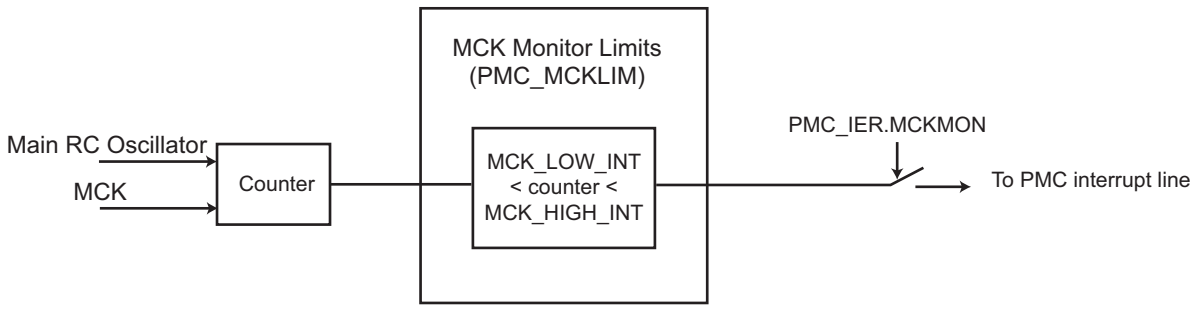
The frequency of MCK can be monitored with the main RC oscillator. This monitoring can only be performed if the MCK frequency is at least three times faster than the embedded main RC oscillator. This function is enabled by writing a ‘1’ to PMC_IER.MCKMON.

An error on the MCK frequency can lead to a PMC interrupt.

When the corresponding PMC interrupt is enabled, the status of the MCK monitoring can be read on PMC_SR.MCKMON. This status is cleared on read.

Once enabled, the monitor continuously counts the number of MCK 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 threshold can be defined to generate an interrupt.

Figure 41.4. MCK Frequency Monitor



41.13. 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 MCK division ratio by setting PMC_CPU_CKR.MDIV. Available values are 0, 1, 2, 3, 4. MCK 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 MCK.
7. Wait for the end of the MCK 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 MCK 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 MCK 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 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. Two 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.
13. 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.

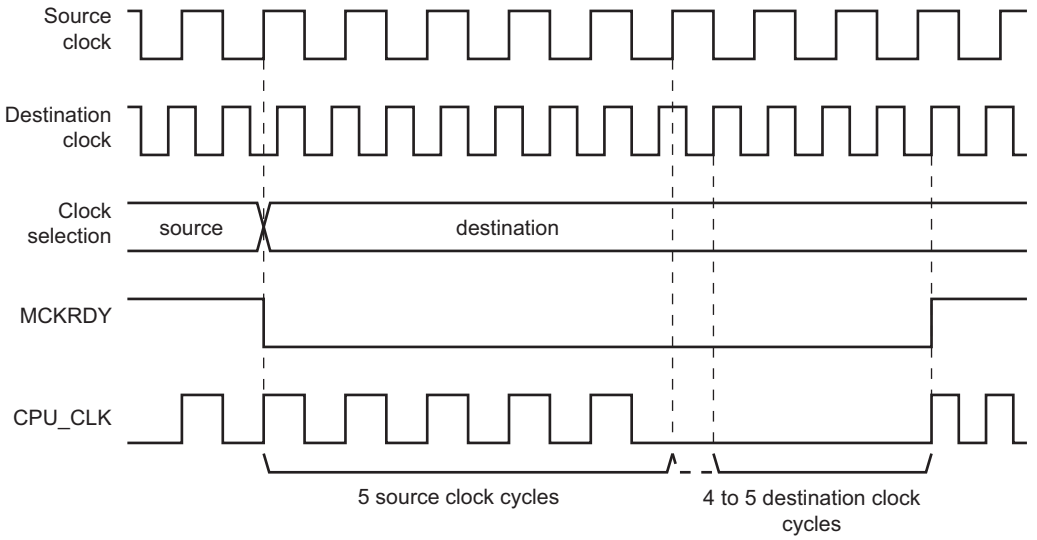
41.14. Clock Switching Details

41.14.1. CPU Clock Switching Timings

The glitch-free clock switcher implemented to control the sources of CPU_CLK and MCK 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 41.5. Switch CPU Clock (CPU_CLK) from Source Clock to Destination Clock



41.15. 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 USB Clock Register
- PMC Programmable Clock Register
- PMC Fast Start-Up Mode Register
- PMC Wake-Up Control Register
- PMC Peripheral Control Register
- PMC MCK 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

41.16. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PMC_SCER	31:24								
		23:16								
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
0x04	PMC_SCDR	31:24								
		23:16								
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
0x08	PMC_SCSR	31:24								
		23:16								
		15:8							PCK1	PCK0
		7:0		UHP				DDRCK		
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			UTMIBG	UTMIVR	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[2:0]		
0x20	CKGR_MOR	31:24		AUTOCPUSW	AUTOMAINSW			XT32KFME	CFDEN	MOSCSEL
		23:16	KEY[7:0]							
		15:8	MOSCXTST[7:0]							
		7:0					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 ... 0x37	Reserved									
0x38	PMC_USB	31:24								
		23:16								
		15:8					USBDIV[3:0]			
		7:0							USBS[1:0]	
0x3C ... 0x3F	Reserved									
0x40	PMC_PCK0	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	PMC_PCK1	31:24								
		23:16								
		15:8	PRES[7:0]							
		7:0				CSS[4:0]				
0x48 ... 0x5F	Reserved									
0x60	PMC_IER	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x64	PMC_IDR	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x68	PMC_SR	31:24							PLL_INT	GCLKRDY
		23:16	MCKMON		XT32KERR	FOS	CFDS	CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x6C	PMC_IMR	31:24							PLL_INT	
		23:16	MCKMON		XT32KERR			CFDEV	MOSCRCS	MOSCSELS
		15:8							PCKRDY1	PCKRDY0
		7:0					MCKRDY			MOSCXTS
0x70	PMC_FSMR	31:24								WLAN0
		23:16						USBAL	RTCAL	RTTAL
		15:8								
		7:0								
0x74	PMC_WCR	31:24								CMD
		23:16							WIP	EN
		15:8								
		7:0					WKPIONB[3:0]			
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]							
		15:8				GCLKCSS[4:0]				
		7:0		PID[6:0]						
0x8C ... 0x9B	Reserved									
0x9C	PMC_MCKLIM	31:24								
		23:16								
		15:8	MCK_HIGH_IT[7:0]							
		7:0	MCK_LOW_IT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xA0	PMC_CSR0	31:24		PID30	PID29	PID28		PID26	PID25	PID24
		23:16	PID23	PID22		PID20	PID19	PID18	PID17	PID16
		15:8	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
		7:0	PID7	PID6	PID5	PID4	PID3	PID2		
0xA4	PMC_CSR1	31:24					PID59			
		23:16		PID54	PID53	PID52	PID51		PID49	PID48
		15:8	PID47		PID45	PID44	PID43	PID42	PID41	PID40
		7:0	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
0xA8 ... 0xBF	Reserved									
0xC0	PMC_GCSR0	31:24		GPID30	GPID29			GPID26	GPID25	GPID24
		23:16					GPID19		GPID17	GPID16
		15:8	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	GPID8
		7:0	GPID7	GPID6	GPID5					
0xC4	PMC_GCSR1	31:24						GPID58		
		23:16	GPID55							
		15:8	GPID47		GPID45			GPID42		
		7:0			GPID37		GPID35	GPID34	GPID33	GPID32
0xC8 ... 0xDF	Reserved									
0xE0	PMC_PLL_IER	31:24								
		23:16					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
		15:8								
		7:0					LOCKLV	LOCKAU	LOCKU	LOCKA
0xE4	PMC_PLL_IDR	31:24								
		23:16					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
		15:8								
		7:0					LOCKLV	LOCKAU	LOCKU	LOCKA
0xE8	PMC_PLL_IMR	31:24								
		23:16					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
		15:8								
		7:0					LOCKLV	LOCKAU	LOCKU	LOCKA
0xEC	PMC_PLL_ISR0	31:24								
		23:16					LVDSUNLOCK	AUDIOUNLOCK	UNLOCKU	UNLOCKA
		15:8								
		7:0					LVDSLOCK	AUDIOLOCK	LOCKU	LOCKA
0xF0	PMC_PLL_ISR1	31:24								
		23:16						OVRAU		
		15:8								
		7:0						UDRAU		

41.16.1. 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
							PCK1	PCK0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
		UHP				DDRCK		
Access		W				W		
Reset		–				–		

Bits 8, 9 – PCKx Programmable Clock x Output Enable

Value	Description
0	No effect.
1	Enables the corresponding Programmable Clock output.

Bit 6 – UHP USB Host OHCI Clocks Enable

Value	Description
0	No effect.
1	Enables the UHP48M and UHP12M OHCI clocks.

Bit 2 – DDRCK MPDDRC Clock Enable

Value	Description
0	No effect.
1	Enables the MPDDRC clock.

41.16.2. 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
							PCK1	PCK0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
		UHP				DDRCK		
Access		W				W		
Reset		–				–		

Bits 8, 9 – PCKx Programmable Clock x Output Disable

Value	Description
0	No effect.
1	Disables the corresponding Programmable Clock output.

Bit 6 – UHP USB Host OHCI Clocks Disable

Value	Description
0	No effect.
1	Disables the UHP48M and UHP12M OHCI clocks.

Bit 2 – DDRCK MPDDRC Clock Disable

Value	Description
0	No effect.
1	Disables the MPDDRC clock.

41.16.3. 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
							PCK1	PCK0
Access							R	R
Reset							0	0

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

Bits 8, 9 – PCKx Programmable Clock x Output Status

Value	Description
0	The corresponding Programmable Clock output is disabled.
1	The corresponding Programmable Clock output is enabled.

Bit 6 – UHP USB Host OHCI Clocks Status

Value	Description
0	The UHP48M and UHP12M OHCI clocks are disabled.
1	The UHP48M and UHP12M OHCI clocks are enabled.

Bit 2 – DDRCK MPDDRC Clock Status

Value	Description
0	The MPDDRC clock is disabled.
1	The MPDDRC clock is enabled.

41.16.4. 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 and is configurable only when the DIVIO fied is programmed to 0.

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 and is configurable only when the DIVPMC fied is programmed to 0.

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 only available for AUDIOPLL. The frequency is defined by the following formula:

$$f_{\text{IOPLLCK}} = \frac{f_{\text{COREPLLCK}}}{(\text{DIVIO} + 1)}$$

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:

$$f_{\text{PLL Clock}} = \frac{f_{\text{COREPLLCK}}}{(\text{DIVPMC} + 1)}$$

For the PLLA, DIVPMC must be lower than or equal to 48 and its value must be different from 10, 12, 16, 18, 21, 22, 25, 26, 28, 30, 31, 32, 33, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46 and 47.

41.16.5. 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

41.16.6. 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.

41.16.7. PMC PLL Analog Control Register

Name: PMC_PLL_ACR
Offset: 0x0018
Reset: 0x00020033
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	1	0

Bit	15	14	13	12	11	10	9	8
			UTMIBG	UTMIVR	CONTROL[11:8]			
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
	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	1	1

Bits 29:24 – LOOP_FILTER[5:0] Loop Filter Selection

Bits 18:16 – LOCK_THR[2:0] PLL Lock Threshold Value Selection

Bit 13 – UTMIBG UPLL Bandgap Control
This bit has no effect when applied to PLLA.

Value	Description
0	The UPLL bandgap is switched off.
1	The UPLL bandgap is switched on.

Bit 12 – UTMIVR UPLL Voltage Regulator Control
This bit has no effect when applied to PLLA.

Value	Description
0	The UPLL voltage regulator is switched off.
1	The UPLL voltage regulator is switched on.

Bits 11:0 – CONTROL[11:0] PLL CONTROL Value Selection
On PLLA, this field controls the DCO analog filters:

Field	Description
CONTROL[1:0]	Analog VCO filter selection

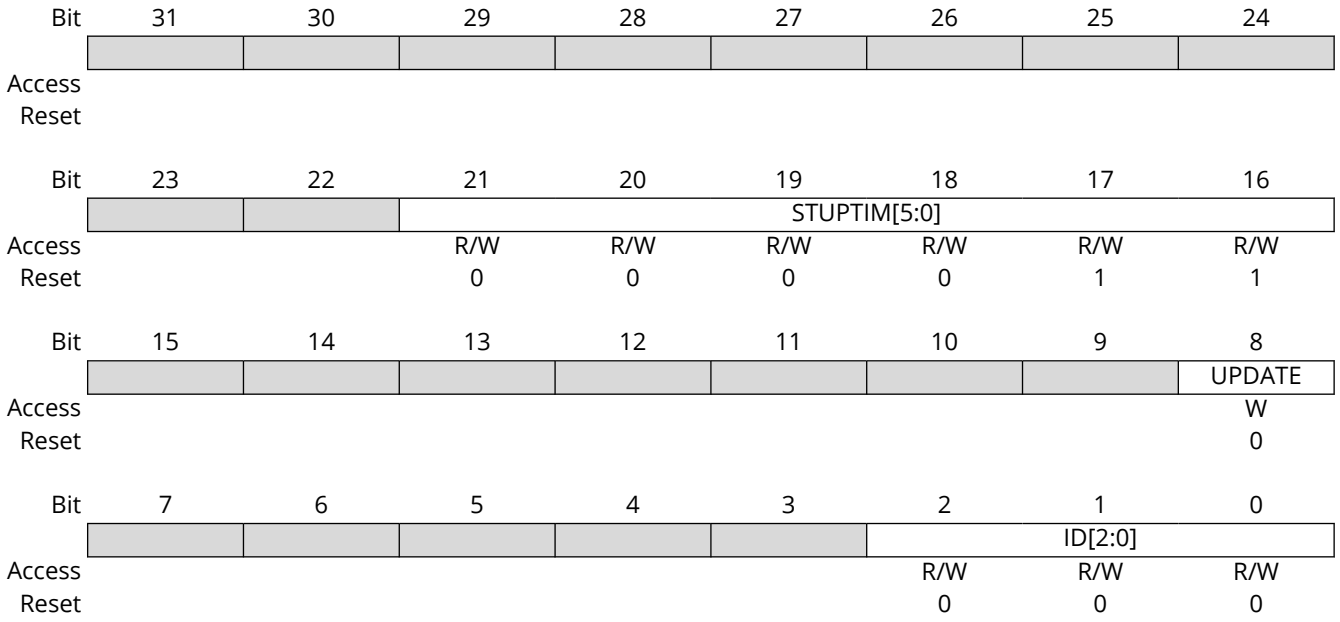
CONTROL (continued)	
Field	**Description**
CONTROL[4:2]	Process configuration
CONTROL[6:5]	VCO gain configuration
CONTROL[7]	Offset frequency adjustment
CONTROL[8]	External pad connection
CONTROL[9]	Test mode dedicated
CONTROL[10]	DAC mode
CONTROL[11]	Enable output phases

On UPLL, this field controls the following PLL ports:

Field	Description
CONTROL[1:0]	Not used
CONTROL[4:2]	Process configuration
CONTROL[6:5]	VCO gain configuration
CONTROL[7]	Offset frequency adjustment
CONTROL[11:8]	Not used

41.16.8. PMC PLL Update Register

Name: PMC_PLL_UPDT
Offset: 0x001C
Reset: 0x00030000
Property: Read/Write



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 2:0 – ID[2:0] 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.

Index	PLL Name	Clock Name	PLL Clock Source	Usage Example
0	PLLA	PLLACK	MAINCK	CPU_CLK and MCK clock sources
1	UPLL	ULLCK	MAIN XTAL OSC	UTMI clock source
2	AUDIOPLL	AUDIOPLLCK	MAIN XTAL OSC	AUDIOCLK output clock source
3	LVDSPLL	LVDSPLLCK	MAIN XTAL OSC	LVDS clock source

ID (continued)				
Index	**PLL Name**	**Clock Name**	**PLL Clock Source**	**Usage Example**
4	PLLA	PLLADIV2CK	MAINCK	This clock is PLLA divided by 2. Usage: GCLK source

41.16.9. PMC Clock Generator Main Oscillator Register

Name: CKGR_MOR
Offset: 0x0020
Reset: 0x00000028
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			XT32KFME	CFDEN	MOSCSEL
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
	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
					MOSCRCEN	ULP1		MOSCXTEN
Access					R/W	W		R/W
Reset					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 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.

Value	Description
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 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 (write-only)

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.

41.16.10.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				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	The 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 the 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:

$f_{SELCLK} = (MAINF \times f_{MD\_SLCK})/16$

where frequency is in MHz.

41.16.11.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 high, then modify the second field and wait for PMC_SR.MCKRDY high, 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] MCK Division

Value	Name	Description
0	EQ_PCK	MCK is FCLK divided by 1.
1	PCK_DIV2	MCK is FCLK divided by 2.
2	PCK_DIV4	MCK is FCLK divided by 4.
3	PCK_DIV3	MCK is FCLK divided by 3.
4	PCK_DIV5	MCK 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] MCK Source Selection

Value	Name	Description
0	SLOW_CLK	MD_SLCK is selected.
1	MAIN_CLK	MAINCK is selected.
2	PLLACK	PLLACK is selected.
3	UPLLCK	UPLLCK is selected.

41.16.12.PMC USB Clock Register

Name: PMC_USB
Offset: 0x0038
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
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					USBDIV[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
							USBS[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 11:8 – USBDIV[3:0] Divider for USB OHCI Clock
USB Clock is input clock divided by USBDIV + 1.

Bits 1:0 – USBS[1:0] USB OHCI/EHCI Input Clock Selection

Value	Name	Description
0	PLLA	The USB clock input is PLLACK.
1	UPLL	The USB clock input is UPLLCK.
2	–	Reserved
3	–	Reserved

41.16.13.PMC Programmable Clock Register

Name: PMC_PCKx
Offset: 0x40 + x*0x04 [x=0..1]
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	The 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	MCK	MCK is selected.
4	PLLA	PLLA is selected.
5	UPLL	UPLL is selected.
6	AUDIOPLL	Audio PLL is selected.

41.16.14.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
							PLL_INT	
Access							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
							PCKRDY1	PCKRDY0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 25 – PLL_INT PLL Interrupt Enable

Bit 23 – MCKMON Main System Bus Clock 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 – PCKRDYx Programmable Clock Ready x Interrupt Enable

Bit 3 – MCKRDY Main System Bus Clock Ready Interrupt Enable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Enable

41.16.15.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
							PLL_INT	
Access							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
							PCKRDY1	PCKRDY0
Access							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
					MCKRDY			MOSCXTS
Access					W			W
Reset					–			–

Bit 25 – PLL_INT PLL Interrupt Disable

Bit 23 – MCKMON Main System Bus Clock 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 – PCKRDYx Programmable Clock Ready x Interrupt Disable

Bit 3 – MCKRDY Main System Bus Clock Ready Interrupt Disable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Disable

41.16.16.PMC Status Register

Name: PMC_SR
Offset: 0x0068
Reset: 0x00030008
Property: Read-only

Bit	31	30	29	28	27	26	25	24
							PLL_INT	GCLKRDY
Access							R	R
Reset							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
							PCKRDY1	PCKRDY0
Access							R	R
Reset							0	0

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

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 Clock Clock Monitor Error
This status is cleared on read.

Value	Description
0	Main system bus clock is correct or the CPU clock monitor is disabled.
1	Main system bus clock 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	The main RC oscillator is not stabilized.
1	The 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 – PCKRDYx Programmable Clock Ready Status

Value	Description
0	Programmable clock x is not ready.
1	Programmable clock x is ready.

Bit 3 – MCKRDY Main System Bus Clock Status

Value	Description
0	Main system bus clock is not ready.
1	Main system bus clock is ready.

Bit 0 – MOSCXTS Main Crystal Oscillator Status

Value	Description
0	The main crystal oscillator is not stabilized.
1	The main crystal oscillator is stabilized.

41.16.17.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
							PLL_INT	
Access							W	
Reset							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
							PCKRDY1	PCKRDY0
Access							R	R
Reset							0	0

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

Bit 25 – PLL_INT PLL Interrupt Mask

Bit 23 – MCKMON Main System Bus Clock 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 – PCKRDYx Programmable Clock Ready x Interrupt Mask

Bit 3 – MCKRDY Main System Bus Clock Ready Interrupt Mask

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Mask

41.16.18.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
								WLAN0
Access								R/W
Reset								0

Bit	23	22	21	20	19	18	17	16
						USBAL	RTCAL	RTTAL
Access						R/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								

Bit 24 – WLAN0 Wake-Up on LAN[x] Enable

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 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.

41.16.19.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[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					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 WIP 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 3:0 – WKPIONB[3:0] 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
WKUP	0
PA2	1
PA7	2

WKPIONB (continued)	
Primary Signal Name	**WKPIONB**
PA8	3
PA20	4
PB0	5
PB3	6
PB18	7
PB25	8
PC23	9
PC24	10
PC31	11
PD25	12
PD26	13

41.16.20.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.

41.16.21.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).

41.16.22.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.

41.16.23.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]							
Access	R/W	R/W	R/W	R/W				
Reset	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 12:8 – GCLKCSS[4:0] Generic Clock Source Selection

Value	Name	Description
0	MD_CLK	MD_SLCK is selected.
1	TD_CLOCK	TD_SLCK is selected.
2	MAINCK	MAINCK is selected.

Value	Name	Description
3	MCK	MCK is selected.
4	–	Reserved
5	UPLLCK	UPLLCK is selected.
6	AUDIOPLLCK	AUDIOPLLCK is selected.
7	–	Reserved
8	PLLADIV2CLK	PLLADIV2CLK 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”.

41.16.24.PMC MCK Monitor Limits Register

Name: PMC_MCKLIM
Offset: 0x009C
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
	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 15:8 – MCK_HIGH_IT[7:0] MCK Monitoring High IT Limit
Beyond this limit, the MCK frequency monitor generates an interrupt.

Bits 7:0 – MCK_LOW_IT[7:0] MCK Monitoring Low IT Limit
Below this limit, the MCK frequency monitor generates an interrupt.

41.16.25.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	PID29	PID28		PID26	PID25	PID24
Access		R	R	R		R	R	R
Reset		0	0	0		0	0	0

Bit	23	22	21	20	19	18	17	16
	PID23	PID22		PID20	PID19	PID18	PID17	PID16
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
	PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
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
	PID7	PID6	PID5	PID4	PID3	PID2		
Access	R	R	R	R	R	R		
Reset	0	0	0	0	0	0		

Bits 28, 29, 30 – PIDx Peripheral Clock x Status

Bits 22, 23, 24, 25, 26 – PIDx Peripheral Clock x Status

Bits 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 – PIDx Peripheral Clock x Status

41.16.26.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
					PID59			
Access					R			
Reset					0			

Bit	23	22	21	20	19	18	17	16
		PID54	PID53	PID52	PID51		PID49	PID48
Access		R	R	R	R		R	R
Reset		0	0	0	0		0	0

Bit	15	14	13	12	11	10	9	8
	PID47		PID45	PID44	PID43	PID42	PID41	PID40
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
	PID39	PID38	PID37	PID36	PID35	PID34	PID33	PID32
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 27 – PIDx Peripheral Clock x Status

Bits 19, 20, 21, 22 – PIDx Peripheral Clock x Status

Bits 15, 16, 17 – PIDx Peripheral Clock x Status

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 – PIDx Peripheral Clock x Status

41.16.27.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	GPID25	GPID24
Access		R	R			R	R	R
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
					GPID19		GPID17	GPID16
Access					R		R	R
Reset					0		0	0

Bit	15	14	13	12	11	10	9	8
	GPID15	GPID14	GPID13	GPID12	GPID11	GPID10	GPID9	GPID8
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
	GPID7	GPID6	GPID5					
Access	R	R	R					
Reset	0	0	0					

Bits 29, 30 – GPIDx Generic Clock x Status

Bits 24, 25, 26 – GPIDx Generic Clock x Status

Bit 19 – GPIDx Generic Clock x Status

Bits 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 – GPIDx Generic Clock x Status

41.16.28.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
						GPID58		
Access						R		
Reset						0		

Bit	23	22	21	20	19	18	17	16
	GPID55							
Access	R							
Reset	0							

Bit	15	14	13	12	11	10	9	8
	GPID47		GPID45			GPID42		
Access	R		R			R		
Reset	0		0			0		

Bit	7	6	5	4	3	2	1	0
			GPID37		GPID35	GPID34	GPID33	GPID32
Access			R		R	R	R	R
Reset			0		0	0	0	0

Bit 26 – GPIDx Generic Clock x Status

Bit 23 – GPIDx Generic Clock x Status

Bit 15 – GPIDx Generic Clock x Status

Bit 13 – GPIDx Generic Clock x Status

Bit 10 – GPIDx Generic Clock x Status

Bit 5 – GPIDx Generic Clock x Status

Bits 0, 1, 2, 3 – GPIDx Generic Clock x Status

41.16.29.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
					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
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
					LOCKLV	LOCKAU	LOCKU	LOCKA
Access					W	W	W	W
Reset					–	–	–	–

Bit 19 – UNLOCKLV LVDS PLL Unlock Interrupt Enable

Bit 18 – UNLOCKAU AUDIO PLL Unlock Interrupt Enable

Bit 17 – UNLOCKU UPLL Unlock Interrupt Enable

Bit 16 – UNLOCKA PLLA Unlock Interrupt Enable

Bit 3 – LOCKLV LVDS PLL Lock Interrupt Enable

Bit 2 – LOCKAU AUDIO PLL Lock Interrupt Enable

Bit 1 – LOCKU UPLL Lock Interrupt Enable

Bit 0 – LOCKA PLLA Lock Interrupt Enable

41.16.30.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
					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
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
					LOCKLV	LOCKAU	LOCKU	LOCKA
Access					W	W	W	W
Reset					–	–	–	–

Bit 19 – UNLOCKLV LVDS PLL Unlock Interrupt Disable

Bit 18 – UNLOCKAU AUDIO PLL Unlock Interrupt Disable

Bit 17 – UNLOCKU UPLL Unlock Interrupt Disable

Bit 16 – UNLOCKA PLLA Unlock Interrupt Disable

Bit 3 – LOCKLV LVDS PLL Lock Interrupt Disable

Bit 2 – LOCKAU AUDIO PLL Lock Interrupt Disable

Bit 1 – LOCKU UPLL Lock Interrupt Disable

Bit 0 – LOCKA PLLA Lock Interrupt Disable

41.16.31.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
					UNLOCKLV	UNLOCKAU	UNLOCKU	UNLOCKA
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
					LOCKLV	LOCKAU	LOCKU	LOCKA
Access					R	R	R	R
Reset					0	0	0	0

Bit 19 – UNLOCKLV LVDS PLL Unlock Interrupt Mask

Bit 18 – UNLOCKAU AUDIO PLL Unlock Interrupt Mask

Bit 17 – UNLOCKU UPLL Unlock Interrupt Mask

Bit 16 – UNLOCKA PLLA Unlock Interrupt Mask

Bit 3 – LOCKLV LVDS PLL Lock Interrupt Mask

Bit 2 – LOCKAU AUDIO PLL Lock Interrupt Mask

Bit 1 – LOCKU UPLL Lock Interrupt Mask

Bit 0 – LOCKA PLLA Lock Interrupt Mask

41.16.32.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
					LVDSUNLOCK	AUDIOUNLOCK	UNLOCKU	UNLOCKA
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
					LVDSLOCK	AUDIOLOCK	LOCKU	LOCKA
Access					R	R	R	R
Reset					0	0	0	0

Bit 19 – LVDSUNLOCK LVDS PLL Unlock Interrupt Status

Value	Description
0	LVDS PLL is not unlocked.
1	LVDS PLL is unlocked.

Bit 18 – AUDIOUNLOCK Audio PLL Unlock Interrupt Status

Value	Description
0	Audio PLL is not unlocked.
1	Audio PLL is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bit 17 – UNLOCKU UPLL Unlock Interrupt Status

Value	Description
0	UPLL is not unlocked.
1	UPLL is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bit 16 – UNLOCKA PLLA Unlock Interrupt Status

Value	Description
0	PLLA is not unlocked.
1	PLLA is unlocked. To know the unlock type, the PMC_PISR1 register can be read.

Bit 3 – LVDSLOCK LVDS PLL Lock Interrupt Status

Value	Description
0	LVDS PLL is not locked.
1	LVDS PLL is locked.

Bit 2 – AUDIOLOCK Audio PLL Lock Interrupt Status

Value	Description
0	Audio PLL is not locked.
1	Audio PLL is locked.

Bit 1 – LOCKU UPLL Lock Interrupt Status

Value	Description
0	UPLL is not locked.
1	UPLL is locked.

Bit 0 – LOCKA PLLA Lock Interrupt Status

Value	Description
0	PLLA is not locked.
1	PLLA is locked.

41.16.33.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
						OVRAU		
Access						R		
Reset						0		

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 18 – OVRAU Audio PLL Overflow

Value	Description
0	AUDIO PLL is not in Overflow state.
1	AUDIO PLL encountered an overflow.

Bit 2 – UDRAU Audio PLL Underflow

Value	Description
0	AUDIO PLL is not in Underflow state.
1	AUDIO PLL encountered an underflow.

42. Parallel Input/Output Controller (PIO)

42.1. Description

The Parallel Input/Output Controller (PIO) manages up to 32 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.

Each I/O line is associated with a bit number in all of the 32-bit registers of the 32-bit wide user interface.

Each I/O line of the PIO Controller features the following:

- An input change interrupt enabling level change detection on any I/O line
- Additional Interrupt modes enabling rising edge, falling edge, low-level or high-level detection on any I/O line
- A glitch filter providing rejection of glitches lower than one-half of peripheral clock cycle
- A debouncing filter providing rejection of unwanted pulses from key or push button operations
- Multi-drive capability similar to an open drain I/O line
- Control of the I/O line pullup and pulldown
- Input visibility and output control

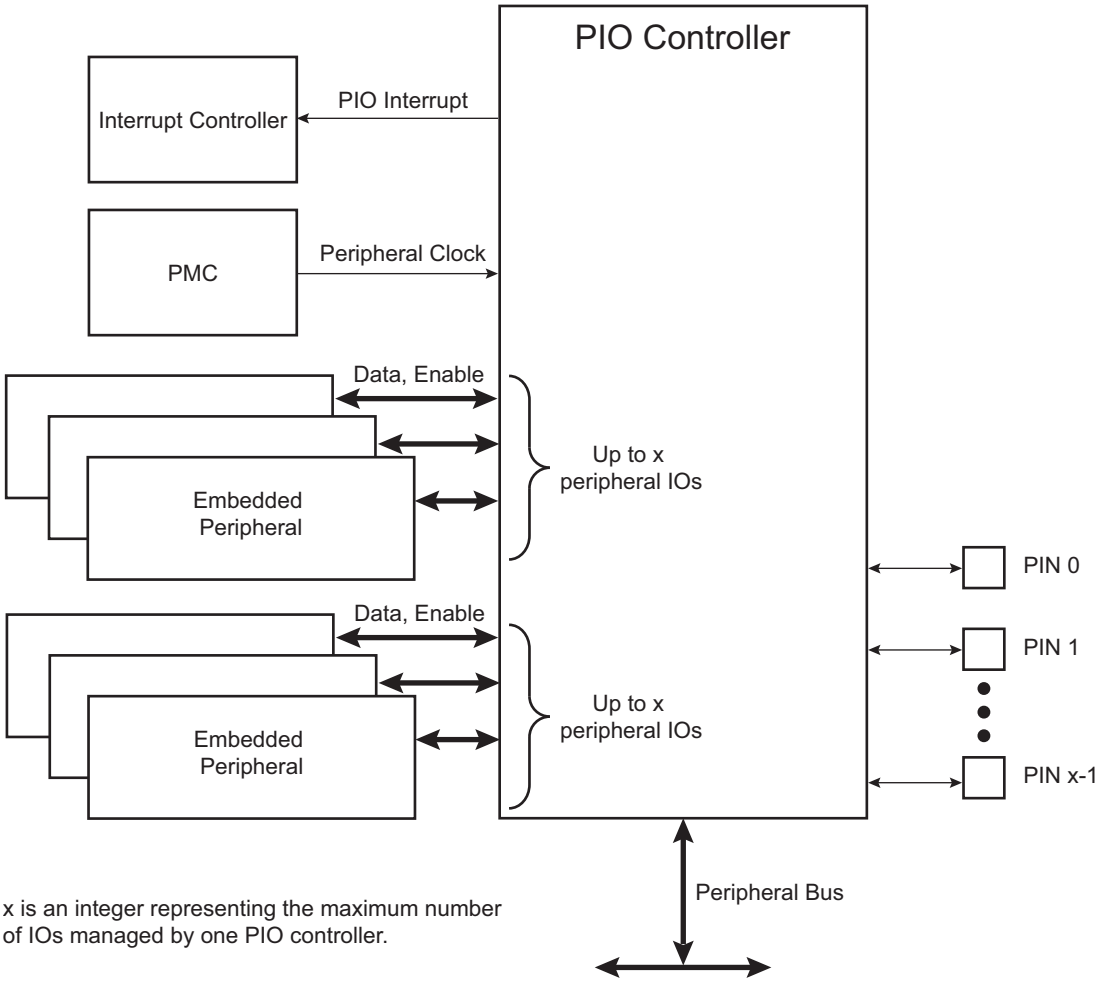
The PIO Controller also features a synchronous output providing up to 32 bits of data output in a single write operation.

42.2. Embedded Characteristics

- Up to 32 Programmable I/O Lines
- Fully Programmable through Set/Clear Registers
- Multiplexing of Four 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 pullup on each I/O line
  - Pin Data Status register provides visibility of the level on the pin at any time
  - Additional interrupt modes on a programmable event: Rising Edge, Falling Edge, Low-Level or High-Level
- Synchronous Output, Provides Set and Clear of Several I/O Lines in a Single Write
- Register Write Protection
- Programmable Schmitt Trigger Inputs
- Programmable Slewrate per I/O Line
- Programmable I/O Drive

42.3. Block Diagram

Figure 42.1. Block Diagram



42.4. Product Dependencies

42.4.1. 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 one or two 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.

42.4.2. External Interrupt Lines

The interrupt signals FIQ and IRQ0 to IRQn are generally multiplexed through the PIO Controllers. However, it is not necessary to assign the I/O line to the interrupt function as the PIO Controller has no effect on inputs and the external interrupt lines are used only as inputs.

42.4.3. Power Management

The Power Management Controller controls the peripheral clock in order to save power. Writing any of the registers of the user interface does not require the peripheral clock to be enabled. This means that the configuration of the I/O lines does not require the peripheral 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 peripheral clock is disabled by default.

The user must configure the Power Management Controller before any access to the input line information.

42.4.4. Interrupt Sources

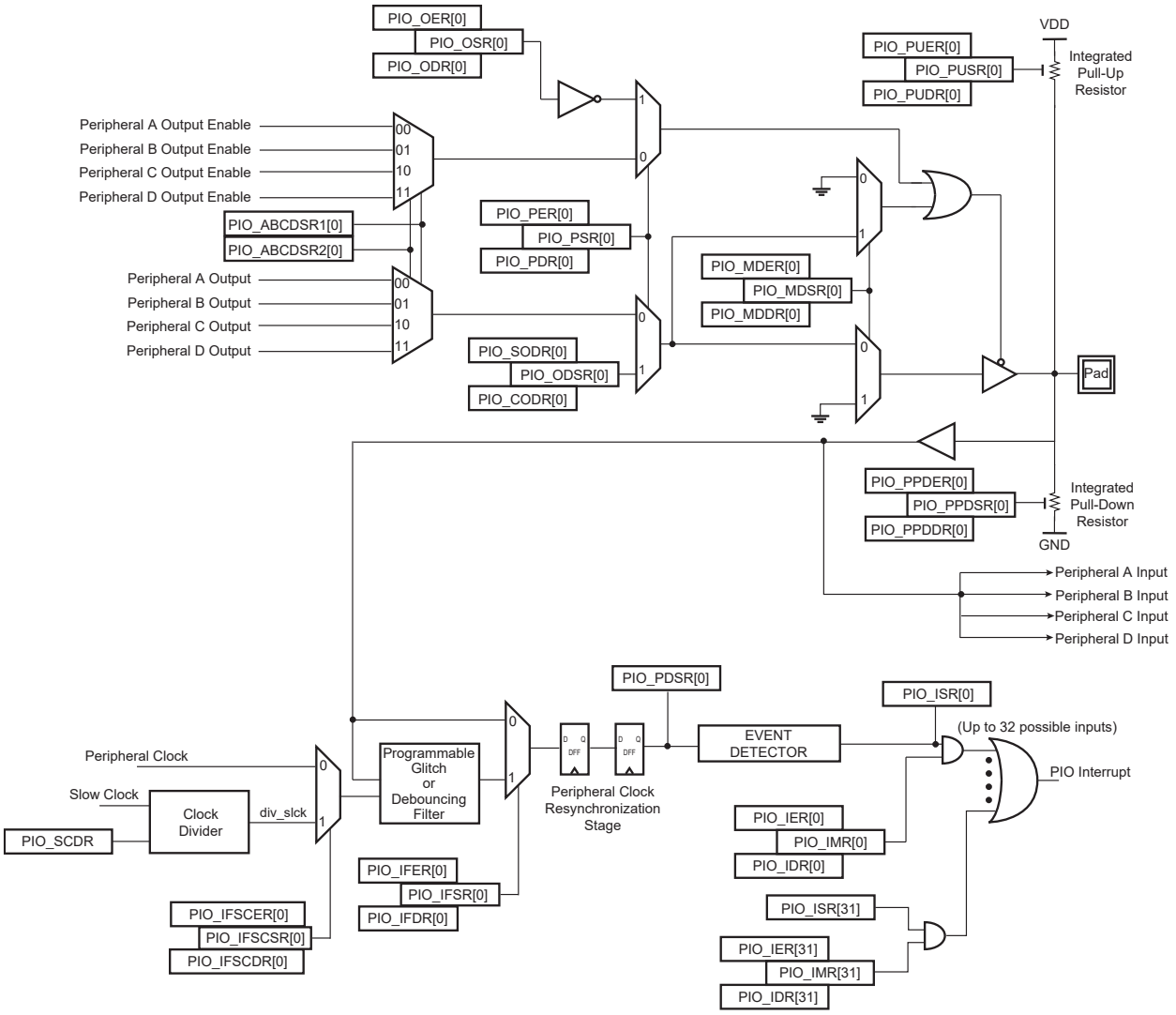
For interrupt handling, the PIO Controllers are considered as user peripherals. This means that the PIO Controller interrupt lines are connected among the interrupt sources. Refer to the PIO Controller peripheral identifier in the Peripheral Identifiers table to identify the interrupt sources dedicated to the PIO Controllers. Using the PIO Controller requires the Interrupt Controller to be programmed first.

The PIO Controller interrupt can be generated only if the peripheral clock is enabled.

42.5. Functional Description

The PIO Controller features up to 32 fully-programmable I/O lines. Most of the control logic associated to each I/O is represented in the following figure. In this description each signal shown represents one of up to 32 possible indexes.

Figure 42.2. I/O Line Control Logic



42.5.1. Pullup and Pulldown Resistor Control

Each I/O line is designed with an embedded pullup resistor and an embedded pulldown resistor. The pullup resistor can be enabled or disabled by writing to the Pull-Up Enable Register (PIO_PUER) or Pull-Up Disable Register (PIO_PUDR), respectively. Writing to these registers results in setting or clearing the corresponding bit in the Pull-Up Status Register (PIO_PUSR). Reading a one in PIO_PUSR means the pullup is disabled and reading a zero means the pullup is enabled. The pulldown resistor can be enabled or disabled by writing the Pull-Down Enable Register (PIO_PPDER) or the Pull-Down Disable Register (PIO_PPDDR), respectively. Writing in these registers results in setting or clearing the corresponding bit in the Pull-Down Status Register (PIO_PPDSR). Reading a one in PIO_PPDSR means the pullup is disabled and reading a zero means the pulldown is enabled.

Enabling the pulldown resistor while the pullup resistor is still enabled is not possible. In this case, the write of PIO_PPDER for the relevant I/O line is discarded. Likewise, enabling the pullup resistor while the pulldown resistor is still enabled is not possible. In this case, the write of PIO_PUER for the relevant I/O line is discarded.

Control of the pullup resistor is possible regardless of the configuration of the I/O line.

After reset, depending on the I/O, pullup or pulldown can be set.

42.5.2. I/O Line or Peripheral Function Selection

When a pin is multiplexed with one or two peripheral functions, the selection is controlled with the Enable Register (PIO_PER) and the Disable Register (PIO_PDR). The Status Register (PIO_PSR) is the result of the set and clear registers and indicates whether the pin is controlled by the corresponding peripheral or by the PIO Controller. A value of zero indicates that the pin is controlled by the corresponding on-chip peripheral selected in the Peripheral ABCD Select registers (PIO_ABCDSR1 and PIO_ABCDSR2). A value of one indicates the pin is controlled by the PIO Controller.

If a pin is used as a general-purpose I/O line (not multiplexed with an on-chip peripheral), PIO_PER and PIO_PDR have no effect and PIO_PSR returns a one for the corresponding bit.

After reset, the I/O lines are controlled by the PIO Controller, i.e., PIO_PSR resets at one. However, in some events, it is important that PIO lines are controlled by the peripheral (as in the case of memory chip select lines that must be driven inactive after reset, or for address lines that must be driven low for booting out of an external memory). Thus, the reset value of PIO_PSR is defined at the product level and depends on the multiplexing of the device.

42.5.3. Peripheral A or B or C or D Selection

The PIO Controller provides multiplexing of up to four peripheral functions on a single pin. The selection is performed by writing PIO_ABCDSR1 and PIO_ABCDSR2.

For each pin:

- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral A is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level zero in PIO_ABCDSR2 means peripheral B is selected.
- The corresponding bit at level zero in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral C is selected.
- The corresponding bit at level one in PIO_ABCDSR1 and the corresponding bit at level one in PIO_ABCDSR2 means peripheral D is selected.

Note that multiplexing of peripheral lines A, B, C and D only affects the output line. The peripheral input lines are always connected to the pin input (see the figure I/O Line Control Logic).

Writing in PIO_ABCDSR1 and PIO_ABCDSR2 manages the multiplexing regardless of the configuration of the pin. However, assignment of a pin to a peripheral function requires a write in PIO_ABCDSR1 and PIO_ABCDSR2 in addition to a write in PIO_PDR.

After reset, PIO_ABCDSR1 and PIO_ABCDSR2 are zero, thus indicating that all the PIO lines are configured on peripheral A. However, peripheral A generally does not drive the pin as the PIO Controller resets in I/O Line mode.

If the software selects a peripheral A, B, C or D which does not exist for a pin, no alternate functions are enabled for this pin and the selection is taken into account. The PIO Controller does not carry out checks to prevent selection of a peripheral which does not exist.

42.5.4. Output Control

When the I/O line is assigned to a peripheral function, i.e., the corresponding bit in PIO_PSR is at zero, the drive of the I/O line is controlled by the peripheral. Peripheral A or B or C or D depending on the value in PIO_ABCDSR1 and PIO_ABCDSR2 determines whether the pin is driven or not.

When the I/O line is controlled by the PIO Controller, the pin can be configured to be driven. This is done by writing the Output Enable Register (PIO_OER) and Output Disable Register (PIO_ODR). The results of these write operations are detected in the Output Status Register (PIO_OSR). When a bit in this register is at zero, the corresponding I/O line is used as an input only. When the bit is at one, the corresponding I/O line is driven by the PIO Controller.

The level driven on an I/O line can be determined by writing in the Set Output Data Register (PIO_SODR) and the Clear Output Data Register (PIO_CODR). These write operations, respectively, set and clear the Output Data Status Register (PIO_ODSR), which represents the data driven on the I/O lines. Writing in PIO_OER and PIO_ODR manages PIO_OSR 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_SODR and PIO_CODR affects PIO_ODSR. This is important as it defines the first level driven on the I/O line.

42.5.5. 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_SODR and PIO_CODR. 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_ODSR. Only bits unmasked by the Output Write Status Register (PIO_OWSR) are written. The mask bits in PIO_OWSR are set by writing to the Output Write Enable Register (PIO_OWER) and cleared by writing to the Output Write Disable Register (PIO_OWDR).

After reset, the synchronous data output is disabled on all the I/O lines as PIO_OWSR resets at 0x0.

42.5.6. Multi-Drive Control (Open Drain)

Each I/O can be independently programmed in open drain by using the multi-drive feature. This feature permits several drivers to be connected on the I/O line which is driven low only by each device. An external pullup resistor (or enabling of the internal one) is generally required to guarantee a high level on the line.

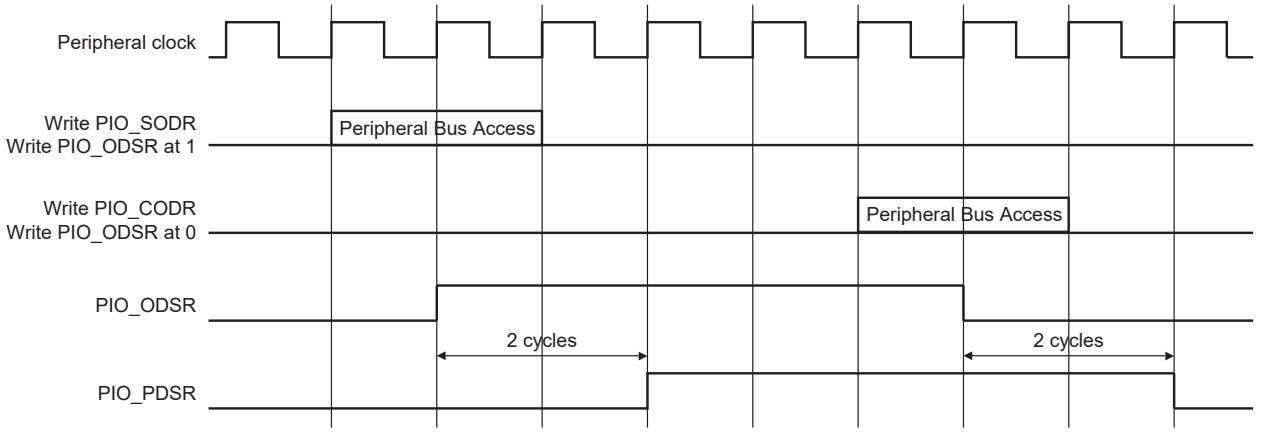
The multi-drive feature is controlled by the Multi-driver Enable Register (PIO_MDER) and the Multi-driver Disable Register (PIO_MDDR). The multi-drive can be selected whether the I/O line is controlled by the PIO Controller or assigned to a peripheral function. The Multi-driver Status Register (PIO_MDSR) indicates the pins that are configured to support external drivers.

After reset, the multi-drive feature is disabled on all pins, i.e., PIO_MDSR resets at value 0x0.

42.5.7. Output Line Timings

The following figure shows how the outputs are driven either by writing PIO_SODR or PIO_CODR, or by directly writing PIO_ODSR. This last case is valid only if the corresponding bit in PIO_OWSR is set. The Output Line Timings figure also shows when the feedback in the Pin Data Status Register (PIO_PDSR) is available.

Figure 42.3. Output Line Timings



42.5.8. Inputs

The level on each I/O line can be read through PIO_PDSR. 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_PDSR reads the levels present on the I/O line at the time the clock was disabled.

42.5.9. 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 1/2 peripheral clock and the debouncing filter can filter a pulse of less than 1/2 period of a programmable divided slow clock.

The selection between glitch filtering or debounce filtering is done by writing in the PIO Input Filter Slow Clock Disable register (PIO_IFSCDR) and the PIO Input Filter Slow Clock Enable register (PIO_IFSCER). Writing PIO_IFSCDR and PIO_IFSCER, respectively, sets and clears bits in the Input Filter Slow Clock Status register (PIO_IFSCSR).

The current selection status can be checked by reading the PIO_IFSCSR.

- If PIO_IFSCSR[i] = 0: The glitch filter can filter a glitch with a duration of less than 1/2 main system bus clock period.
- If PIO_IFSCSR[i] = 1: The debouncing filter can filter a pulse with a duration of less than 1/2 programmable divided slow clock period.

For the debouncing filter, the period of the divided slow clock is defined by writing in the DIV field of the Slow Clock Divider Debouncing register (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 1/2 selected clock cycle (selected clock represents peripheral clock or divided slow clock depending on PIO_IFSCDR and PIO_IFSCER programming) is automatically rejected, while a pulse with a duration of one selected clock (peripheral clock or divided slow clock) cycle or more is accepted. For pulse durations between 1/2 selected clock cycle and one selected clock cycle, the pulse may or may not be taken into account, depending on the precise timing of its occurrence. Thus for a pulse to be visible, it must exceed one selected clock cycle, whereas for a glitch to be reliably filtered out, its duration must not exceed 1/2 selected clock cycle.

The filters also introduce some latencies, illustrated in the following two figures.

The glitch filters are controlled by the Input Filter Enable register (PIO_IFER), the Input Filter Disable register (PIO_IFDR) and the Input Filter Status register (PIO_IFSR). Writing PIO_IFER and PIO_IFDR respectively sets and clears bits in PIO_IFSR. This last register enables the glitch filter on the I/O lines.

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_PDSR and on the input change interrupt detection. The glitch and debouncing filters require that the peripheral clock is enabled.

Figure 42.4. Input Glitch Filter Timing

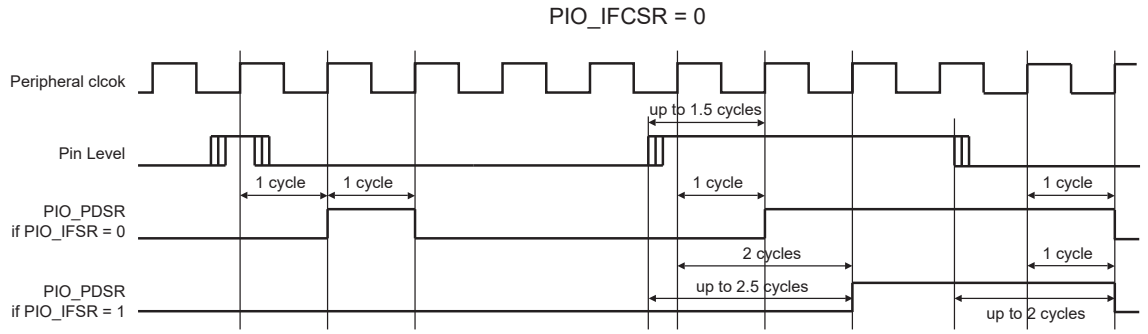
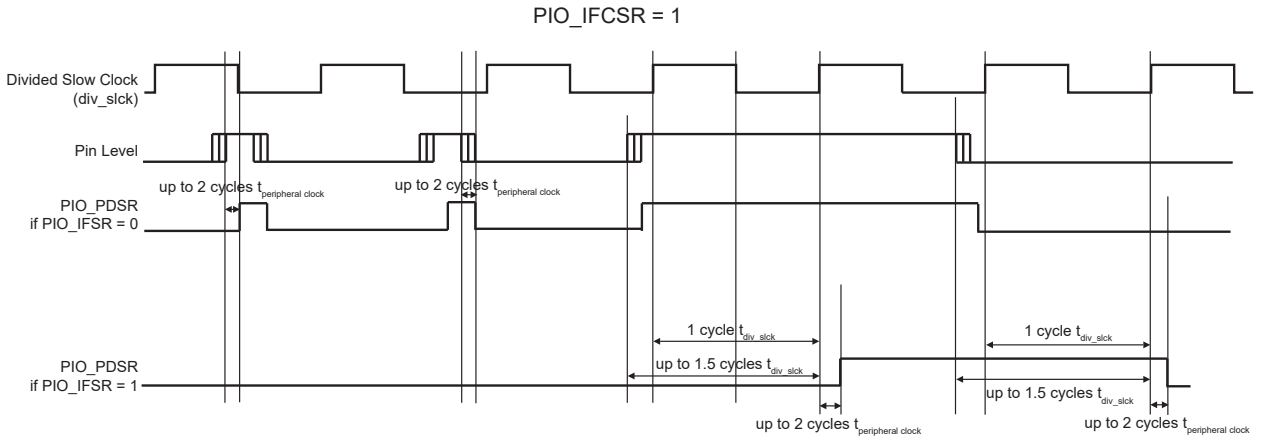


Figure 42.5. Input Debouncing Filter Timing



42.5.10. Input Edge/Level Interrupt

The PIO Controller can be programmed to generate an interrupt when it detects an edge or a level on an I/O line. The Input Edge/Level interrupt is controlled by writing the Interrupt Enable Register (PIO_IER) and the Interrupt Disable Register (PIO_IDR), which enable and disable the input change interrupt respectively by setting and clearing the corresponding bit in the Interrupt Mask Register (PIO_IMR). As input change detection is possible only by comparing two successive samplings of the input of the I/O line, the peripheral 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.

By default, the interrupt can be generated at any time an edge is detected on the input.

Some additional interrupt modes can be enabled/disabled by writing in the Additional Interrupt Modes Enable Register (PIO_AIMER) and Additional Interrupt Modes Disable Register (PIO_AIMDR). The current state of this selection can be read through the Additional Interrupt Modes Mask Register (PIO_AIMMR).

These additional modes are:

- Rising edge detection
- Falling edge detection
- Low-level detection
- High-level detection

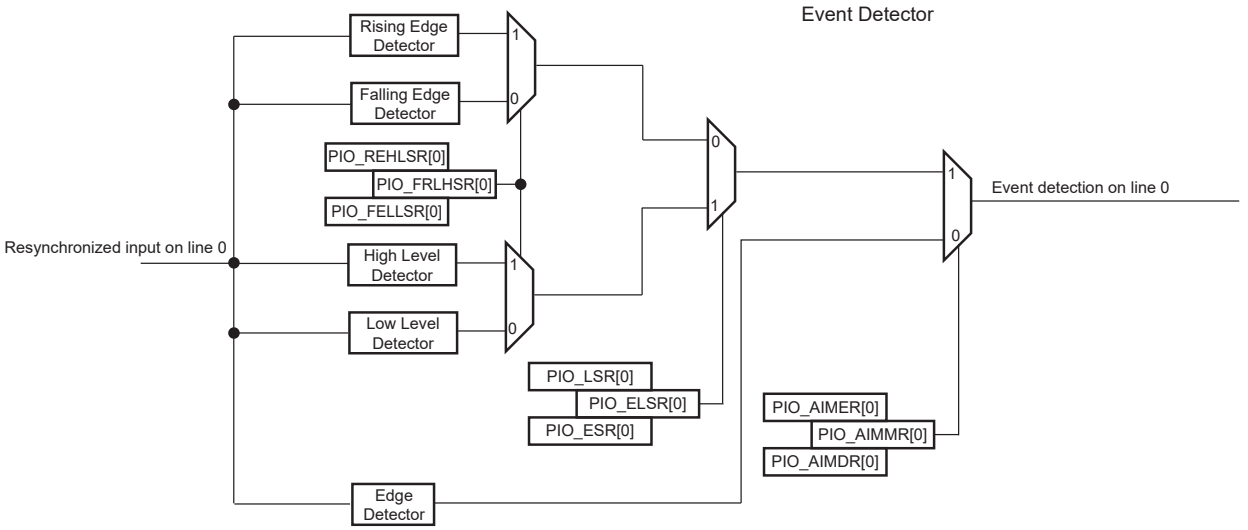
In order to select an additional interrupt mode:

- The type of event detection (edge or level) must be selected by writing in the Edge Select Register (PIO_ESR) and Level Select Register (PIO_LSR) which select, respectively, the edge and level detection. The current status of this selection is accessible through the Edge/Level Status Register (PIO_ELSR).
- The polarity of the event detection (rising/falling edge or high/low-level) must be selected by writing in the Falling Edge/Low-Level Select Register (PIO_FELLSR) and Rising Edge/High-Level Select Register (PIO_REHLSR) which allow to select falling or rising edge (if edge is selected in PIO_ELSR) edge or high- or low-level detection (if level is selected in PIO_ELSR). The current status of this selection is accessible through the Fall/Rise - Low/High Status Register (PIO_FRLHSR).

When an input edge or level is detected on an I/O line, the corresponding bit in the Interrupt Status Register (PIO_ISR) is set. If the corresponding bit in PIO_IMR is set, the PIO Controller interrupt line is asserted.The interrupt signals of the 32 channels are ORed-wired together to generate a single interrupt signal to the interrupt controller.

When the software reads PIO_ISR, all the interrupts are automatically cleared. This signifies that all the interrupts that are pending when PIO_ISR is 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_ISR are performed.

Figure 42.6. Event Detector on Input Lines (Figure Represents Line 0)



Example of interrupt generation on following lines:

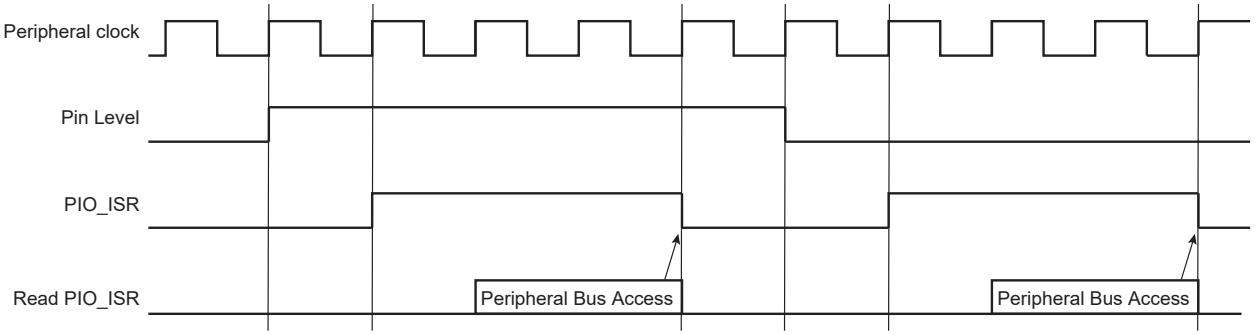
- Rising edge on PIO line 0
- Falling edge on PIO line 1
- Rising edge on PIO line 2
- Low-level on PIO line 3
- High-level on PIO line 4
- High-level on PIO line 5
- Falling edge on PIO line 6
- Rising edge on PIO line 7
- Any edge on the other lines

The following table provides the required configuration for this example.

Table 42.1. Configuration for Example Interrupt Generation

Configuration	Description
Interrupt Mode	All the interrupt sources are enabled by writing 32'hFFFF_FFFF in PIO_IER. Then the additional Interrupt mode is enabled for lines 0 to 7 by writing 32'h0000_00FF in PIO_AIMER.
Edge or Level Detection	Lines 3, 4 and 5 are configured in level detection by writing 32'h0000_0038 in PIO_LSR. The other lines are configured in edge detection by default, if they have not been previously configured. Otherwise, lines 0, 1, 2, 6 and 7 must be configured in edge detection by writing 32'h0000_00C7 in PIO_ESR.
Falling/Rising Edge or Low/High-Level Detection	Lines 0, 2, 4, 5 and 7 are configured in rising edge or high-level detection by writing 32'h0000_00B5 in PIO_REHLSR. The other lines are configured in falling edge or low-level detection by default if they have not been previously configured. Otherwise, lines 1, 3 and 6 must be configured in falling edge/low-level detection by writing 32'h0000_004A in PIO_FELLSR.

Figure 42.7. Input Change Interrupt Timings When No Additional Interrupt Modes



42.5.11. Programmable Schmitt Trigger

It is possible to configure each input for the Schmitt trigger. By default the Schmitt trigger is active. Disabling the Schmitt trigger is requested when using the QTouch® Library.

42.5.12. I/O Lines Programming Example

The programming example shown in the following table is used to obtain the following configuration:

- 4-bit output port on I/O lines 0 to 3 (should be written in a single write operation), open-drain, with pullup resistor
- Four output signals on I/O lines 4 to 7 (to drive LEDs for example), driven high and low, no pullup resistor, no pulldown resistor
- Four input signals on I/O lines 8 to 11 (to read push-button states for example), with pullup resistors, glitch filters and input change interrupts
- Four input signals on I/O line 12 to 15 to read an external device status (polled, thus no input change interrupt), no pullup resistor, no glitch filter
- I/O lines 16 to 19 assigned to peripheral A functions with pullup resistor
- I/O lines 20 to 23 assigned to peripheral B functions with pulldown resistor
- I/O lines 24 to 27 assigned to peripheral C with input change interrupt, no pullup resistor and no pulldown resistor
- I/O lines 28 to 31 assigned to peripheral D, no pullup resistor and no pulldown resistor

Table 42.2. Programming Example

Register	Value to be Written
PIO_PER	0x0000_FFFF

Table 42.2. Programming Example (continued)

Register	Value to be Written
PIO_PDR	0xFFFF_0000
PIO_OER	0x0000_00FF
PIO_ODR	0xFFFF_FF00
PIO_IFER	0x0000_0F00
PIO_IFDR	0xFFFF_F0FF
PIO_SODR	0x0000_0000
PIO_CODR	0x0FFF_FFFF
PIO_IER	0x0F00_0F00
PIO_IDR	0xF0FF_F0FF
PIO_MDER	0x0000_000F
PIO_MDDR	0xFFFF_FFF0
PIO_PUDR	0xFFF0_00F0
PIO_PUER	0x000F_FF0F
PIO_PPDDR	0xFF0F_FFFF
PIO_PPDER	0x00F0_0000
PIO_ABCDSR1	0xF0F0_0000
PIO_ABCDSR2	0xFF00_0000
PIO_OWER	0x0000_000F
PIO_OWDR	0x0FFF_FFF0

42.5.13. 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 the WPEN bit in the PIO Write Protection Mode Register (PIO_WPMR).

If a write access to a 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.

The WPVS bit is automatically cleared after reading PIO_WPSR.

The following registers can be write-protected:

- PIO Enable Register
- PIO Disable Register
- PIO Output Enable Register
- PIO Output Disable Register
- PIO Input Filter Enable Register
- PIO Input Filter Disable Register
- PIO Multi-driver Enable Register
- PIO Multi-driver Disable Register
- PIO Pull-Up Disable Register
- PIO Pull-Up Enable Register
- PIO Peripheral ABCD Select Register 1
- PIO Peripheral ABCD Select Register 2
- PIO Output Write Enable Register
- PIO Output Write Disable Register

- PIO Pad Pull-Down Disable Register
- PIO Pad Pull-Down Enable Register

42.6. Register Summary

Each I/O line controlled by the PIO Controller is associated with a bit in each of the PIO Controller User Interface registers. Each register is 32 bits wide. If a parallel I/O line is not defined, writing to the corresponding bits has no effect. Undefined bits read zero. If the I/O line is not multiplexed with any peripheral, the I/O line is controlled by the PIO Controller and PIO_PSR returns one systematically.

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PIO_PER	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
0x04	PIO_PDR	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
0x08	PIO_PSR	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 ... 0x0F	Reserved									
0x10	PIO_OER	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_ODR	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_OSR	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_IFER	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_IFDR	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_IFSR	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 ... 0x2F	Reserved									
0x30	PIO_SODR	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
0x34	PIO_CODR	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	PIO_ODSR	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
0x3C	PIO_PDSR	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
0x40	PIO_IER	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
0x44	PIO_IDR	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
0x48	PIO_IMR	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_ISR	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_MDER	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_MDDR	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_MDSR	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_PUDR	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_PUER	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_PUSR	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 ... 0x6F	Reserved									
0x70	PIO_ABCDSR1	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	PIO_ABCDSR2	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
0x78 ... 0x7F	Reserved									
0x80	PIO_IFSCDR	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
0x84	PIO_IFSCER	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
0x88	PIO_IFSCSR	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_SCDR	31:24								
		23:16								
		15:8			DIV[13:8]					
		7:0	DIV[7:0]							
0x90	PIO_PPDDR	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_PPDER	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_PPDSR	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_OWER	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_OWDR	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_OWSR	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 ... 0xAF	Reserved									
0xB0	PIO_AIMER	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB4	PIO_AIMDR	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
0xB8	PIO_AIMMR	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
0xBC ... 0xBF	Reserved									
0xC0	PIO_ESR	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
0xC4	PIO_LSR	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
0xC8	PIO_ELSR	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 ... 0xCF	Reserved									
0xD0	PIO_FELLSR	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_REHLSR	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_FRLHSR	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 ... 0xE3	Reserved									
0xE4	PIO_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN
0xE8	PIO_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0xEC ... 0xFF	Reserved									
0x0100	PIO_SCHMITT	31:24	SCHMITT31	SCHMITT30	SCHMITT29	SCHMITT28	SCHMITT27	SCHMITT26	SCHMITT25	SCHMITT24
		23:16	SCHMITT23	SCHMITT22	SCHMITT21	SCHMITT20	SCHMITT19	SCHMITT18	SCHMITT17	SCHMITT16
		15:8	SCHMITT15	SCHMITT14	SCHMITT13	SCHMITT12	SCHMITT11	SCHMITT10	SCHMITT9	SCHMITT8
		7:0	SCHMITT7	SCHMITT6	SCHMITT5	SCHMITT4	SCHMITT3	SCHMITT2	SCHMITT1	SCHMITT0
0x0104 ... 0x010F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0110	PIO_SLEWR	31:24	SR31	SR30	SR29	SR28	SR27	SR26	SR25	SR24
		23:16	SR23	SR22	SR21	SR20	SR19	SR18	SR17	SR16
		15:8	SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8
		7:0	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
0x0114 ... 0x0117	Reserved									
0x0118	PIO_DRIVER1	31:24	DR31	DR30	DR29	DR28	DR27	DR26	DR25	DR24
		23:16	DR23	DR22	DR21	DR20	DR19	DR18	DR17	DR16
		15:8	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
		7:0	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

42.6.1. PIO Enable Register

Name: PIO_PER
Offset: 0x0000
Reset: –
Property: Write-only

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
	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 – P PIO Enable

Value	Description
0	No effect.
1	Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

42.6.2. PIO Disable Register

Name: PIO_PDR
Offset: 0x0004
Reset: –
Property: Write-only

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
	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 – P PIO Disable

Value	Description
0	No effect.
1	Disables the PIO from controlling the corresponding pin (enables peripheral control of the pin).

42.6.3. PIO Status Register

Name: PIO_PSR
Offset: 0x0008
Property: Read-only

Reset values depend on the product implementation.

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 – P PIO Status

Value	Description
0	PIO is inactive on the corresponding I/O line (peripheral is active).
1	PIO is active on the corresponding I/O line (peripheral is inactive).

42.6.4. PIO Output Enable Register

Name: PIO_OER
Offset: 0x0010
Reset: –
Property: Write-only

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
	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 – P PIO Output Enable

Value	Description
0	No effect.
1	Enables the output on the I/O line.

42.6.5. PIO Output Disable Register

Name: PIO_ODR
Offset: 0x0014
Reset: –
Property: Write-only

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
	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 – P PIO Output Disable

Value	Description
0	No effect.
1	Disables the output on the I/O line.

42.6.6. PIO Output Status Register

Name: PIO_OSR
Offset: 0x0018
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 – P PIO Output Status

Value	Description
0	The I/O line is a pure input.
1	The I/O line is enabled in output.

42.6.7. PIO Input Filter Enable Register

Name: PIO_IFER
Offset: 0x0020
Reset: –
Property: Write-only

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
	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 – P PIO Input Filter Enable

Value	Description
0	No effect.
1	Enables the input glitch filter on the I/O line.

42.6.8. PIO Input Filter Disable Register

Name: PIO_IFDR
Offset: 0x0024
Reset: –
Property: Write-only

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
	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 – P PIO Input Filter Disable

Value	Description
0	No effect.
1	Disables the input glitch filter on the I/O line.

42.6.9. PIO Input Filter Status Register

Name: PIO_IFSR
Offset: 0x0028
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 – P PIO Input Filter Status

Value	Description
0	The input glitch filter is disabled on the I/O line.
1	The input glitch filter is enabled on the I/O line.

42.6.10. PIO Set Output Data Register

Name: PIO_SODR
Offset: 0x0030
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 – P PIO Set Output Data

Value	Description
0	No effect.
1	Sets the data to be driven on the I/O line.

42.6.11. PIO Clear Output Data Register

Name: PIO_CODR
Offset: 0x0034
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 – P PIO Clear Output Data

Value	Description
0	No effect.
1	Clears the data to be driven on the I/O line.

42.6.12. PIO Output Data Status Register

Name: PIO_ODSR
Offset: 0x0038
Reset: –
Property: Read-only
or Read/Write

PIO_ODSR is Read-only or Read/Write depending on PIO_OWSR I/O lines.

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/W	R or R/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 – P PIO Output Data Status

Value	Description
0	The data to be driven on the I/O line is 0.
1	The data to be driven on the I/O line is 1.

42.6.13. PIO Pin Data Status Register

Name: PIO_PDSR
Offset: 0x003C
Property: Read-only

Reset values depend 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 – P PIO Output Data Status

Value	Description
0	The I/O line is at level 0.
1	The I/O line is at level 1.

42.6.14. PIO Interrupt Enable Register

Name: PIO_IER
Offset: 0x0040
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 – P PIO Input Change Interrupt Enable

Value	Description
0	No effect.
1	Enables the input change interrupt on the I/O line.

42.6.15. PIO Interrupt Disable Register

Name: PIO_IDR
Offset: 0x0044
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 – P PIO Input Change Interrupt Disable

Value	Description
0	No effect.
1	Disables the input change interrupt on the I/O line.

42.6.16. PIO Interrupt Mask Register

Name: PIO_IMR
Offset: 0x0048
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 – P PIO Input Change Interrupt Mask

Value	Description
0	Input change interrupt is disabled on the I/O line.
1	Input change interrupt is enabled on the I/O line.

42.6.17. PIO Interrupt Status Register

Name: PIO_ISR
Offset: 0x004C
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 – P PIO Input Change Interrupt Status

Value	Description
0	No input change has been detected on the I/O line since PIO_ISR was last read or since reset.
1	At least one input change has been detected on the I/O line since PIO_ISR was last read or since reset.

42.6.18. PIO Multi-driver Enable Register

Name: PIO_MDER
Offset: 0x0050
Reset: –
Property: Write-only

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
	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 – P PIO Multi-drive Enable

Value	Description
0	No effect.
1	Enables multi-drive on the I/O line.

42.6.19. PIO Multi-driver Disable Register

Name: PIO_MDDR
Offset: 0x0054
Reset: –
Property: Write-only

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
	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 – P PIO Multi-drive Disable

Value	Description
0	No effect.
1	Disables multi-drive on the I/O line.

42.6.20. PIO Multi-driver Status Register

Name: PIO_MDSR
Offset: 0x0058
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 – P PIO Multi-drive Status

Value	Description
0	The multi-drive is disabled on the I/O line. The pin is driven at high- and low-level.
1	The multi-drive is enabled on the I/O line. The pin is driven at low-level only.

42.6.21. PIO Pull-Up Disable Register

Name: PIO_PUDR
Offset: 0x0060
Reset: –
Property: Write-only

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
	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 – P PIO Pull-Up Disable

Value	Description
0	No effect.
1	Disables the pullup resistor on the I/O line.

42.6.22. PIO Pull-Up Enable Register

Name: PIO_PUER
Offset: 0x0064
Reset: –
Property: Write-only

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
	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 – P PIO Pull-Up Enable

Value	Description
0	No effect.
1	Enables the pullup resistor on the I/O line.

42.6.23. PIO Pull-Up Status Register

Name: PIO_PUSR
Offset: 0x0068
Property: Read-only

Reset values depend on the product implementation.

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 – P PIO Pull-Up Status

Value	Description
0	Pullup resistor is enabled on the I/O line.
1	Pullup resistor is disabled on the I/O line.

42.6.24. PIO Peripheral ABCD Select Register 1

Name: PIO_ABCDSR1
Offset: 0x0070
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
	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 – P PIO Peripheral Select

If the same bit is set to '0' in PIO_ABCDSR2:
0: Assigns the I/O line to the Peripheral A function.
1: Assigns the I/O line to the Peripheral B function.
If the same bit is set to '1' in PIO_ABCDSR2:
0: Assigns the I/O line to the Peripheral C function.
1: Assigns the I/O line to the Peripheral D function.

42.6.25. PIO Peripheral ABCD Select Register 2

Name: PIO_ABCDSR2
Offset: 0x0074
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
	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 – P PIO Peripheral Select

If the same bit is set to '0' in PIO_ABCDSR1:
0: Assigns the I/O line to the Peripheral A function.
1: Assigns the I/O line to the Peripheral C function.
If the same bit is set to '1' in PIO_ABCDSR1:
0: Assigns the I/O line to the Peripheral B function.
1: Assigns the I/O line to the Peripheral D function.

42.6.26. PIO Input Filter Slow Clock Disable Register

Name: PIO_IFSCDR
Offset: 0x0080
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 – P PIO Peripheral Clock Glitch Filtering Select

Value	Description
0	No effect.
1	The glitch filter is able to filter glitches with a duration < $t_{peripheral\ clock}/2$.

42.6.27. PIO Input Filter Slow Clock Enable Register

Name: PIO_IFSCER
Offset: 0x0084
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 – P PIO Slow Clock Debouncing Filtering Select

Value	Description
0	No effect.
1	The debouncing filter is able to filter pulses with a duration < $t_{div\_slck}/2$.

42.6.28. PIO Input Filter Slow Clock Status Register

Name: PIO_IFSCSR
Offset: 0x0088
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 – P PIO Glitch or Debouncing Filter Selection Status

Value	Description
0	The glitch filter is able to filter glitches with a duration < $t_{peripheral\ clock}/2$.
1	The debouncing filter is able to filter pulses with a duration < $t_{div\_slck}/2$.

42.6.29. PIO Slow Clock Divider Debouncing Register

Name: PIO_SCDR
Offset: 0x008C
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
			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

$t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$

42.6.30. PIO Pad Pull-Down Disable Register

Name: PIO_PPDDR
Offset: 0x0090
Reset: –
Property: Write-only

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
	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 – P PIO Pull-Down Disable

Value	Description
0	No effect.
1	Disables the pull-down resistor on the I/O line.

42.6.31. PIO Pad Pull-Down Enable Register

Name: PIO_PPDER
Offset: 0x0094
Reset: –
Property: Write-only

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
	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 – P PIO Pull-Down Enable

Value	Description
0	No effect.
1	Enables the pull-down resistor on the I/O line.

42.6.32. PIO Pad Pull-Down Status Register

Name: PIO_PPDSR
Offset: 0x0098
Property: Read-only

Reset values depend on the product implementation.

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 – P PIO Pull-Down Status

Value	Description
0	Pull-down resistor is enabled on the I/O line.
1	Pull-down resistor is disabled on the I/O line.

42.6.33. PIO Output Write Enable Register

Name: PIO_OWER
Offset: 0x00A0
Reset: –
Property: Write-only

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
	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 – P PIO Output Write Enable

Value	Description
0	No effect.
1	Enables writing PIO_ODSR for the I/O line.

42.6.34. PIO Output Write Disable Register

Name: PIO_OWDR
Offset: 0x00A4
Reset: –
Property: Write-only

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
	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 – P PIO Output Write Disable

Value	Description
0	No effect.
1	Disables writing PIO_ODSR for the I/O line.

42.6.35. PIO Output Write Status Register

Name: PIO_OWSR
Offset: 0x00A8
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 – P PIO Output Write Status

Value	Description
0	Writing PIO_ODSR does not affect the I/O line.
1	Writing PIO_ODSR affects the I/O line.

42.6.36. PIO Additional Interrupt Modes Enable Register

Name: PIO_AIMER
Offset: 0x00B0
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 – P PIO Additional Interrupt Modes Enable

Value	Description
0	No effect.
1	The interrupt source is the event described in PIO_ELSR and PIO_FRLHSR.

42.6.37. PIO Additional Interrupt Modes Disable Register

Name: PIO_AIMDR
Offset: 0x00B4
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 – P PIO Additional Interrupt Modes Disable

Value	Description
0	No effect.
1	The Interrupt mode is set to the default Interrupt mode (Both-edge Detection).

42.6.38. PIO Additional Interrupt Modes Mask Register

Name: PIO_AIMMR
Offset: 0x00B8
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 – P PIO I/O Line Index

Selects the I/O event type triggering an interrupt.

Value	Description
0	The interrupt source is a both-edge detection event.
1	The interrupt source is described by the registers PIO_ELSR and PIO_FRLHSR.

42.6.39. PIO Edge Select Register

Name: PIO_ESR
Offset: 0x00C0
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 – P PIO Edge Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is an edge-detection event.

42.6.40. PIO Level Select Register

Name: PIO_LSR
Offset: 0x00C4
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 – P PIO Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is a level-detection event.

42.6.41. PIO Edge/Level Status Register

Name: PIO_ELSR
Offset: 0x00C8
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 – P PIO Edge/Level Interrupt Source Selection

Value	Description
0	The interrupt source is an edge-detection event.
1	The interrupt source is a level-detection event.

42.6.42. PIO Falling Edge/Low-Level Select Register

Name: PIO_FELLSR
Offset: 0x00D0
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 – P PIO Falling Edge/Low-Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is set to a falling edge detection or low-level detection event, depending on PIO_ELSR.

42.6.43. PIO Rising Edge/High-Level Select Register

Name: PIO_REHLSR
Offset: 0x00D4
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 – P PIO Rising Edge/High-Level Interrupt Selection

Value	Description
0	No effect.
1	The interrupt source is set to a rising edge detection or high-level detection event, depending on PIO_ELSR.

42.6.44. PIO Fall/Rise - Low/High Status Register

Name: PIO_FRLHSR
Offset: 0x00D8
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 – P PIO Edge/Level Interrupt Source Selection

Value	Description
0	The interrupt source is a falling edge detection (if PIO_ELSR = 0) or low-level detection event (if PIO_ELSR = 1).
1	The interrupt source is a rising edge detection (if PIO_ELSR = 0) or high-level detection event (if PIO_ELSR = 1).

42.6.45. PIO Write Protection Mode Register

Name: PIO_WPMR
Offset: 0x00E4
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
0x50494F	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 0x50494F (“PIO” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x50494F (“PIO” in ASCII).

42.6.46. PIO Write Protection Status Register

Name: PIO_WPSR
Offset: 0x00E8
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.

42.6.47. PIO Schmitt Trigger Register

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

Bit	31	30	29	28	27	26	25	24
	SCHMITT31	SCHMITT30	SCHMITT29	SCHMITT28	SCHMITT27	SCHMITT26	SCHMITT25	SCHMITT24
Access	R/W	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
	SCHMITT23	SCHMITT22	SCHMITT21	SCHMITT20	SCHMITT19	SCHMITT18	SCHMITT17	SCHMITT16
Access	R/W	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
	SCHMITT15	SCHMITT14	SCHMITT13	SCHMITT12	SCHMITT11	SCHMITT10	SCHMITT9	SCHMITT8
Access	R/W	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
	SCHMITT7	SCHMITT6	SCHMITT5	SCHMITT4	SCHMITT3	SCHMITT2	SCHMITT1	SCHMITT0
Access	R/W	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 – SCHMITTx PIO Schmitt Trigger Control

Value	Description
0	Schmitt trigger is enabled.
1	Schmitt trigger is disabled.

42.6.48. PIO I/O Slewrate Control Register

Name: PIO_SLEWR
Offset: 0x0110
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	SR31	SR30	SR29	SR28	SR27	SR26	SR25	SR24
Access	R/W	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
	SR23	SR22	SR21	SR20	SR19	SR18	SR17	SR16
Access	R/W	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
	SR15	SR14	SR13	SR12	SR11	SR10	SR9	SR8
Access	R/W	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
	SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Access	R/W	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 – SRx Slewrate Control for IO line x

Refer to section “Electrical characteristics” for recommended usage of this bit.

Value	Name	Description
0	DISABLED	No slewrate control
1	ENABLED	Slewrate controlled

42.6.49. PIO I/O Drive Register 1

Name: PIO_DRIVER1
Offset: 0x0118
Reset: 0x00000000
Property: Read/Write

Refer to section “Electrical characteristics” for recommended usage of the following bits.

Bit	31	30	29	28	27	26	25	24
	DR31	DR30	DR29	DR28	DR27	DR26	DR25	DR24
Access	R/W	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
	DR23	DR22	DR21	DR20	DR19	DR18	DR17	DR16
Access	R/W	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
	DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8
Access	R/W	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
	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0
Access	R/W	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 – DRx Drive of PIO Line

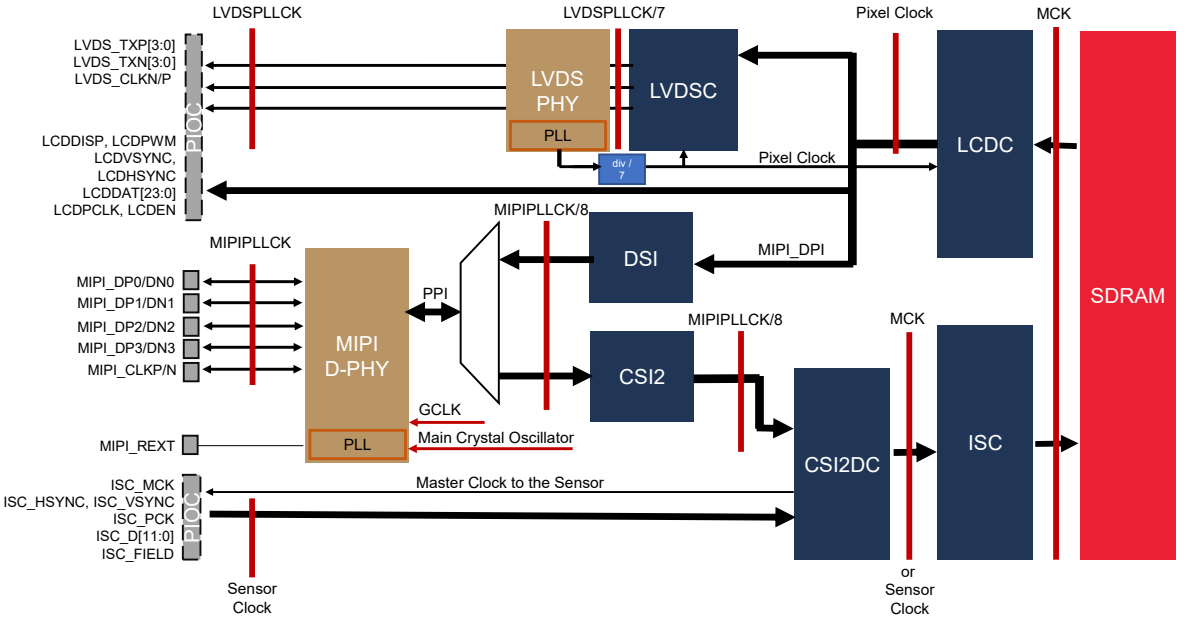
Value	Name	Description
0	LOW_DRIVE	Lowest drive
1	HIGH_DRIVE	Highest drive

Image Subsystem

43. Overview

43.1. Block Diagram

Figure 43.1. Image Subsystem Block Diagram



43.2. Components

The image subsystem is composed of two sections: Image and Display, made up of MIPI and RGB parallel interfaces. Their components are:

- Image capture:
  - Image Sensor Controller (ISC)
  - CSI-2 Demultiplexer Controller (CSI2DC)
  - Camera Sensor Host Interface (CSI2)
  - Physical interface to connect CSI-2 camera sensors (D-PHY)
- Display:
  - LCD Controller with parallel interface (LCDC)
  - Display serial interface with MIPIPLL (DSI)
  - LVDS Controller with LVDSPLL (LVDSC)
  - 2D Graphical Engine (GFX2D)

43.3. Product Dependencies

43.3.1. Clocks

In addition to MCK fed by PLLA:

- CSI2 has a GCLK input.
- DSI embeds a PLL, MIPIPLL, configurable using the DSI interface.
- LVDSC embeds a PLL controlled in the Power Management Controller (PMC).

43.3.2. Interrupts

Refer to the table Peripheral Identifiers.

43.3.3. Reset

Image peripherals are connected to the processor and peripherals reset line.

43.3.4. I/Os

- MIPI I/Os are high-speed I/Os powered by VDDMIPI/GND.
- ISC I/Os are multiplexed with other peripherals and powered by VDDLVDS or VDDIOP1.
- LVDS I/Os are multiplexed with other peripherals and powered by VDDLVDS.

Note: When the LVDS PHY is used, VDDLVDS must be connected to VDDOUT25.

43.3.5. 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 D-PHY:
   - 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 D-PHY 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.

43.4. LVDS I/O Configuration

43.4.1. Pre-Emphasis

A pre-emphasis booster is a symmetrical switch cap circuit whose ideal gain is determined by the ratio of charges injected/subtracted to/from output nodes versus the reference charges stored in the output load.

When operated, this circuit provides a faster rise/fall transition time to help improve eye diagram aperture – without affecting the DC operating point in the output load.

Capacitors are programmable via the 3-bit trimming field in the LVDSC_ACR register, independently for each data and clock lane.

43.4.2. Common and Bias

The LVDS common has a 5-bit trimmable current source used to bias analog lanes and adjust the LVDS DC output signal amplitude. This is programmed using the LVDSC_ACR register, with the field DCBIAS.

43.5. Special Functions in SFR

Select the MIPI direction (TX or RX) using the SFR_ISS_CFG.MODE bit.

Program 0 for CSI2, or 1 for DSI.

44. LCD Controller (LCDC)

44.1. Description

The LCD Controller (LCDC) consists of logic for transferring LCD image data from an external display buffer to a TFT LCD panel. The LCDC has one display input buffer per layer that fetches pixels through the single bus host interface and a look-up table to allow palletized display configurations. The LCDC is programmable on a per layer basis, and supports different LCD resolutions, window sizes, image formats and pixel depths.

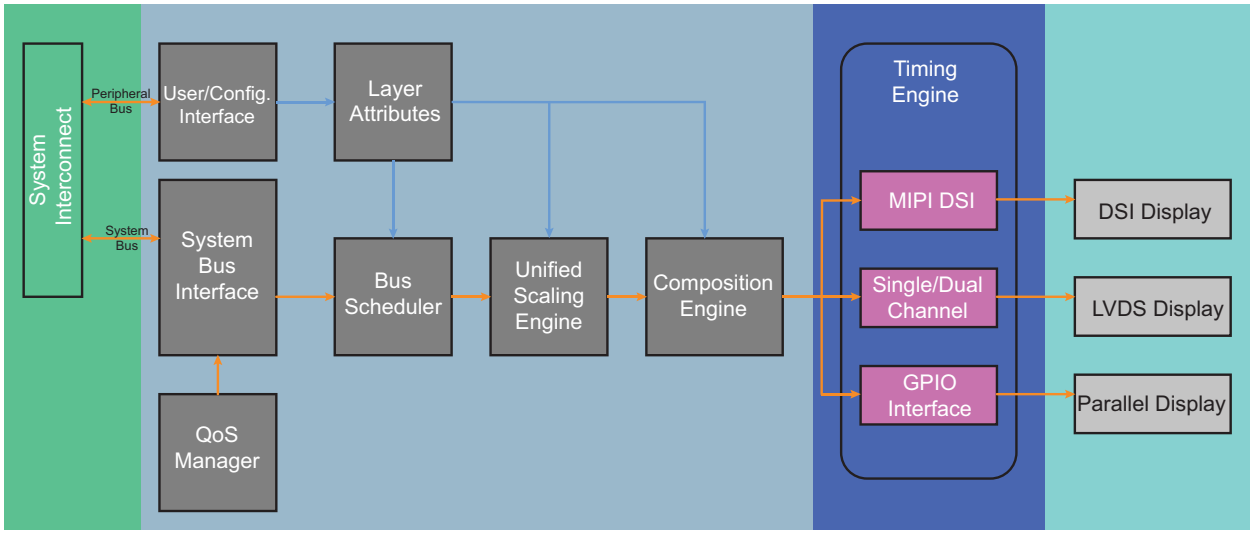
The LCDC is connected to the system bus as a host for reading pixel data. It also integrates a peripheral bus interface to configure its registers.

44.2. Embedded Characteristics

- Up to 32-bit Host Interface
- Supports Single Scan Active TFT Display
- Supports Parallel RGB, MIPI DSI and Single Channel LVDS Interfaces
- Supports 12-, 16-, 18- and 24-bit Output Mode
- Supports Spatial Dithering for 12-, 16-, and 18-bit Output Mode
- Asynchronous Output Mode Supported
- 1, 2, 4, 8 bits per Pixel (Palletized)
- 12, 16, 18, 19, 24, 25 and 32 bits per Pixel (Non-palletized)
- Supports One Base Layer (Background)
- Supports Overlay 1 Layer
- Supports Overlay 2 Layer
- Supports High-End Overlay (HEO) Layer
- High-End Overlay Supports 4:2:0 Planar Mode and Semiplanar Mode
- High-End Overlay Supports 4:2:2 Planar Mode, Semiplanar Mode and Packed
- Little Endian Memory Organization
- Programmable Timing Engine, with Integer Clock Divider
- Programmable Polarity for Data, Line Synchro and Frame Synchro
- Up to 1024x768 (XGA) with Overlay (Application-Dependent). Still Image up to 1280x720 (720p)
- Color Look-up Table (CLUT) with up to 256 Entries and Predefined 8-bit Alpha
- Gamma Correction through Color Look-Up Table (CLUT)
- Programmable Negative and Positive Row Striding for all Layers
- Horizontal and Vertical Rescaling Unit with Edge Interpolation and Independent Non-Integer Ratio, up to 1280x720 (720p)
- Bandwidth and Power Optimization with Hidden Section of Base Layer
- Integrates Fully Programmable Color Space Conversion
- Blender Function Supports Arbitrary 8-bit Alpha Value and Chroma Keying

44.3. Block Diagram

Figure 44.1. Block Diagram



44.4. I/O Lines Description

Table 44.1. I/O Lines Description

Name	Description	Type
LCDC_PWM	Contrast control signal, using Pulse Width Modulation	Output
LCDC_HSYNC	Horizontal Synchronization Pulse	Output
LCDC_VSYNC	Vertical Synchronization Pulse	Output
LCDC_DAT[23:0]	LCD 24-bit data bus	Output
LCDC_DEN	Data Enable	Output
LCDC_DISP	Display Enable signal	Output
LCDC_PCK	Pixel Clock	Output
MIPI_CLKP	MIPI D-PHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI D-PHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI D-PHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_DP2	MIPI D-PHY differential output data lane 2	Input/Output
MIPI_DN2		
MIPI_DP3	MIPI D-PHY differential output data lane 3	Input/Output
MIPI_DN3		
MIPI_REXT	Calibration reference resistor	Input/Output
LVDS_CLK1P	Differential LVDS clock line transceiver output	Output
LVDS_CLK1M		
LVDS_A0P	Differential LVDS data 0 line transceiver output	Output
LVDS_A0M		
LVDS_A1P	Differential LVDS data 1 line transceiver output	Output
LVDS_A1M		

Table 44.1. I/O Lines Description (continued)

Name	Description	Type
LVDS_A2P	Differential LVDS data 2 line transceiver output	Output
LVDS_A2M		
LVDS_A3P	Differential LVDS data 3 line transceiver output	Output
LVDS_A3M		

44.5. Product Dependencies

44.5.1. I/O Lines

The pins used for interfacing the LCDC 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 LCDC are not used by the application, they can be used for other purposes by the PIO Controller.

44.5.2. Power Management

The LCDC is not continuously clocked. Before using it, the user must first enable the LCDC peripheral and generic (GCLK) clocks in the Power Management Controller.

If the LVDS interface is used, once the LVDS PLL is enabled the pixel clock is used as the clock for LCDC and the GCLK is no longer needed.

44.5.3. Interrupt Sources

The LCDC interrupt line is connected to one of the internal sources of the interrupt controller. Using the LCDC interrupt requires prior programming of the interrupt controller (see Interrupt Software Operations).

44.6. Functional Description

The LCDC module integrates the following digital blocks:

- DMA Engine—this block performs data prefetch and requests access to the system bus interface.
- Input Layer FIFO—stores the stream of pixels
- Color Look-up Table (CLUT)—these 256 RAM-based look-up table entries are selected when the color depth is set to 1, 2, 4 or 8 bpp.
- Color Space Conversion (CSC)—changes the color space from YCbCr to RGB
- Two Dimension Scaler (2DSC)—resizes the image
- Global Alpha Blender (GAB)—performs programmable 256-level alpha blending
- Output FIFO—stores the blended pixel prior to display
- LCD Timing Engine—provides a fully programmable HSYNC-VSYNC interface

The DMA controller reads the image through the system bus host interface. The LCDC engine formats the display data, then the GAB performs alpha blending if required, and writes the final pixel into the output FIFO. The programmable timing engine drives a valid pixel onto the LCD output interface.

44.6.1. Timing Engine Configuration

Before writing in the Configuration registers, LCDC_LCDCFGx, of the timing engine, the bit SIPSTS in the Status register (LCDC_LCDSR) must be cleared.

44.6.1.1. Pixel Clock Period Configuration

The pixel clock (LCDC_PCK) generated by the timing engine is the source clock divided by the field CLKDIV in the Configuration register 0 (LCDC_LCDCFG0). The source clock is the GCLK clock.

The pixel clock period formula is:

$$\text{LCDC\_PCK} = \frac{\text{source clock}}{\text{CLKDIV}+2}$$

To obtain LCDC_PCK = source clock, the clock divider can be disabled by configuring LCDC_LCDCFG0.CLKBYP.

The pixel clock polarity can be programmed in LCDC_LCDCFG0.CLKPOL.

44.6.1.2. Horizontal and Vertical Synchronization Configuration

The following fields are used to configure the timing engine:

- LCDC_LCDCFG1.HSPW
- LCDC_LCDCFG1.VSPW
- LCDC_LCDCFG2.VFPW
- LCDC_LCDCFG2.VBPW
- LCDC_LCDCFG3.HFPW
- LCDC_LCDCFG3.HBPW
- LCDC_LCDCFG4.PPL
- LCDC_LCDCFG4.RPF

The polarity of output signals is also programmable.

44.6.1.3. Timing Engine Power-Up Software Operation

The following sequence enables the display:

1. Configure the LCD timing parameters, signal polarity and clock period in LCDC_LCDCFGx.
2. Enable the pixel clock by writing a one to CLKEN in the Enable register (LCDC_LCDEN).
3. Poll CLKSTS in the Status register (LCDC_LCDSR) to check that the clock is running.
4. Enable horizontal and vertical synchronization by writing a one to LCDC_LCDEN.SYNCEN.
5. Poll LCDC_LCDSR.LCDSTS to check that the synchronization is up.
6. Enable the display power signal by writing a one to LCDC_LCDEN.DISPEN.
7. Poll LCDC_LCDSR.DISPSTS to check that the power signal is activated.
8. If the MIPI output interface is selected and if the display supports the shutdown command, turn on the display by writing a one to LCDC_LCDEN.SDEN.
9. If the MIPI output interface is selected and if the display supports the shutdown command, poll LCDC_LCDSR.SDSTS to check that the shutdown signal output is zero.

LCDC_LCDCFG5.GUARDTIME is used to configure the number of frames before the assertion of the signal LCDC_DISP.

44.6.1.4. Timing Engine Power-Down Software Operation

The following sequence disables the display:

1. If the MIPI output interface is selected and if the display supports the shutdown command, turn off the display by writing a one to LCDC_LCDDIS.SDDIS.
2. If the MIPI output interface is selected and if the display supports the shutdown command, poll LCDC_LCDSR.SDSTS to check that the shutdown signal output is one.
3. Disable LCDC_DISP by writing DISPDIS in the Disable register (LCDC_LCDDIS).
4. Poll LCDC_LCDSR.DISPSTS to verify that LCDC_DISP is no longer enabled.
5. Disable the HSYNC and VSYNC signals by writing a one to LCDC_LCDDIS.SYNCDIS.
6. Poll LCDC_LCDSR.LCDSTS to check that the synchronization is off.
7. Disable the pixel clock by writing a one to LCDC_LCDDIS.CLKDIS.

44.6.2. Interrupt Software Operations

The LCDC includes two interrupt levels. Specific modules or layers have their own low-level interrupt vector (Base Layer, Overlay 1, Overlay 2, High-End Overlay, Write Protection). Each of their sources can be controlled by an Interrupt Enable register, Interrupt Disable register, and Interrupt Mask register, and can be monitored by an Interrupt Status register. The table below shows the different layers and the registers dedicated to interrupts.

Table 44.2. Low-level Interrupt Register List

Module	Interrupt Enable Register	Interrupt Disable Register	Interrupt Mask Register	Interrupt Status Register
Base Layer	LCDC_BASEIER	LCDC_BASEIDR	LCDC_BASEIMR	LCDC_BASEISR
Overlay 1	LCDC_OVR1IER	LCDC_OVR1IDR	LCDC_OVR1IMR	LCDC_OVR1ISR
Overlay 2	LCDC_OVR2IER	LCDC_OVR2IDR	LCDC_OVR2IMR	LCDC_OVR2ISR
High-End Overlay	LCDC_HEOIER	LCDC_HEOIDR	LCDC_HEOIMR	LCDC_HEOISR
Write Protection	–	–	–	LCDC_WPSR

Each low-level interrupt is summarized in a single source for the main-level Interrupt vector which is controlled by LCDC_LCDIER, LCDC_LCDIDR and LCDC_LCDIMR, and can be monitored with LCDC_LCDISR. It can be driven by the following sources:

- Start of frame event
- LCD disable terminated
- Display power-up/power-down terminated
- Configurable active row number event
- Output FIFO underflow
- Base Layer interrupt
- Overlay 1 interrupt
- Overlay 2 interrupt
- High-End Overlay interrupt
- Write Protection interrupt

If any low-level interrupt is triggered, then the main-level interrupt is triggered as well if the source related to the corresponding module is enabled in LCDC_LCDIER. If any main-level interrupt is triggered, then the LCDC interrupt line is raised.

The sequence to clear a low-level interrupt is the following:

1. Read LCDC_LCDISR and identify the source module.
2. Read the corresponding low-level Interrupt Status register.
3. In case of persistent error, take action to stop the error source.
4. Re-read the corresponding low-level Interrupt Status register to clear it.

Once a low-level interrupt is cleared, the corresponding main-level source is automatically cleared.

44.6.3. DMA Software Operations

44.6.3.1. Enabling DMA Channels

Follow the steps below to enable DMA channels:

1. Check that all fields are cleared in the Attribute Status register (LCDC_ATTRS).
2. Write the Configuration registers of each layer.
3. Set the ENABLE field in the Enable register of each layer (LCDC_xxxEN) .

4. Write a one to the Update Attribute field for each layer and, if CLUT mode is used and if the CLUT content is fetched by the DMA, write a one to the Update Color LUT field for each layer in the Attribute Enable register (LCDC_ATTRE).

44.6.3.2. Disabling DMA Channels

Follow the steps below to disable DMA channels:

1. Check that all fields are cleared in LCDC_ATTRS.
2. Clear the ENABLE field in the Enable registers of each layer (LCDC_xxxEN).
3. Write a one to the Update Attribute field for each layer in LCDC_ATTRE.

44.6.3.3. Modifying the Layer Attributes

Follow the steps below to modify layer attributes:

1. Check that all fields are cleared in LCDC_ATTRS.
2. Write the desired attribute fields in the Configuration registers, Frame Buffer registers and Enable registers of each layer.
3. Write a one to the Update Attribute field for each layer and, if required, write a one to the Update Color LUT field for each layer in LCDC_ATTRE.

44.6.3.4. DMA Interrupt Generation

The DMA Controller operation sets the following interrupt flags in the layer Interrupt Status registers:

- END—indicates that the current frame DMA transfer is completed.
- ERROR—indicates that a bus transfer error has been detected.
- OVF—indicates that a DMA transfer is still pending at the end of frame.

44.6.3.5. Frame Buffer DMA Address Alignment Requirements

When programming the Frame Buffer registers of each layer (LCDC_xxxFBA), the following requirements must be met:

Table 44.3. Frame Buffer DMA Address Alignment when CLUT Mode is Selected

CLUT Mode	DMA Address Alignment
1 bpp	8 bits
2 bpp	8 bits
4 bpp	8 bits
8 bpp	8 bits

Table 44.4. Frame Buffer DMA Address Alignment when RGB Mode is Selected

RGB Mode	DMA Address Alignment
12 bpp RGB 444	16 bits
16 bpp ARGB 4444	16 bits
16 bpp RGBA 4444	16 bits
16 bpp RGB 565	16 bits
16 bpp ARGB 1555	16 bits
18 bpp RGB 666	32 bits
18 bpp RGB 666 PACKED	8 bits
19 bpp ARGB 1666	32 bits
19 bpp ARGB 1666 PACKED	8 bits
24 bpp RGB 888	32 bits
24 bpp RGB 888 PACKED	8 bits

Table 44.4. Frame Buffer DMA Address Alignment when RGB Mode is Selected (continued)

RGB Mode	DMA Address Alignment
25 bpp ARGB 1888	32 bits
32 bpp ARGB 8888	32 bits
32 bpp RGBA 8888	32 bits

Table 44.5. Frame Buffer DMA Address Alignment when YCbCr Mode is Selected

YCbCr Mode	DMA Address Alignment
32 bpp AYCbCr	32 bits
16 bpp YCbCr 4:2:2	32 bits
16 bpp semiplanar YCbCr 4:2:2	Y 8 bits
	CbCr 16 bits
16 bpp planar YCbCr 4:2:2	Y 8 bits
	Cb 8 bits
	Cr 8 bits
12 bpp semiplanar YCbCr 4:2:0	Y 8 bits
	CbCr 16 bits
12 bpp planar YCbCr 4:2:0	Y 8 bits
	Cb 8 bits
	Cr 8 bits

44.6.4. Layer Software Configuration

44.6.4.1. System Bus Access Attribute

This attribute is defined to improve bandwidth of the layer.

- BLEN field—Defines the maximum burst length of the DMA channel.

44.6.4.2. Color Attributes

- CLUTMODE field—Selects the CLUT mode.
- RGBMODE field—Selects the RGB mode.
- GAM field—Activates gamma correction when true RGB color is activated.
- YCCMODE field—Selects the Luma Chroma mode.

44.6.4.3. Color Look-Up Table Configuration

For each layer, the LCDC includes four 256-entry color look-up tables (CLUT) that can be used either for indexed color formats (up to 8bpp) or for gamma correction. One CLUT is dedicated to each pixel component (A, R, G and B).

When indexed color format is used, the frame buffer contains the CLUT read address used to recall the corresponding real ARGB pixel value previously stored.

For more details about gamma correction, see Gamma Correction.

There are two ways to load CLUT values:

1. through the User Interface
2. by DMA access

From the User Interface, the CLUT can be accessed directly at the corresponding address space, but only when the targeted layer CLUT and gamma correction are disabled (see LCDC_BASECLA, LCDC_OVR1CLA, LCDC_OVR2CLA, LCDC_HEOCLA). The CLUT content can also be fetched by the DMA at the next start of frame, before fetching the frame buffer. The following sequence is used to update the CLUT:

1. Write the CLUT content(s) in memory.
2. Configure the CLA field in the Color Look-Up Table Address registers of each layer (LCDC_xxxCLA).
3. Check that all fields in LCDC_ATTRS are cleared.
4. Write a one to the Update Color LUT field for each layer in LCDC_ATTRE.

44.6.4.4. Window Position, Size, Scaling and Striding Attributes

- XPOS and YPOS fields—Defines the position of the layer window.
- XSIZE and YSIZE fields—Defines the size of the displayed window.
- XMEMSIZE and YMEMSIZE fields—Defines the size of the image frame buffer.
- XSTRIDE —Defines the line striding.
- PSTRIDE —Defines the pixel striding.
- VXSYFACTOR and VXSCFACTOR fields—Defines the scaling ratio for the vertical scaler.
- HXSYFACTOR and HXSCFACTOR fields—Defines the scaling ratio for the horizontal scaler.

The position and size attributes are programmed to keep the window within the display area.

When CLUT mode is enabled, the restrictions detailed in the following table apply on the horizontal, vertical window sizes and discard attributes (Discard Area Horizontal Position, Discard Area Vertical Position, Discard Area Horizontal Size, Discard Area Vertical Size).

Table 44.6. CLUT Mode and Window Size/Discard Attributes

CLUT Mode	X-Y Size/Discard Attributes Requirement
1 bpp	Multiple of 8 pixels
2 bpp	Multiple of 4 pixels
4 bpp	Multiple of 2 pixels
8 bpp	Free size

Pixel striding is disabled when CLUT mode is enabled.

Depending on the input mode, the restrictions detailed in the following table apply on the window size.

Table 44.7. Window Size

Mode	X-Y Requirement, Scaling Turned Off	X-Y Requirement, Scaling Turned On	XMEM_SIZE-YMEM_SIZE Requirement, Scaling Turned On
ARGB/CLUT	Free size	XSIZE ≥ 3 pixels YSIZE ≥ 3 pixels	XMEM_SIZE ≥ 3 pixels YMEM_SIZE ≥ 3 pixels
AYCbCr 4:4:4	Free size	XSIZE ≥ 3 pixels YSIZE ≥ 3 pixels	XMEM_SIZE ≥ 3 pixels YMEM_SIZE ≥ 3 pixels
YCbCr 4:2:2 packed	XSIZE ≥ 6 pixels Free YSIZE	XSIZE ≥ 6 pixels YSIZE ≥ 3 pixels	XMEM_SIZE ≥ 6 pixels, even YMEM_SIZE ≥ 3 pixels
YCbCr 4:2:2 semiplanar	XSIZE ≥ 6 pixels Free YSIZE	XSIZE ≥ 6 pixels YSIZE ≥ 3 pixels	XMEM_SIZE ≥ 6 pixels YMEM_SIZE ≥ 3 pixels
YCbCr 4:2:2 planar	XSIZE ≥ 6 pixels Free YSIZE	XSIZE ≥ 6 pixels YSIZE ≥ 3 pixels	XMEM_SIZE ≥ 6 pixels YMEM_SIZE ≥ 3 pixels
YCbCr 4:2:0 semiplanar	XSIZE ≥ 6 pixels YSIZE ≥ 6 pixels	XSIZE ≥ 6 pixels YSIZE ≥ 6 pixels	XMEM_SIZE ≥ 6 pixels YMEM_SIZE ≥ 6 pixels
YCbCr 4:2:0 planar	XSIZE ≥ 6 pixels YSIZE ≥ 6 pixels	XSIZE ≥ 6 pixels YSIZE ≥ 6 pixels	XMEM_SIZE ≥ 6 pixels YMEM_SIZE ≥ 6 pixels

44.6.4.5.Overlay Blender Attributes

When two or more video layers are used, alpha blending is performed to define the final image displayed. Each window has its own blending attributes.

- CRKEY bit—Enables the chroma keying and match logic
- DSTKEY bit—When written to one, destination keying is enabled
- SFACTC—Source factor for the color channel blending equation
- SFACTA—Source factor for the Alpha channel blending equation
- DFACTC—Destination factor for the color channel blending equation
- DFACTA—Destination factor
- A0 field—Defines the source window alpha value 0
- A1 field—Defines the source window alpha value 1

44.6.4.6.Layer Attributes Update Operation

A set of attribute configuration registers (LCDC_xxxCFGx, LCDC_xxxFBA and LCDC_xxxEN, where xxx indicates the layer) is dedicated to each layer and defines attributes specific to that layer. These registers can be updated by using the following procedure:

1. Check that all fields in LCDC_ATTRS are cleared.
2. Write the attribute configuration registers of each layer.
3. Write a one to the Update Attribute field for each layer in LCDC_ATTRE.

44.6.4.7.Bandwidth and Power Optimization with Hidden Section of Base Layer

When the base layer is combined with at least one active overlay (100% opacity overlay), by default, the whole base layer frame is retrieved from the memory though it is not visible.

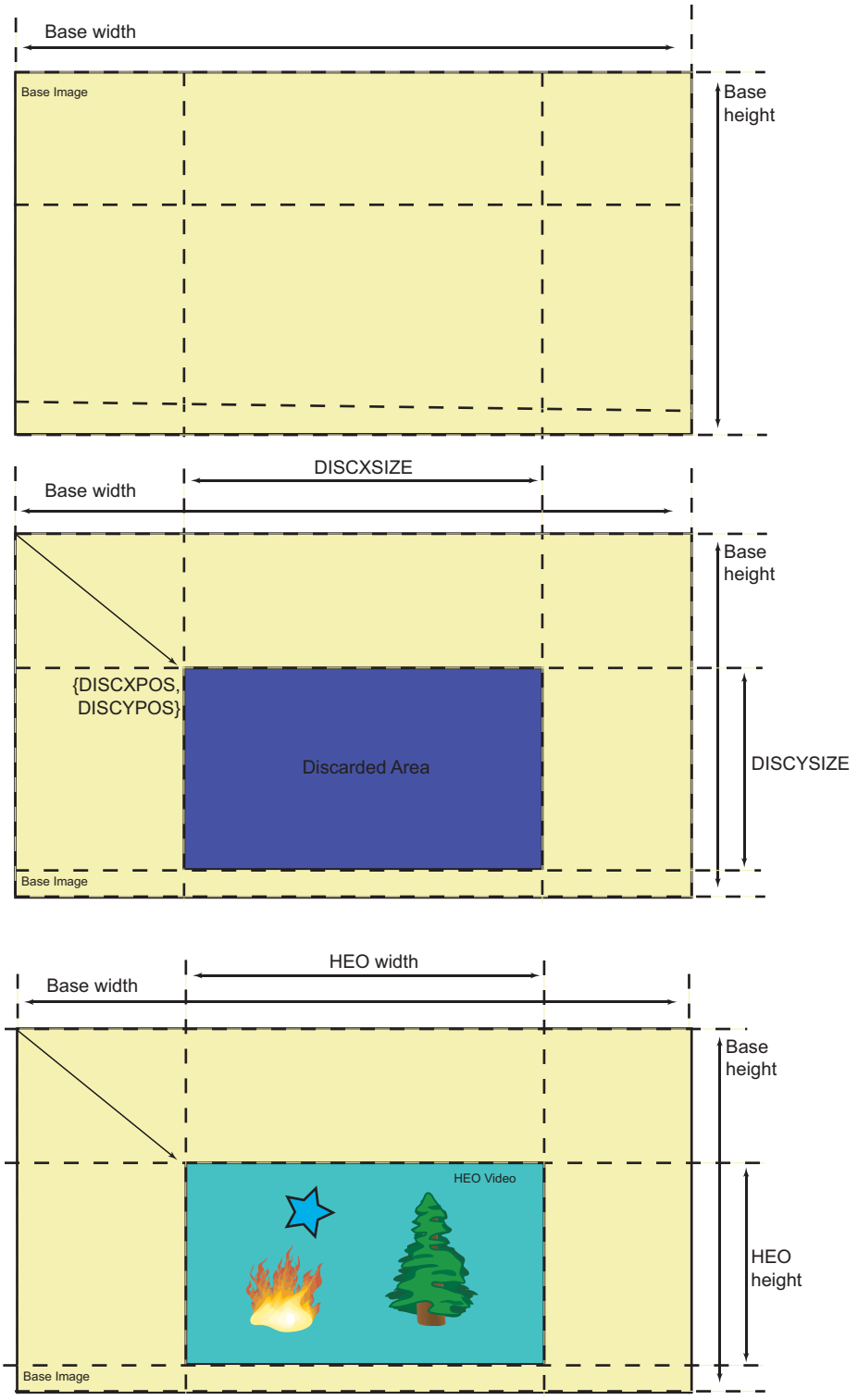
To optimize the system bandwidth, the LCDC can be configured to prevent the unuseful data from being fetched from system memory.

The following registers are used to disable an invisible area of the base layer:

- LCDC_BASECFG5:
  - field DISCXPOS (Discard Area Horizontal Position)
  - field DISCYPOS (Discard Area Vertical Position)
- LCDC_BASECFG6:
  - field DISCXSIZE (Discard Area Horizontal Size)
  - field DISCYSIZE (Discard Area Vertical Size)
- LCDC_BASECFG4: bit DISCEN (Discard Area Enable)

Each time the overlay window is resized and/or moved, these configuration registers must be reconfigured according to the new overlay window features.

Figure 44.2. Base Layer Discard Area



44.6.5. RGB Frame Buffer Memory Bitmap

44.6.5.1.1 bpp Through Color Look-up Table

Table 44.8. 1 bpp Memory Mapping, Little Endian Organization

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
Pixel 1 bpp	p31	p30	p29	p28	p27	p26	p25	p24	p23	p22	p21	p20	p19	p18	p17	p16	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0

44.6.5.2.2 bpp Through Color Look-up Table

Table 44.9. 2 bpp Memory Mapping, Little Endian Organization

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
Pixel 2 bpp	p15		p14		p13		p12		p11		p10		p9		p8		p7		p6		p5		p4		p3		p2		p1		p0	

44.6.5.3.4 bpp Through Color Look-up Table

Table 44.10. 4 bpp Memory Mapping, Little Endian Organization

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
Pixel 4 bpp	p7				p6				p5				p4				p3				p2				p1				p0			

44.6.5.4.8 bpp Through Color Look-up Table

Table 44.11. 8 bpp Memory Mapping, Little Endian Organization

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
Pixel 8 bpp	p3								p2								p1								p0							

44.6.5.5.12 bpp Memory Mapping, RGB 4:4:4

Table 44.12. 12 bpp Memory Mapping, Little Endian Organization

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
Pixel 12 bpp	–				R1[3:0]				G1[3:0]				B1[3:0]				–				R0[3:0]				G0[3:0]				B0[3:0]			

44.6.5.6.16 bpp Memory Mapping with Alpha Channel, ARGB 4:4:4:4

Table 44.13. 16 bpp Memory Mapping, Little Endian Organization

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
Pixel 16 bpp	A1[3:0]				R1[3:0]				G1[3:0]				B1[3:0]				A0[3:0]				R0[3:0]				G0[3:0]				B0[3:0]			

44.6.5.7.16 bpp Memory Mapping with Alpha Channel, RGBA 4:4:4:4

Table 44.14. 16 bpp Memory Mapping, Little Endian Organization

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
Pixel 16 bpp	R1[3:0]				G13:0]				B1[3:0]				A1[3:0]				R0[3:0]				G0[3:0]				B0[3:0]				A0[3:0]			

44.6.5.8.16 bpp Memory Mapping with Alpha Channel, RGB 5:6:5

Table 44.15. 16 bpp Memory Mapping, Little Endian Organization

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

Table 44.15. 16 bpp Memory Mapping, Little Endian Organization (continued)

Mem addr	0x3	0x2	0x1	0x0
Pixel 16bpp	R1[4:0] G1[5:0]	B1[4:0]	R0[4:0]	G0[5:0] B0[4:0]

44.6.5.9.16 bpp Memory Mapping with Transparency Bit, ARGB 1:5:5:5

Table 44.16. 16 bpp Memory Mapping, Little Endian Organization

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
Pixel 4 bpp	A1	R1[4:0]							G1[4:0]								B1[4:0]								A0	R0[4:0]						
	G0[4:0]								B0[4:0]																							

44.6.5.10.18 bpp Unpacked Memory Mapping, RGB 6:6:6

Table 44.17. 18 bpp Unpacked Memory Mapping, Little Endian Organization

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
Pixel 18 bpp	–								–								R0[5:0]								G0[5:0]							
	B0[5:0]																															

44.6.5.11.18 bpp Packed Memory Mapping, RGB 6:6:6

Table 44.18. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

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
Pixel 18 bpp	G1[1:0]		B1[5:0]						–								R0[5:0]								G0[5:0]							
	B0[5:0]																															

Table 44.19. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 18 bpp	R2[3:0]				G2[5:0]								B2[5:0]								–				R1[5:0]							
	G1[5:2]																															

Table 44.20. 18 bpp Packed Memory Mapping, Little Endian Organization at Address 0x8, 0x9, 0xA, 0xB

Mem addr	0xB								0xA								0x9								0x8							
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
Pixel 18 bpp	–								R[5:0]								G3[5:0]								B3[5:0]							
	–								R2[5:4]																							

44.6.5.12.19 bpp Unpacked Memory Mapping with Transparency Bit, ARGB 1:6:6:6

Table 44.21. 19 bpp Unpacked Memory Mapping, Little Endian Organization

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
Pixel 19 bpp	–								–					A0			R0[5:0]								G0[5:0]							
	B0[5:0]																															

44.6.5.13.19 bpp Packed Memory Mapping with Transparency Bit, ARGB 1:6:6:6

Table 44.22. 19 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

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
Pixel 19 bpp	G1[1:0]		B1[5:0]						–					A0			R0[5:0]								G0[5:0]							
	B0[5:0]																															

Table 44.23. 19 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 19 bpp	R2[3:0]				G2[5:0]								B2[5:0]								–			A1	R1[5:0]							
	G1[5:2]																															

Table 44.24. 19 bpp Packed Memory Mapping, Little Endian Organization at Address 0x8, 0x9, 0xA, 0xB

Mem addr	0xB								0xA								0x9								0x8							
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
Pixel 19 bpp	–					A3			R3[5:0]								G3[5:0]								B3[5:0]							
	–								A2	R2[5:4]																						

44.6.5.14.24 bpp Unpacked Memory Mapping, RGB 8:8:8

Table 44.25. 24 bpp Memory Mapping, Little Endian Organization

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
Pixel 24 bpp	–								R0[7:0]								G0[7:0]								B0[7:0]							

44.6.5.15.24 bpp Packed Memory Mapping, RGB 8:8:8

Table 44.26. 24 bpp Packed Memory Mapping, Little Endian Organization at Address 0x0, 0x1, 0x2, 0x3

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
Pixel 24 bpp	B1[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

Table 44.27. 24 bpp Packed Memory Mapping, Little Endian Organization at Address 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 24 bpp	G2[7:0]								B2[7:0]								R1[7:0]								G1[7:0]							

44.6.5.16.25 bpp Memory Mapping with Transparency Bit, ARGB 1:8:8:8

Table 44.28. 25 bpp Memory Mapping, Little Endian Organization

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
Pixel 25 bpp	–							A0	R0[7:0]								G0[7:0]								B0[7:0]							

44.6.5.17.32 bpp Memory Mapping, ARGB 8:8:8:8

Table 44.29. 32 bpp Memory Mapping, Little Endian Organization

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
Pixel 32 bpp	A0[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

44.6.5.18.32 bpp Memory Mapping, RGBA 8:8:8:8

Table 44.30. 32 bpp Memory Mapping, Little Endian Organization

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
Pixel 32 bpp	R0[7:0]								G0[7:0]								B0[7:0]								A0[7:0]							

44.6.6. YCbCr Frame Buffer Memory Mapping

44.6.6.1.AYCbCr 4:4:4 Packed Frame Buffer Memory Mapping

Table 44.31. 32 bpp Memory Mapping, Little Endian Organization

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
Pixel 16 bpp	A0[7:0]								Y0[7:0]								Cb0[7:0]								Cr0[7:0]							

44.6.6.2.4:2:2 Packed Mode Frame Buffer Memory Mapping

Table 44.32. 16 bpp 4:2:2 Packed Mode 0

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
Pixel 16 bpp	Cr0[7:0]								Y1[7:0]								Cb0[7:0]								Y0[7:0]							

Table 44.33. 16 bpp 4:2:2 Packed Mode 1

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
Pixel 16 bpp	Y1[7:0]								Cr0[7:0]								Y0[7:0]								Cb0[7:0]							

Table 44.34. 16 bpp 4:2:2 Packed Mode 2

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
Pixel 16 bpp	Cb0[7:0]								Y1[7:0]								Cr0[7:0]								Y0[7:0]							

Table 44.35. 16 bpp 4:2:2 Packed Mode 3

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
Pixel 16 bpp	Y1[7:0]								Cb0[7:0]								Y0[7:0]								Cr0[7:0]							

44.6.6.3.4:2:2 Semiplanar Mode Frame Buffer Memory Mapping

Table 44.36. 4:2:2 Semiplanar Luma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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
Pixel 16 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 44.37. 4:2:2 Semiplanar Chroma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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
Pixel 16 bpp	Cb2[7:0]								Cr2[7:0]								Cb0[7:0]								Cr0[7:0]							

44.6.6.4.4:2:2 Planar Mode Frame Buffer Memory Mapping

Table 44.38. 4:2:2 Planar Mode Luma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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
Pixel 16 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 44.39. 4:2:2 Planar Mode Chroma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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
Pixel 16 bpp	C3[7:0] C3=Cr/Cb								C2[7:0] C2=Cr/Cb								C1[7:0] C1=Cr/Cb								C0[7:0] C0=Cr/Cb							

44.6.6.5.4:2:0 Planar Mode Frame Buffer Memory Mapping

In Planar mode, the three video components Y, Cr and Cb are split into three memory areas and stored in a raster-scan order. These three memory planes are always aligned on a 32-bit boundary.

Table 44.40. 4:2:0 Planar Mode Luma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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
Pixel 12 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 44.41. 4:2:0 Planar Mode Luma Memory Mapping, Little Endian Organization for Byte 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 12 bpp	Y7[7:0]								Y6[7:0]								Y5[7:0]								Y4[7:0]							

Table 44.42. 4:2:0 Planar Mode Chroma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3

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

Table 44.42. 4:2:0 Planar Mode Chroma Memory Mapping, Little Endian Organization for Byte 0x0, 0x1, 0x2, 0x3 (continued)

Mem addr	0x3	0x2	0x1	0x0
Pixel 12 bpp	C3[7:0] C3=Cr/Cb	C2[7:0] C2=Cr/Cb	C1[7:0] C1=Cr/Cb	C0[7:0] C0=Cr/Cb

Table 44.43. 4:2:0 Planar Mode Chroma Memory Mapping, Little Endian Organization for Byte 0x4, 0x5, 0x6, 0x7

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 12 bpp	C7[7:0] C7=Cr/Cb								C6:[7:0] C6=Cr/Cb								C5[7:0] C5=Cr/Cb								C4[7:0] C4=Cr/Cb							

44.6.6.6.4:2:0 Semiplanar Frame Buffer Memory Mapping

Table 44.44. 4:2:0 Semiplanar Mode Luma Memory Mapping, Little Endian Organization

Mem addr	0x7								0x6								0x5								0x4							
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
Pixel 12 bpp	Y3[7:0]								Y2[7:0]								Y1[7:0]								Y0[7:0]							

Table 44.45. 4:2:0 Semiplanar Mode Chroma Memory Mapping, Little Endian Organization

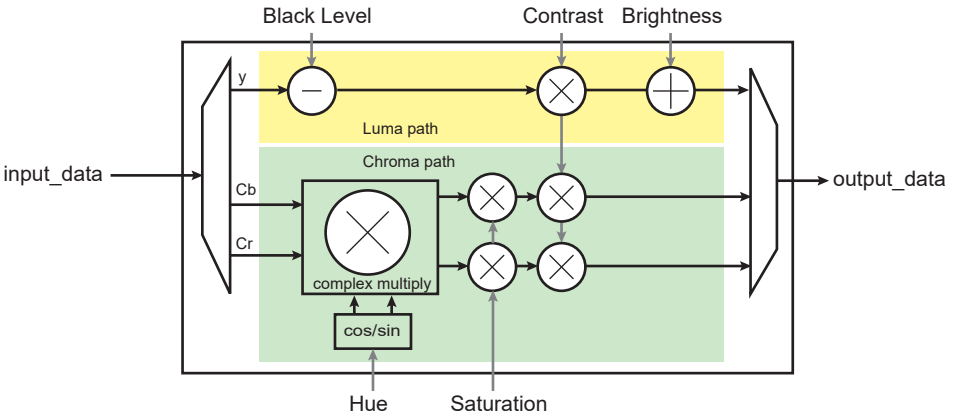
Mem addr	0x3								0x2								0x1								0x0							
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel 12 bpp	Cb1[7:0]								Cr1[7:0]								Cb0[7:0]								Cr0[7:0]							

44.6.7. Contrast Brightness, Hue, and Saturation

This module is for YCbCr format purposes. Brightness offset adapts the luma component, hue is used for chroma phase adjustment, and color saturation for chroma amplitude. Contrast gain is applied on all pixel components (luma and chroma).

A black level compensation is also included for YCbCr limited range, when the Luma component is limited to 16-235 instead of 0-255 range. It consists of subtracting the minimum value for limited range (16) from the Luma component to obtain an output Luma range starting at 0, which corresponds to the full-range black level value.

Figure 44.3. CBHS Block Diagram



44.6.8. Chroma Upsampling Unit

The LCDC supports YCbCr4:2:2 and 4:2:0 frame buffers. In 4:2:2, the chroma components are sampled at half the luma rate. The horizontal resolution is halved.

In 4:2:0, the chroma components are also sampled at half the luma rate, vertically and horizontally.

When these modes are selected, the Unified Scaling Engine is used to resample the chroma component and interpolate the missing components.

Figure 44.4. Vertical Chroma and Luma Locations for Progressive Content, 4:2:0 Encoded Frame

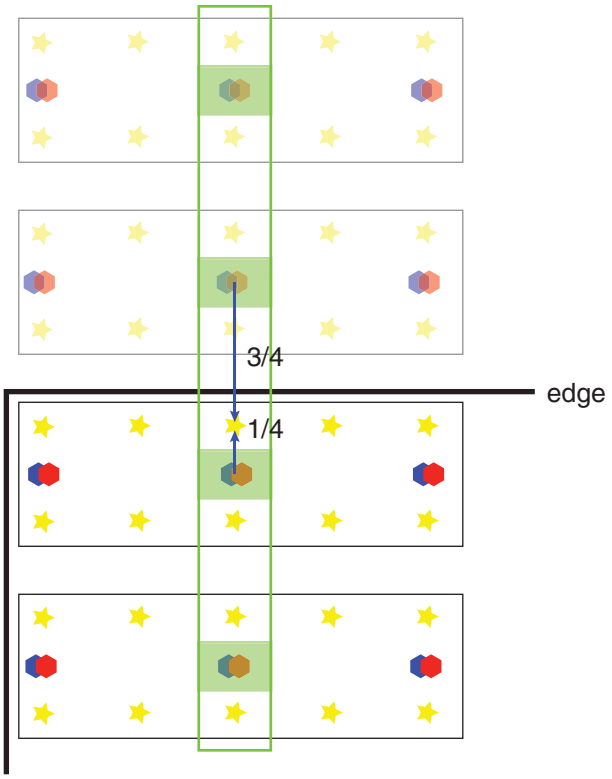


Figure 44.5. Horizontal Chroma and Luma Location for 4:2:2 and 4:2:0 YCbCr Encoded Frame

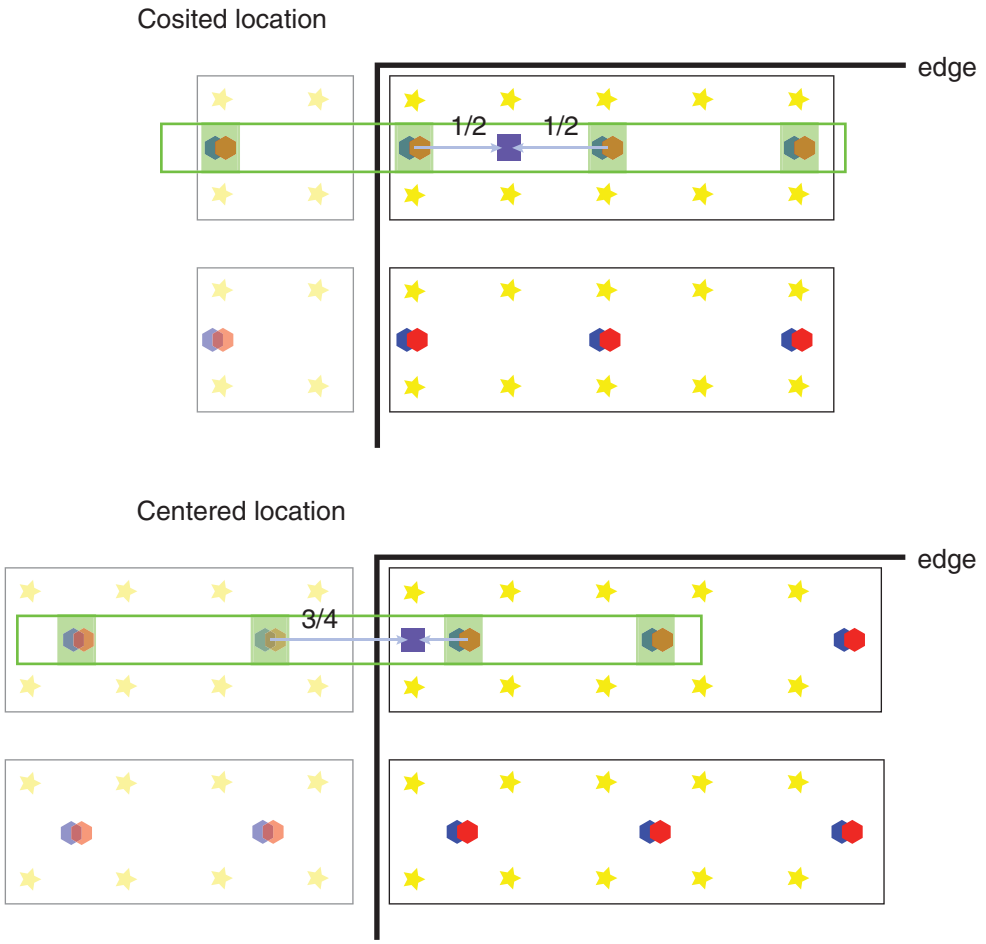
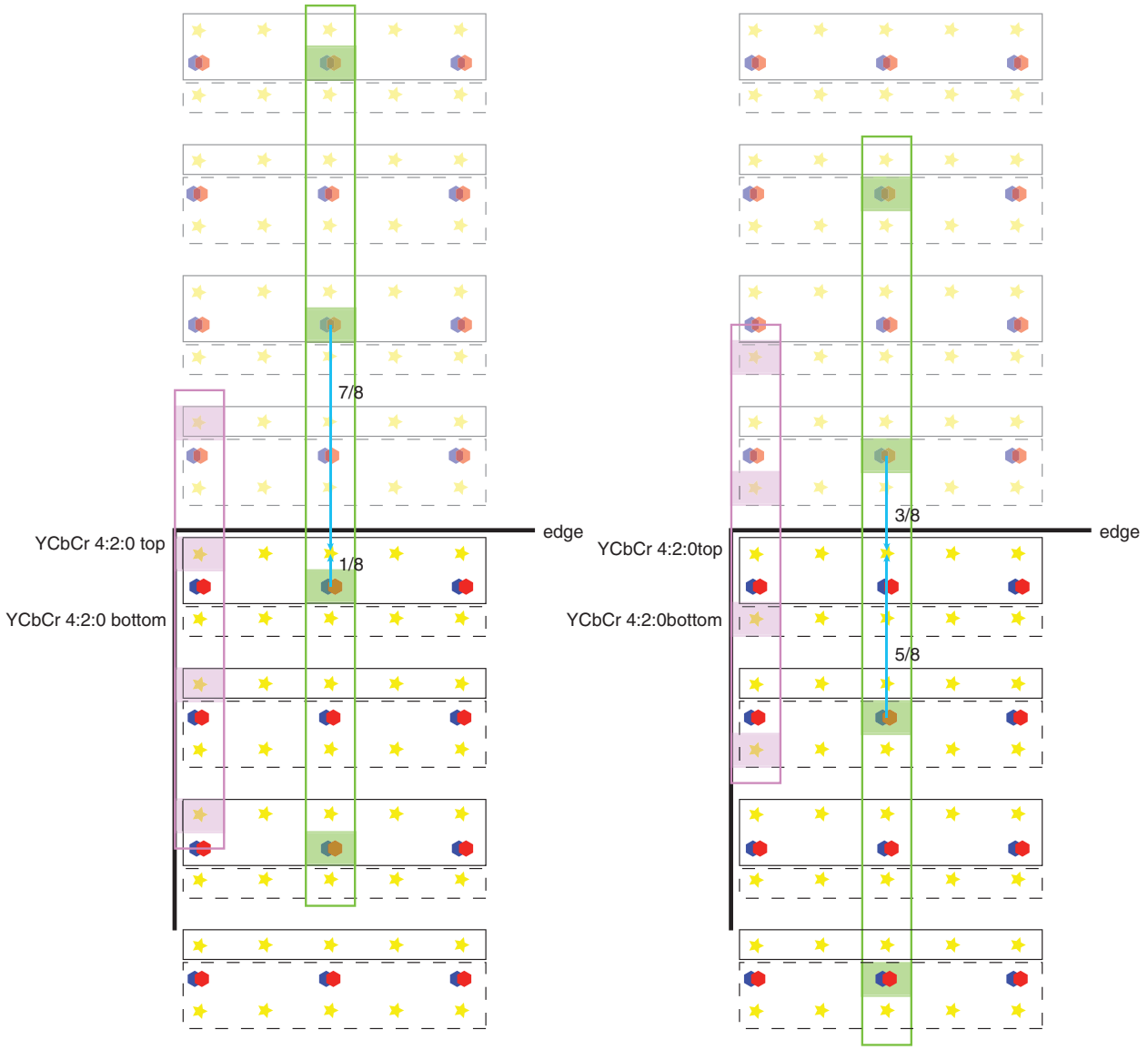


Figure 44.6. Chroma and Luma Locations for Vertical Scaling, With Interlaced Content



44.6.9. Striding

The LCDC includes a technique to increment the memory address by a programmable amount when the end of line has been reached. This offset is referred to as XSTRIDE and is defined on a per layer basis. Additionally, the PSTRIDE field allows a programmable jump at the pixel level. Pixel stride is the value from one pixel to the next.

44.6.9.1. Line Striding

When the end of line has been reached, the DMA address counter points to the next pixel address. The channel DMA current address value is added to the XSTRIDE field, and then updated. If XSTRIDE is set to zero, the DMA address register remains unchanged. The XSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The XSTRIDE field is a two's complement number. The following formula applies at the line boundary and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$\text{NextPixelAddress} = \text{CurrentPixelAddress} + \text{Sizeof}(\text{pixel}) + \text{XSTRIDE}$$

44.6.9.2. Pixel Striding

At each pixel, the DMA address counter points to the next pixel address. The channel DMA current address value is added to the PSTRIDE field and then updated. If PSTRIDE is set to 0, the DMA address register remains unchanged. The PSTRIDE field of the channel configuration register is aligned to the pixel size boundary. The PSTRIDE field is a two's complement number. The following formula applies at each pixel and indicates how the DMA controller computes the next pixel address. The function Sizeof() returns the number of bytes required to store a pixel.

$$\text{NextPixelAddress} = \text{CurrentPixelAddress} + \text{Sizeof}(\text{pixel}) + \text{PSTRIDE}$$

The Base Layer does not support pixel striding.

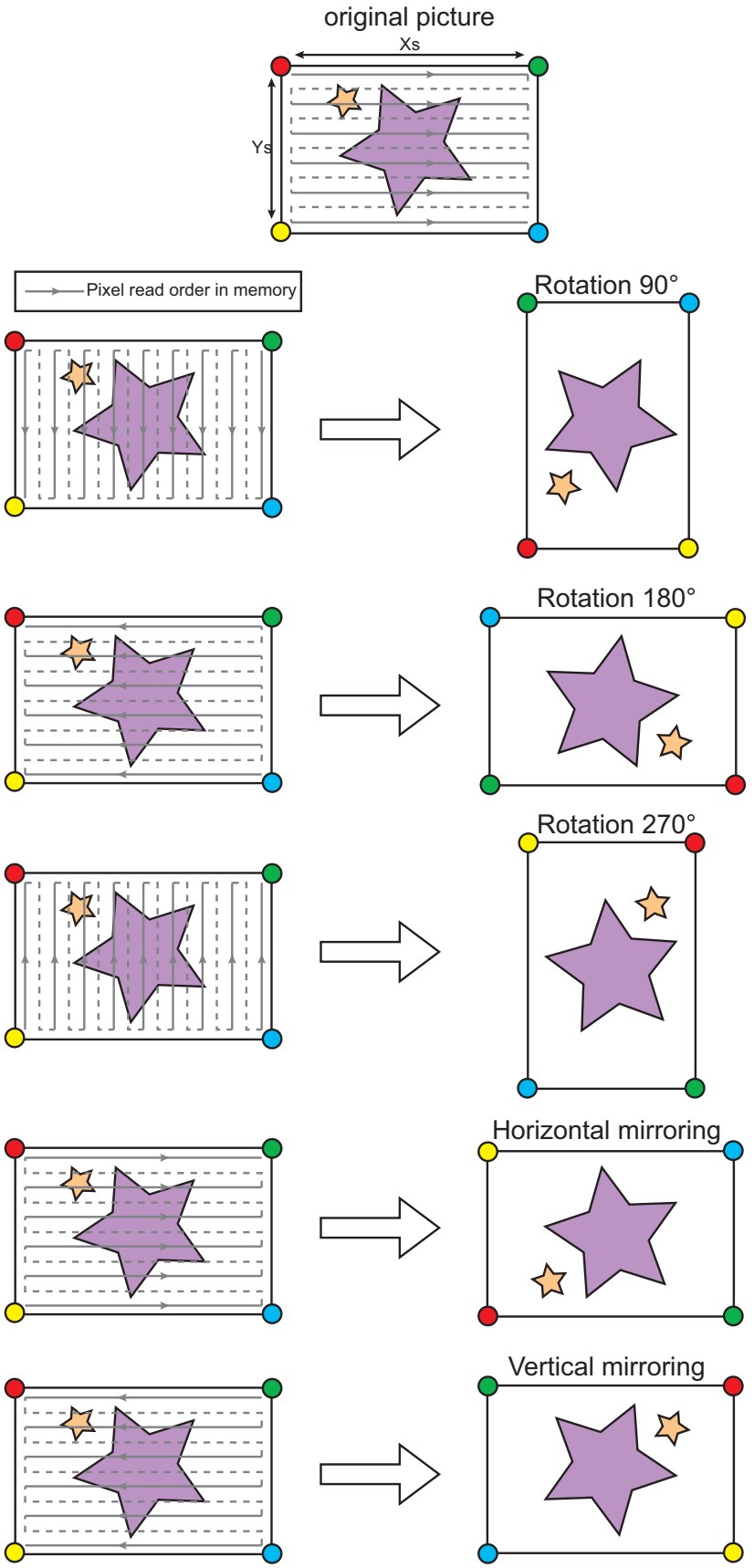
For overlays, pixel striding is not supported for the following modes:

- YCrCb 4:2:2 packed modes:
  - 16BPP_YCBCR_MODE0
  - 16BPP_YCBCR_MODE1
  - 16BPP_YCBCR_MODE2
  - 16BPP_YCBCR_MODE3
- CLUT modes less than 8bpp
  - CLUT_1BPP
  - CLUT_2BPP
  - CLUT_4BPP

44.6.9.3. Image Rotation And Mirroring

Image rotation or mirroring is produced by combining pixel and line striding to read image pixels in a different order, as shown in the figure below.

Figure 44.7. Rotation and Mirroring



The Configuration tables below show how to configure registers according to image transformation. The following notations are used:

- FPA = original image First Pixel Address
- Xs = original image horizontal number of pixels
- Ys = original image vertical number of pixels
- Ps = original image Pixel size expressed in byte (see Pixel Size Values)
- HXs = floor{(Xs+1)/2} , the rounded half of Xs
- HYs = floor{(Ys+1)/2} , the rounded half of Ys

Table 44.46. Pixel Size Values

Input Mode	Ps Value (in bytes)
12BPP_RGB_444	2
16BPP_ARGB_4444	2
16BPP_RGBA_4444	2
16BPP_RGB_565	2
16BPP_ARGB_1555	2
18BPP_RGB_666	4
18BPP_RGB_666PACKED	3
19BPP_ARGB_1666	4
19BPP_ARGB_PACKED	3
24BPP_RGB_888	4
24BPP_RGB_888_PACKED	3
25BPP_ARGB_1888	4
32BPP_ARGB_8888	4
32BPP_RGBA_8888	4
32BPP_AYCBCR	4
16BPP_YCBCR_MODE0	1
16BPP_YCBCR_MODE1	
16BPP_YCBCR_MODE2	
16BPP_YCBCR_MODE3	
16BPP_YCBCR_SEMIPLANAR	
16BPP_YCBCR_PLANAR	
12BPP_YCBCR_SEMIPLANAR	
12BPP_YCBCR_PLANAR	

Table 44.47. Rotation Configuration (All Input Modes, All Overlays)

Rotation Angle	0°	90°	180°	270°
FBA	FPA	FPA + (Xs-1)*Ps	FPA + (Xs*Ys - 1)*Ps	FPA + Xs*(Ys-1)*Ps
XSIZE/XMEMSIZE	Xs-1	Ys-1	Xs-1	Ys-1
YSIZE/YMEMSIZE	Ys-1	Xs-1	Ys-1	Xs-1
XSTRIDE	0	-((Ys-1)*Xs + 2)*Ps	-2*Ps	Xs*(Ys-1)*Ps
PSTRIDE	0	(Xs-1)*Ps	-2*Ps	-(Xs+1)*Ps

For rotation transformation in High-End Overlay with YCbCr planar and semi-planar modes, additional registers must be configured, as shown in the following tables.

Table 44.48. YCbCr 4:2:2 Semi-planar Specific Rotation Configuration

Rotation Angle	0°	90°	180°	270°
HEOCFG1.YCC422ROT	0	1	0	1
HEOCBFBA0	FPA	FPA + 2*HXs - 2	FPA + 2*HXs*Ys - 2	FPA + 2*HXs*(Ys-1)
HEOCFG7.CCXSTRIDE	0	-2*HXs*(Ys-1) - 4	-4	2*HXs*(Ys-1)
HEOCFG8.CCPSTRIDE	0	2*HXs - 2	-4	-2*HXs - 2

Table 44.49. YCbCr 4:2:2 Planar Specific Rotation Configuration

Rotation Angle	0°	90°	180°	270°
HEOCFG1.YCC422ROT	0	1	0	1
HEOCBFBA0	FPA	FPA + HXs - 1	FPA + HXs*Ys - 1	FPA + HXs*(Ys-1)
HEOCRFBA0	FPA	FPA + HXs - 1	FPA + HXs*Ys - 1	FPA + HXs*(Ys-1)
HEOCFG7.CCXSTRIDE	0	-HXs*(Ys-1) - 2	-2	HXs*(Ys-1)
HEOCFG8.CCPSTRIDE	0	HXs - 1	-2	-HXs - 1

Table 44.50. YCbCr 4:2:0 Semi-planar Specific Rotation Configuration

Rotation Angle	0°	90°	180°	270°
HEOCBFBA0	FPA	FPA + 2*HXs - 2	FPA + 2*HXs*HYs - 2	FPA + 2*HXs*(HYs-1)
HEOCFG7.CCXSTRIDE	0	-2*HXs*(HYs-1) - 4	-4	2*HXs*(HYs-1)
HEOCFG8.CCPSTRIDE	0	2*HXs - 2	-4	-2*HXs - 2

Table 44.51. YCbCr 4:2:0 Planar Specific Rotation Configuration

Rotation Angle	0°	90°	180°	270°
HEOCBFBA0	FPA	FPA + HXs - 1	FPA + HXs*Ys - 1	FPA + HXs*(HYs-1)
HEOCRFBA0	FPA	FPA + HXs - 1	FPA + HXs*Ys - 1	FPA + HXs*(HYs-1)
HEOCFG7.CCXSTRIDE	0	-HXs*(HYs-1) - 2	-2	HXs*(HYs-1)
HEOCFG8.CCPSTRIDE	0	HXs - 1	-2	-HXs - 1

Table 44.52. Mirroring Configuration (All Input Modes, All Overlays)

Mirroring Axis	Vertical	Horizontal
FBA	FPA + (Xs-1)*Ps	FPA + Xs*(Ys-1)*Ps
XSIZE/XMEMSIZE	Xs-1	Xs-1
YSIZE/YMEMSIZE	Ys-1	Ys-1
XSTRIDE	2*(Xs-1)*Ps	-2*Xs*Ps
PSTRIDE	-2*Ps	0

For mirroring transformation in High-End Overlay with YCbCr planar and semi-planar modes, additional registers have to be configured, as shown in the following tables.

Table 44.53. YCbCr 4:2:2 Semi-planar Specific Mirroring Configuration

Mirroring Axis	Vertical	Horizontal
HEOCBFBA0	FPA + 2*HXs - 2	FPA + 2*HXs*(Ys-1)
HEOCFG7.CCXSTRIDE	4*HXs - 4	-4*HXs
HEOCFG8.CCPSTRIDE	-4	0

Table 44.54. YCbCr 4:2:2 Planar Specific Mirroring Configuration

Mirroring Axis	Vertical	Horizontal
HEOCBFBA0	FPA + HXs - 1	FPA + HXs*(Ys-1)
HEOCRFBA0	FPA + HXs - 1	FPA + HXs*(Ys-1)
HEOCFG7.CCXSTRIDE	2*HXs - 2	-2HXs

Table 44.54. YCbCr 4:2:2 Planar Specific Mirroring Configuration (continued)

Mirroring Axis	Vertical	Horizontal
HEOCFG8.CCPSTRIDE	-2	0

Table 44.55. YCbCr 4:2:0 Semi-planar Specific Mirroring Configuration

Mirroring Axis	Vertical	Horizontal
HEOCBFBA0	FPA + 2*HXs - 2	FPA + 2*HXs*(HYs-1)
HEOCFG7.CCXSTRIDE	4*HXs - 4	-4*HXs
HEOCFG8.CCPSTRIDE	-4	0

Table 44.56. YCbCr 4:2:0 Planar Specific Mirroring Configuration

Mirroring Axis	Vertical	Horizontal
HEOCBFBA0	FPA + HXs - 1	FPA + HXs*(HYs-1)
HEOCRFBA0	FPA + HXs - 1	FPA + HXs*(HYs-1)
HEOCFG7.CCXSTRIDE	2*HXs - 2	-2HXs
HEOCFG8.CCPSTRIDE	-2	0

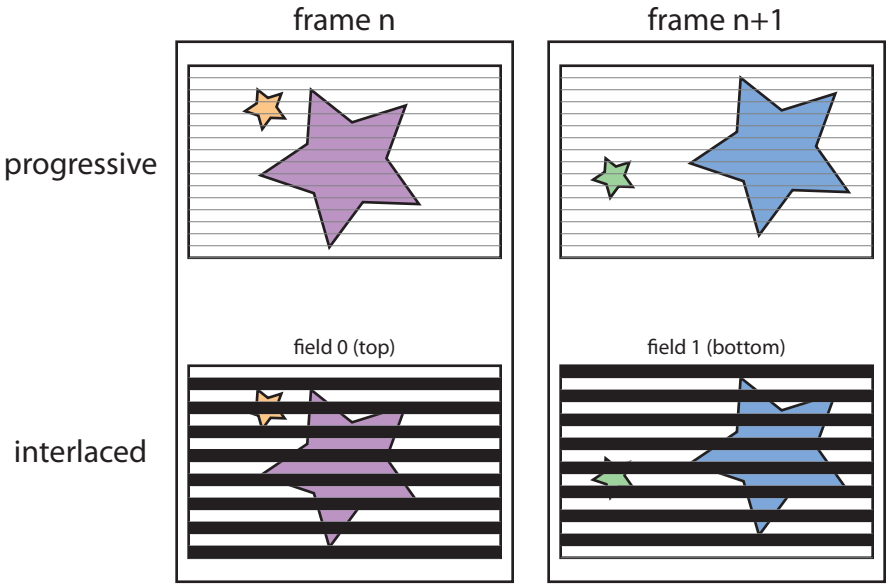
44.6.10. Gamma Correction

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 approximated by using the four 256-entry Color LUT. The table values are programmable through the User Interface when the CLUT mode and gamma correction are disabled (CLUTEN and GAM fields of the Configuration register 1 of the layer are cleared), or by the DMA as described in Color Look-Up Table Configuration. Each ARGB component input value can be associated with the gamma corrected value written in the CLUT.

44.6.11. Interlaced Frame Content

The LCDC supports progressive and interlaced content. Interlaced video signals provide only half the lines in each frame. Two fields are used alternately, each containing half the lines of the original frame. The top field, with index 0, contains every other line, starting with the first line. The bottom field, with index 1, contains every other line, starting with the second line. This is illustrated in the figure below.

Figure 44.8. Progressive and Interlaced Content



When this mode is selected, the Unified Scaling Engine is used to resample the current frame and interpolate the missing lines. To select Interlaced Content mode, set LCDC_HEOCFG1.ILD to one.

To handle both fields, specific configuration registers are duplicated depending on the field index, as described in the table below.

Table 44.57. Interlaced Configuration Registers

Field	Progressive	Interlaced Field 0 (Top)	Interlaced Field 1 (Bottom)
Frame buffer address	LCDC_HEOYFBA0 LCDC_HEOCBFBA0 LCDC_HEOCRFBA0		LCDC_HEOYFBA1 LCDC_HEOCBFBA1 LCDC_HEOCRFBA1
Scaling pixel alignment	LCDC_HEOCFG28.VXYOFF LCDC_HEOCFG28.VXCOFF LCDC_HEOCFG30.VXSYCFG LCDC_HEOCFG30.VXSCCFG		LCDC_HEOCFG28.VXYOFF1 LCDC_HEOCFG28.VXCOFF1 LCDC_HEOCFG30.VXSYCFG1 LCDC_HEOCFG30.VXSCCFG1

44.6.12. Color Space Conversion Unit

The color space conversion unit converts the Luma Chroma color space into the Red Green Blue (RGB) color space. The conversion matrix is defined below and is fully programmable through the User Interface.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \text{RYGAIN} & \text{RCBGAIN} & \text{RCRGAIN} \\ \text{GYGAIN} & \text{GCBGAIN} & \text{GCRGAIN} \\ \text{BYGAIN} & \text{BCBGAIN} & \text{BCRGAIN} \end{bmatrix} \cdot \begin{bmatrix} Y - \text{Yoff} \\ \text{Cb} - \text{Cboff} \\ \text{Cr} - \text{Croff} \end{bmatrix}$$

Color space conversion coefficients are defined with 1 sign bit, 2 integer bits and 10 fractional bits. The range of the ijGAIN coefficients is -4<ijGAIN<+4 with a step of 1/1024.

Additionally, a set scaling offset {Yoff=16, Cboff=128, Croff=128} can be applied by configuring LCDC_HEOCFG22.

44.6.13. Unified Scaling Engine

The High-End Overlay (HEO) data path includes a multi-mode hardware scaler that allows an image resize in both the horizontal and the vertical directions. It performs chroma upsampling for YCbCr

4:2:2 and YCbCr 4:2:0 formats, and also rescales the alpha channel. It integrates a two-tap or four-tap filter architecture with programmable polyphase coefficients or fixed bilinear interpolation coefficients.

44.6.13.1. Video Scaler Description

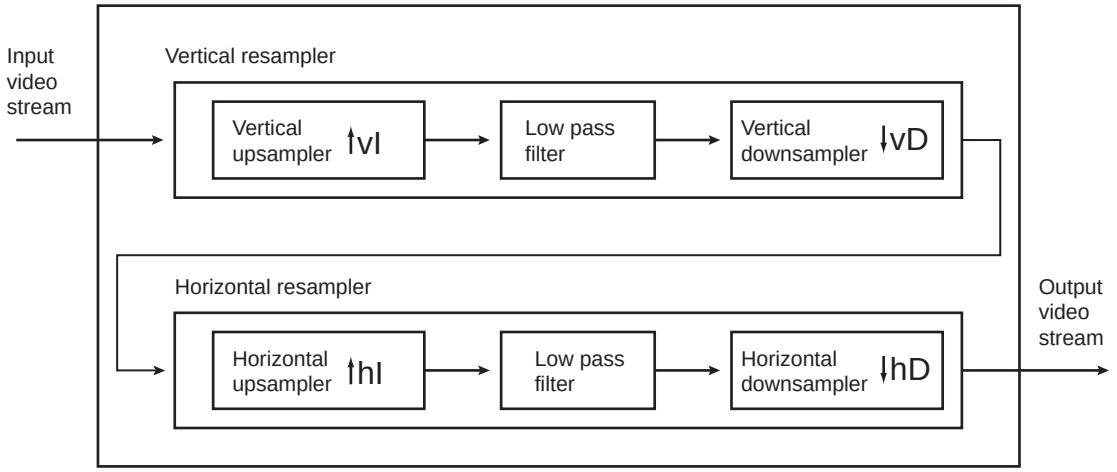
The scaling operation is based on a vertical and horizontal resampling algorithm. The sampling rate of the original image is increased when the video is upscaled, and decreased when the video is downscaled. A vertical resampler is used to perform vertical interpolation by a factor of vI, and a decimation by a factor of vD. A horizontal resampler is used to perform vertical interpolation by a factor of hI, and a decimation by a factor of hD. The horizontal and vertical low pass filters are both designed to minimize the aliasing effect. The frequency response of the low pass filter has the following characteristics:

$$H(\omega) = \begin{cases} I & \text{when } 0 \le |\omega| \le \min\left(\frac{\pi}{I}, \frac{\pi}{D}\right) \\ 0 & \text{otherwise} \end{cases}$$

Taking into account the linear phase condition and anticipating the filter length M, the desired frequency response is modified.

$$H(\omega) = \begin{cases} Ie^{-j\omega\frac{M}{2}} & \text{when } 0 \le |\omega| \le \min\left(\frac{\pi}{I}, \frac{\pi}{D}\right) \\ 0 & \text{otherwise} \end{cases}$$

Figure 44.9. Video Resampler Architecture



The impulse response of the defined low pass filter is:

$$h(n) = \begin{cases} I \times \frac{\omega_c}{\pi} \text{ when } n = 0 \\ I \times \frac{\omega_c}{\pi} \times \frac{\sin\omega cn}{\omega_c n} \text{otherwise} \end{cases}$$

Or, for the filter of length M:

$$h(n) = \begin{cases} I \times \frac{\omega_c}{\pi} \text{ when } n = \frac{M}{2} \\ I \times \frac{\omega_c}{\pi} \times \frac{\sin\omega cn - M2}{\omega_c\left(n - \frac{M}{2}\right)} \text{otherwise} \end{cases}$$

This ideal filter is non-causal and cannot be realized. The unit sample response h(n) is infinite in duration and must be truncated depending on the expected length M of the filter. This truncation is equivalent to the multiplication of the impulse response by a window function w(n).

Table 44.58. Window Function for a Filter Length M

Name of Window Function	Time Domain Sequence w(n)
Barlett	$1 - \frac{2 \times \left\lvert n - \frac{M-1}{2} \right\rvert}{M-1}$
Blackman	$0.42 - 0.5 \times \cos 2\pi nM-1 + 0.08 \times \cos 4\pi nM-1$
Hamming	$0.54 - 0.46 \times \cos 2\pi nM-1$
Hanning	$0.5 - 0.5 \times \cos 2\pi nM-1$

The horizontal resampler includes a 16-phase 4-tap filter equivalent to a 64-tap FIR illustrated in Horizontal Resampler Filter Architecture below.

The vertical resampler includes a 16-phase 4-tap filter equivalent to a 64-tap FIR illustrated in Vertical Resampler Filter Architecture below.

Figure 44.10. Horizontal Resampler Filter Architecture

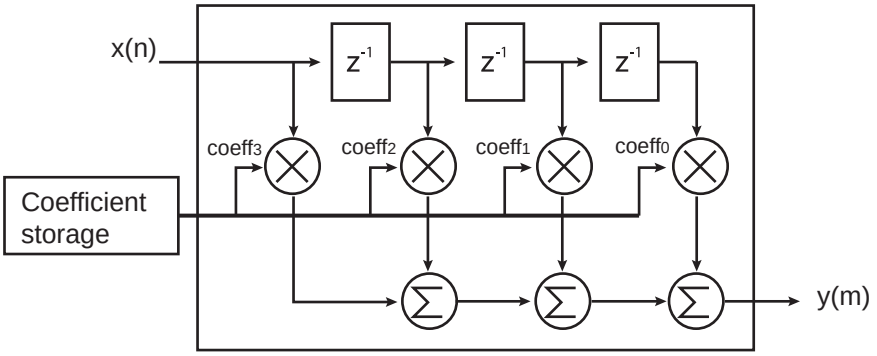
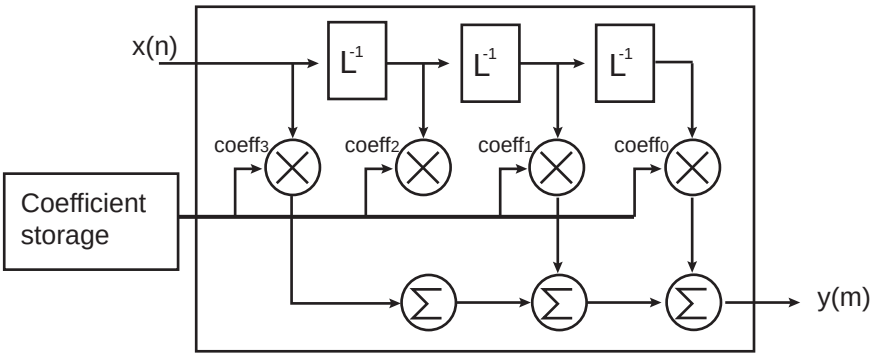


Figure 44.11. Vertical Resampler Filter Architecture



44.6.13.2.Horizontal Scaler Main Configuration

LCDC_HEOCFG4.XMEMSIZE indicates the horizontal size minus one of the image in the system memory. LCDC_HEOCFG3.XSIZE contains the horizontal size minus one of the window.

The YCbCr upsampling process requires that the horizontal scaler is composed of two channels. The first channel processes A and Y components (alph/luma) and the second channel processes Cb and Cr (chroma) components. Thus different scaling configurations are available for both channels.

In modes other than YCbCr 4:2:2 or YCbCr 4:2:0, both channels must be configured similarly to preserve image integrity.

LCDC_HEOCFG23.HXSYEN and LCDC_HEOCFG23.HXSCEN, respectively for Alpha/Luma and chroma channels, are used to activate the scaler. The scaling factors are programmed in LCDC_HEOCFG26.HXSYFACT and LCDC_HEOCFG27.HXSCFACT, respectively for alpha/luma and chroma channels.

The following equation calculates the horizontal scaling factor (HFACTOR) value:

$$\mathrm{HFACTOR} = \mathrm{round}\left(\frac{2^{20} \times \mathrm{XMEMSIZE}}{\mathrm{XSIZE}}\right)$$

The values in HXSYFACT and HXSCFACT are 24 bits wide (4-bit integer part, 20-bit fractional part), hence the horizontal downsampling capacity is limited to 16.0.

When YCbCr 4:2:2, or YCbCr 4:2:0 formats are used to resample the output window, the value of HXSCFACT must consider chroma upsampling by taking the half of HFACTOR.

When YCbCr 4:2:2, or YCbCr 4:2:0 formats are used without resampling the output window, , LCDC_HEOCFG23.HXSYEN and LCDC_HEOCFG23.HXSCEN can be set to zero. The scaler engine is then internally configured to process chroma upsampling without any configuration of LCDC_HEOCFG26 and LCDC_HEOCFG27.

Scaler Use Cases gives register values according to the context.

44.6.13.3.Vertical Scaler

LCDC_HEOCFG4.YMEMSIZE indicates the vertical size minus one of the image in the system memory. LCDC_HEOCFG3.YSIZE contains the vertical size minus one of the window.

For the YCbCr upsampling process, the vertical scaler is composed of two channels. The first channel processes A and Y components (alpha/luma), the second channel processes Cb and Cr (chroma). This allows different scaling configurations for both channels. In modes other than YCbCr 4:2:2 or YCbCr 4:2:0, both channels must be configured similarly to preserve image integrity.

LCDC_HEOCFG23.VXSYEN LCDC_HEOCFG23.VXSCEN, respectively for Alpha/Luma and chroma channels, are used to activate the scaler. The scaling factors are programmed in LCDC_HEOCFG24.VXSYFACT and LCDC_HEOCFG25.VXSCFACT, respectively for alpha/luma and chroma channels.

Use the following equation to calculate the VFACTOR value:

$$\mathrm{VFACTOR} = \mathrm{round}\left(\frac{2^{20} \times \mathrm{YMEMSIZE}}{\mathrm{YSIZE}}\right)$$

The VXSYFACT and VXSCFACT fields values are 24 bits wide (4-bit integer part, 20-bit fractional part), hence the vertical downsampling capacity is limited to 16.0.

When YCbCr 4:2:0 formats are used with resampling of the output window, the VXSCFACT field value has to consider the chroma upsampling by taking the half of VFACTOR.

When YCbCr 4:2:2, or YCbCr 4:2:0 formats are used without resampling the output window, then the LCDC_HEOCFG23.VXSYEN and LCDC_HEOCFG23.VXSCEN can be set to zero. The scaler engine is then internally configured to process chroma upsampling without any configuration of LCDC_HEOCFG24 and LCDC_HEOCFG25.

When Interlaced mode is used, the vertical scaler must be enabled and configured as detailed in Interlaced YCbCr 4:2:0, Chroma Upsampling (cosited), Top/Bottom Interpolation, With Window Resampling Configuration.

Scaler Use Cases gives register values according to the context.

44.6.13.4.Input/Output Pixel Alignment

Depending on application requirements, output pixel alignment according to input pixels can be tuned for the horizontal and vertical scaler. This tuning modifies the position of the output image within the range of the filter length. Two different types of configuration fields are used, Filter Taps Shift and Filter Init Phase Offset. Each of them produces a filter spatial response shift as described below.

Filter Taps Shift determines which set of input pixels is used at filter input to process a given output pixel. The following equation shows how the output pixel is calculated through the filter:

$$\mathrm{Pout}(m) = \sum_{k=0}^{N-1} \mathrm{Pin}(n + N - 1 - k - \mathrm{Tshift}) \times h(k, p)$$

With :

- Pin—input pixel component value (any A, R, G or B in ARGB mode, and A, Y, Cb or Cr in YCbCr mode)
- N—the filter number of taps (N=16)
- Tshift—Filter taps shift
- h(k, p)—Filter taps value depending on tap number k, and phase index p
- Pout—output pixel component value (A, R, G or B in ARGB mode, and A, Y, Cb or Cr in YCbCr mode)

The output image is shifted by the value Tshift/FACT. Figure 44.12 and Figure 44.13 illustrate an upsampling by 4 use case (FACT=0.25x2^{20}):

Figure 44.12. Filter Taps Shift

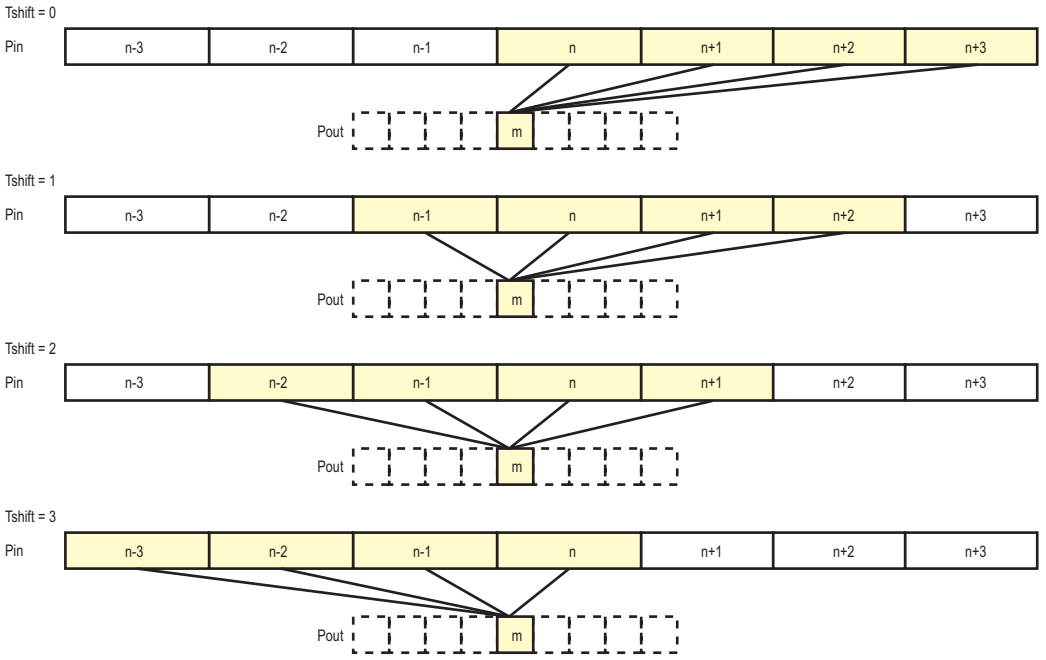


Figure 44.13. Filter Taps Shift Results for Different Horizontal Tshift Values (Upsampling by 4)

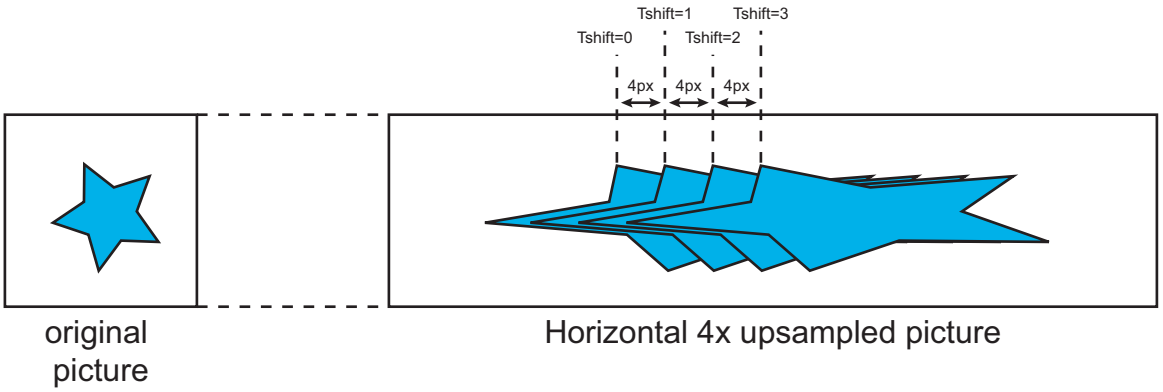
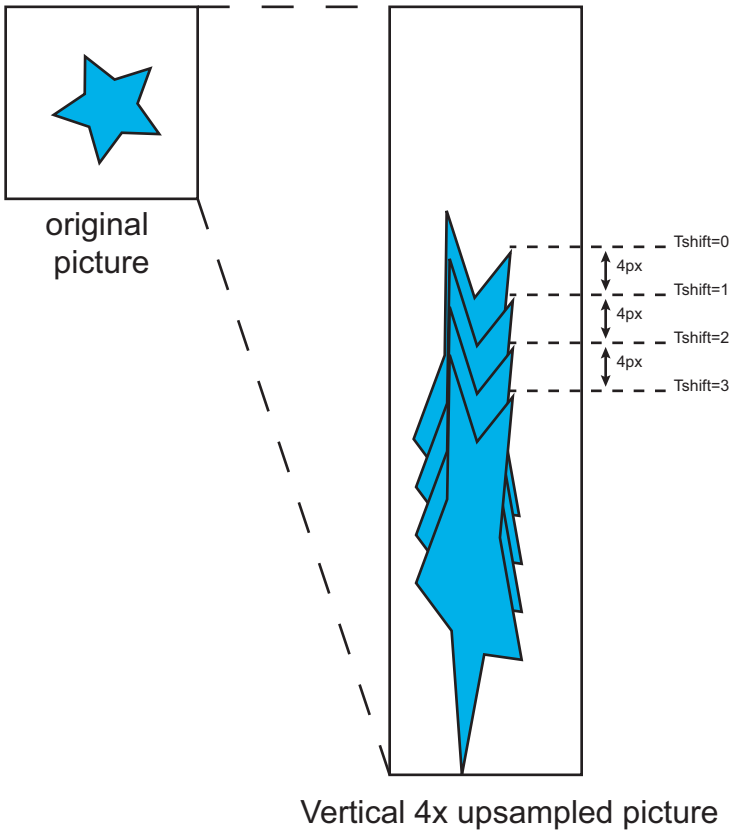


Figure 44.14. Filter Taps Shift Results for Different Vertical Tshift Values (Upsampling by 4)



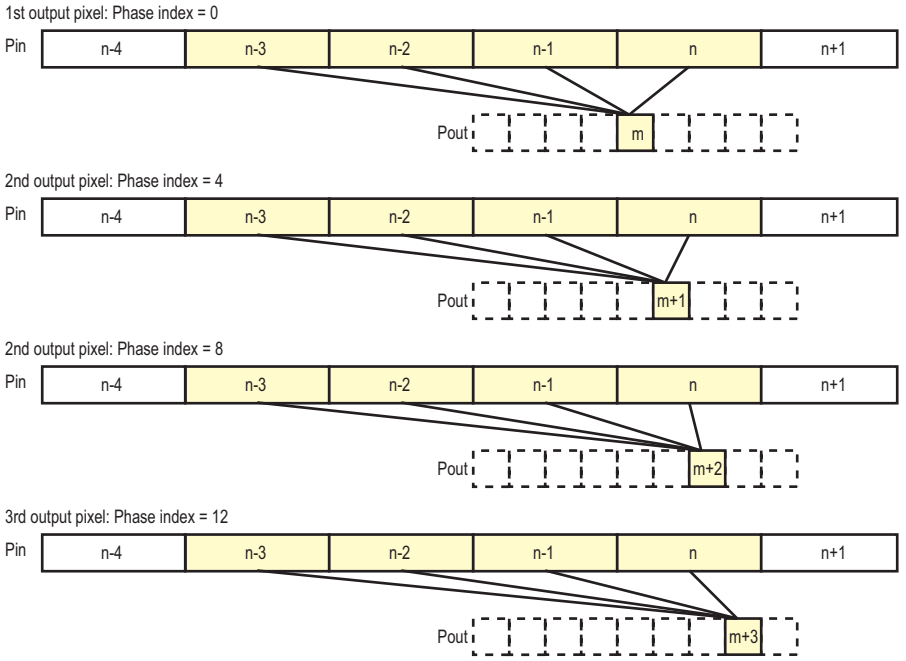
Filter taps shift Tshift is controlled by LCDC_HEOCFG30.HXSYCFG and LCDC_HEOCFG30.VXSCCFG for the horizontal scaler, and LCDC_HEOCFG30.VXSYCFG and LCDC_HEOCFG30.VXSCCFG for the vertical scaler.

LCDC_HEOCFG30.VXSYCFG1 and LCDC_HEOCFG30.VXSCCFG1 should also be considered in case of interlaced content.

During the resampling process, a filter tap phase is selected for each output pixel, representing the fractional part of the output pixel spatial position, according to the input pixel width. An input pixel width is subsampled in 16 phases values, each corresponding to a subset of filter taps p. Phase 0 corresponds to 0 fractional shift. Incrementing the phase index by 1 corresponds to a spatial shift of 1/16 pixel width.

The figure below illustrates an upsampling by 4 use case (4 output pixels for one input pixel), with a fixed Tshift value.

Figure 44.15. Filter Phase Index (Tshift = 3)



An additional phase offset can be configured in HEO_CFG28 and HEO_CFG29 and adds a fractional shift to all pixels of the frame. This is illustrated in the figure below for the use case when upsizing by 4.

Figure 44.16. Horizontal Initial Phase Offset On Upsizing by 4

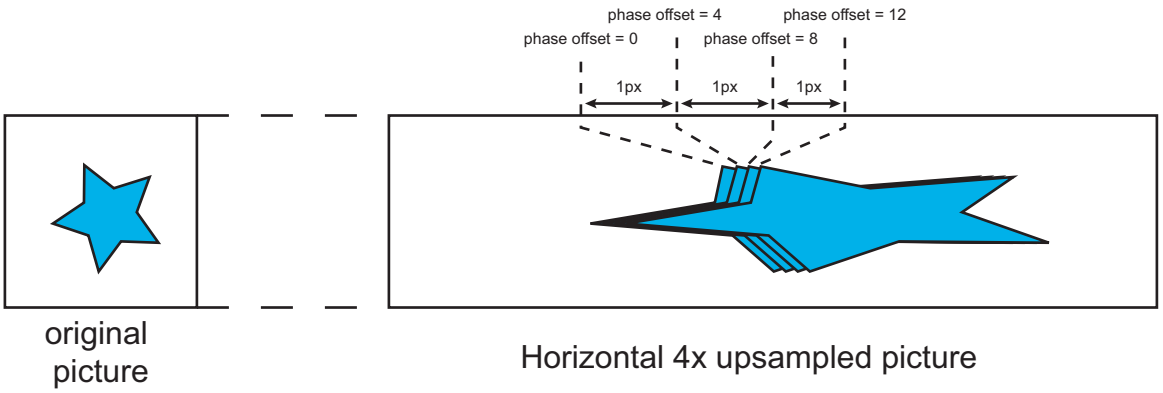
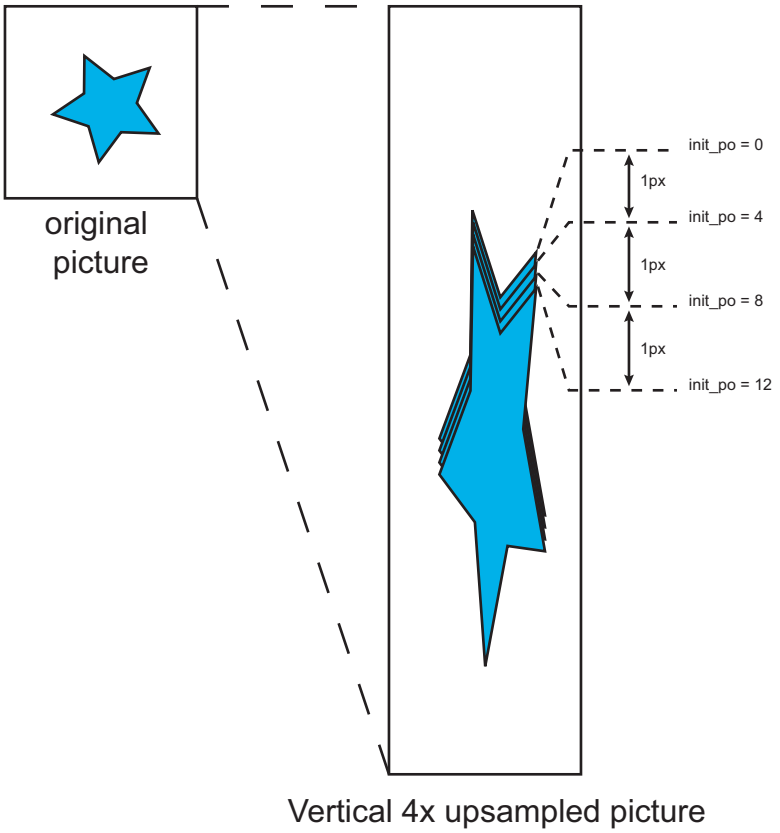


Figure 44.17. Vertical Initial Phase Offset on Upsizing by 4



The initial phase offset is controlled by LCDC_HEOCFG29.HXSCOFF and LCDC_HEOCFG29.HXSYOFF for the horizontal scaler, and LCDC_HEOCFG28.VXSYOFF and LCDC_HEOCFG28.VXSCOFF for the vertical scaler.

LCDC_HEOCFG28.VXSYOFF1 and LCDC_HEOCFG28.VXSCOFF1 should also be considered in case of interlaced content.

44.6.13.5.Scaler Use Cases

Table 44.59. Progressive ARGB/**YCbCr 4:4:4**, Window Resampling Configuration

Unified Scaling Config	Bit/Field	Value
HEOCFG23	VXSYEN	Set both bits to 1 to activate the vertical filter
HEOCFG23	VXSCEN	
HEOCFG23	HXSYEN	Set both bits to 1 to activate the horizontal filter
HEOCFG23	HXSCEN	
HEOCFG24	VXSYFACT	VFACTOR
HEOCFG25	VXSCFACT	
HEOCFG26	HXSYFACT	HFACTOR
HEOCFG27	HXSCFACT	
HEOCFG28	VXSYOFF	0
	VXSYOFF1	Unused in this mode
	VXSCOFF	0
	VXSCOFF1	Unused in this mode
HEOCFG29	HXSYOFF	0
	HXSCOFF	0

Table 44.59. Progressive ARGB/**YCbCr 4:4:4**, Window Resampling Configuration (continued)

Unified Scaling Config	Bit/Field	Value
HEOCFG30	VXSYCFG	1
	VXSYTAP2	0
	VXSY1201N	0
	VXSYBICU	0
	VXSYCFG1	Unused in this mode
	VXSCCFG	1
	VXSCTAP2	0
	VXSC1201N	0
	VXSCBICU	0
	VXSCCFG1	Unused in this mode
HEOCFG31	HXSYCFG	1
	HXSYTAP2	0
	HXSYBICU	0
	HXSCCFG	1
	HXSCTAP2	0
	HXSCBICU	0
HEOVTAP10PHIx	TAP0	When the vertical filter is enabled, program the filter coefficients here
	TAP1	
HEOVTAP32PHIx	TAP2	
	TAP3	
HEOHTAP10PHIx	TAP0	When the horizontal filter is enabled, program the filter coefficients here
	TAP1	
HEOHTAP32PHIx	TAP2	
	TAP3	

Table 44.60. Progressive YCbCr 4:2:2 Chroma Upsampling (cosited), with Window Resampling Configuration

Unified Scaling Config	Bit/Field	Resampling Type		
		Nearest Neighbor	Bilinear	Bicubic
HEOCFG23	VXSYEN	Set both bits to 1 to activate the vertical filter		
HEOCFG23	VXSCEN			
HEOCFG23	HXSYEN	Set to 1 to activate the horizontal filter		
HEOCFG23	HXSCEN	1		
HEOCFG24	VXSYFACT	VFACTOR		
HEOCFG25	VXSCFACT			
HEOCFG26	HXSYFACT	HFACTOR		
HEOCFG27	HXSCFACT	round(HFACTOR/2.0) divided by 2.0 to perform chroma upsampling (4:2:2 to 4:4:4 operation)		
HEOCFG28	VXSYOFF	0		
	VXSYOFF1	Unused in this mode		
	VXSCOFF	0		
	VXSCOFF1	Unused in this mode		
HEOCFG29	HXSYOFF	Unused in this mode		
	HXSCOFF	0		

Table 44.60. Progressive YCbCr 4:2:2 Chroma Upsampling (cosited), with Window Resampling Configuration (continued)

Unified Scaling Config	Bit/Field	Resampling Type		
		Nearest Neighbor	Bilinear	Bicubic
HEOCFG30	VXSYCFG	1		
	VXSYTAP2	0	1	0
	VXSY1201N	0		
	VXSYBICU	0	0	1
	VXSYCFG1	Unused in this mode		
	VXSCCFG	1		
	VXSCTAP2	0	1	0
	VXSC1201N	0		
	VXSCBICU	0	0	1
	VXSCCFG1	Unused in this mode		
HEOCFG31	HXSYCFG	1		
	HXSYTAP2	0	1	0
	HXSYBICU	0	0	1
	HXSCCFG	1		
	HXSCTAP2	0	1	0
	HXSCBICU	0	0	1
HEOVTAP10PHIx	TAP0	0.0	Unused in this mode	
	TAP1	1.0		
HEOVTAP32PHIx	TAP2	0.0		
	TAP3	0.0		
HEOHTAP10PHIx	TAP0	0.0		
	TAP1	1.0		
HEOHTAP32PHIx	TAP2	0.0		
	TAP3	0.0		

Table 44.61. Progressive YCbCr 4:2:0, Chroma Upsampling (cosited), with Window Resampling Configuration

Unified Scaling Config	Bit/Field	Value
HEOCFG23	VXSYEN	1
HEOCFG23	VXSCEN	1
HEOCFG23	HXSYEN	1
HEOCFG23	HXSCEN	1
HEOCFG24	VXSYFACT	VFACTOR
HEOCFG25	VXSCFACT	round(VFACTOR/2.0) divided by 2.0 to perform vertical chroma upsampling (4:2:0 to 4:2:2 operation)
HEOCFG26	HXSYFACT	HFACTOR
HEOCFG27	HXSCFACT	round(HFACTOR/2.0) divided by 2.0 to perform horizontal chroma upsampling (4:2:2 to 4:4:4 operation)
HEOCFG28	VXSYOFF	0
	VXSYOFF1	Unused in this mode
	VXSCOFF	0
	VXSCOFF1	Unused in this mode
HEOCFG29	HXSYOFF	0
	HXSCOFF	0

Table 44.61. Progressive YCbCr 4:2:0, Chroma Upsampling (cosited), with Window Resampling Configuration (continued)

Unified Scaling Config	Bit/Field	Value
HEOCFG30	VXSYCFG	1
	VXSYTAP2	1
	VXSY1201N	0
	VXSYBICU	0
	VXSYCFG1	Unused in this mode
	VXSCCFG	1
	VXSCTAP2	1
	VXSC1201N	0
	VXSCBICU	0
	VXSCCFG1	Unused in this mode
HEOCFG31	HXSYCFG	1
	HXSYTAP2	1
	HXSYBICU	0
	HXSCCFG	1
	HXSCTAP2	1
	HXSCBICU	0
HEOVTAP10PHIx	TAP0	Unused in this mode
	TAP1	
HEOVTAP32PHIx	TAP2	
	TAP3	
HEOHTAP10PHIx	TAP0	
	TAP1	
HEOHTAP32PHIx	TAP2	
	TAP3	

Table 44.62. Interlaced YCbCr 4:2:0, Chroma Upsampling (cosited), Top/Bottom Interpolation, With Window Resampling Configuration

Unified Scaling Config	Bit/Field	Value
HEOCFG23	VXSYEN	1
HEOCFG23	VXSCEN	1
HEOCFG23	HXSYEN	1
HEOCFG23	HXSCEN	1
HEOCFG24	VXSYFACT	round(VFACTOR/2.0) divided by 2.0 to perform interlaced content upscaling
HEOCFG25	VXSCFACT	round(VFACTOR/4.0) divided by 4.0 to perform interlaced content upscaling and chroma upscaling (4:2:0 to 4:2:2)
HEOCFG26	HXSYFACT	HFACTOR
HEOCFG27	HXSCFACT	round(HFACTOR/2.0) divided by 2.0 to perform horizontal chroma upsampling (4:2:2 to 4:4:4 operation)
HEOCFG28	VXSYOFF	0 (top)
	VXSYOFF1	8 (bottom)
	VXSCOFF	0 (top)
	VXSCOFF1	8 (bottom)

Table 44.62. Interlaced YCbCr 4:2:0, Chroma Upsampling (cosited), Top/Bottom Interpolation, With Window Resampling Configuration (continued)

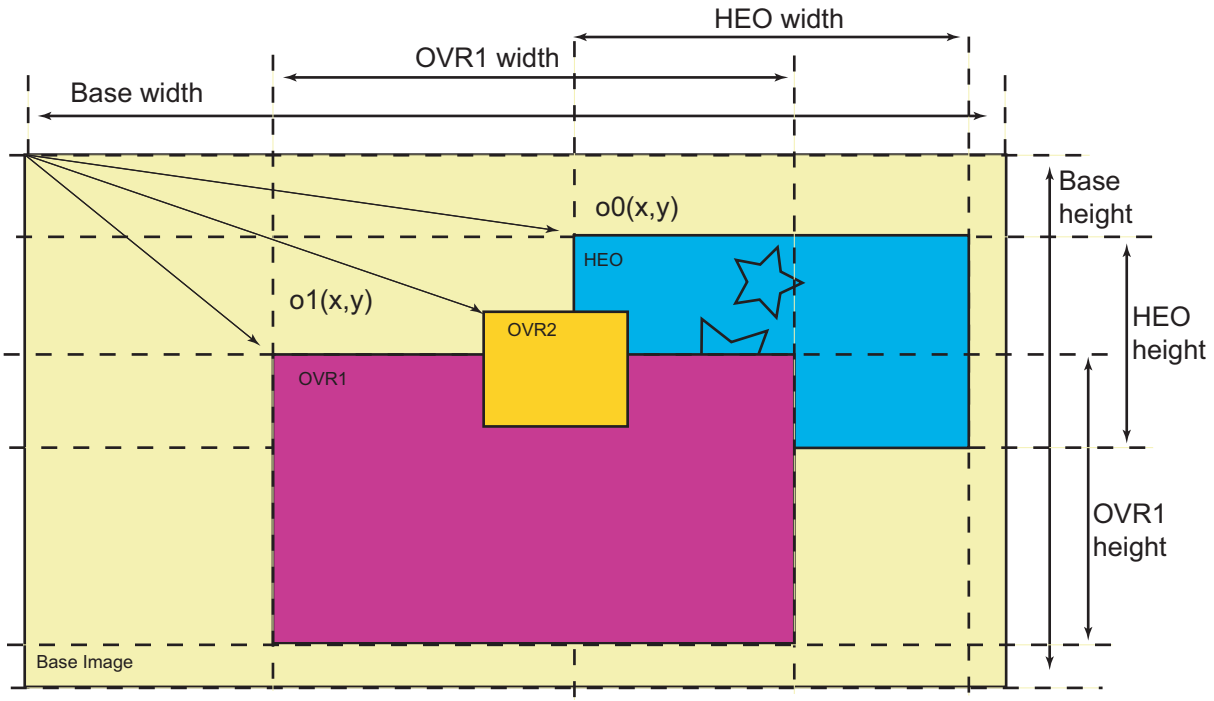
Unified Scaling Config	Bit/Field	Value
HEOCFG29	HXSYOFF	0
	HXSCOFF	0
HEOCFG30	VXSYCFG	1 (top)
	VXSYTAP2	0
	VXSY1201N	0
	VXSYBICU	0
	VXSYCFG1	2 (bottom)
	VXSCCFG	1 (top)
	VXSCTAP2	1
	VXSC1201N	0
	VXSCBICU	0
	VXSCCFG1	2 (bottom)
HEOCFG31	HXSYCFG	1
	HXSYTAP2	1
	HXSYBICU	0
	HXSCCFG	1
	HXSCTAP2	1
	HXSCBICU	0
HEOVTAP10PHIx	TAP0	Coefficients are used for luma vertical resampling only (chroma is bilinear)
	TAP1	
HEOVTAP32PHIx	TAP2	
	TAP3	
HEOHTAP10PHIx	TAP0	Coefficients are used for luma horizontal resampling only (chroma is bilinear)
	TAP1	
HEOHTAP32PHIx	TAP2	
	TAP3	

44.6.14. Color Combine Unit

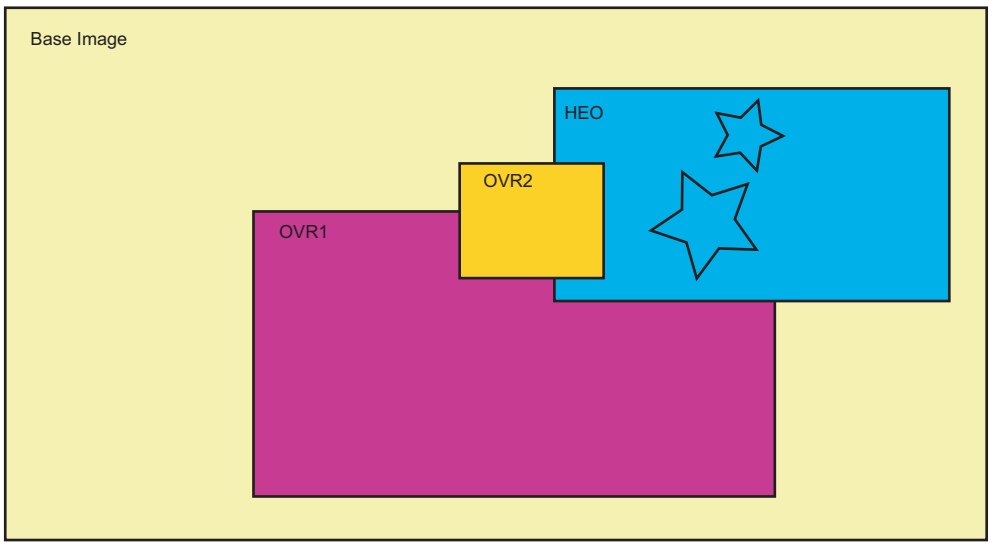
44.6.14.1.Window Overlay

The LCDC provides hardware support for multiple “overlay planes” that can be used to display windows on top of the image without destroying the image located below. The overlay image can use any color depth. Using the overlay alleviates the need to re-render the occluded portion of the image. When pixels are combined together through the alpha blending unit, a new color is created. This new pixel is called an iterated pixel and is passed to the next blending stage. Then, this pixel may be combined again with another pixel. LCDC_HEOCFG12.VIDPRI configures the video priority algorithm used to display the layers. When VIDPRI is written to zero, the OVR1 layer is located above the HEO layer. When VIDPRI is written to one, OVR1 is located below the HEO layer.

Figure 44.18. Overlay Example with Two Different Video Prioritization Algorithms



Video Prioritization Algorithm 1 : OVR2 > OVR1 > HEO > BASE



Video Prioritization Algorithm 2 : OVR2 > HEO > OVR1 > BASE

44.6.14.2.Overlay Blending

The blending function requires two pixels (one iterated from the previous blending stage and one from the current overlay color) and a set of blending configuration parameters. These parameters are separated according to component type: A for Alpha, or C for any color component R, G or B.

Figure 44.19. Alpha Blending Operation Block Diagram

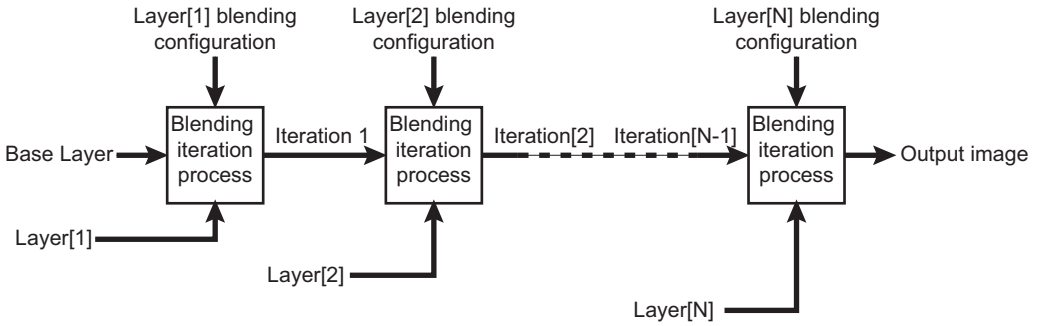
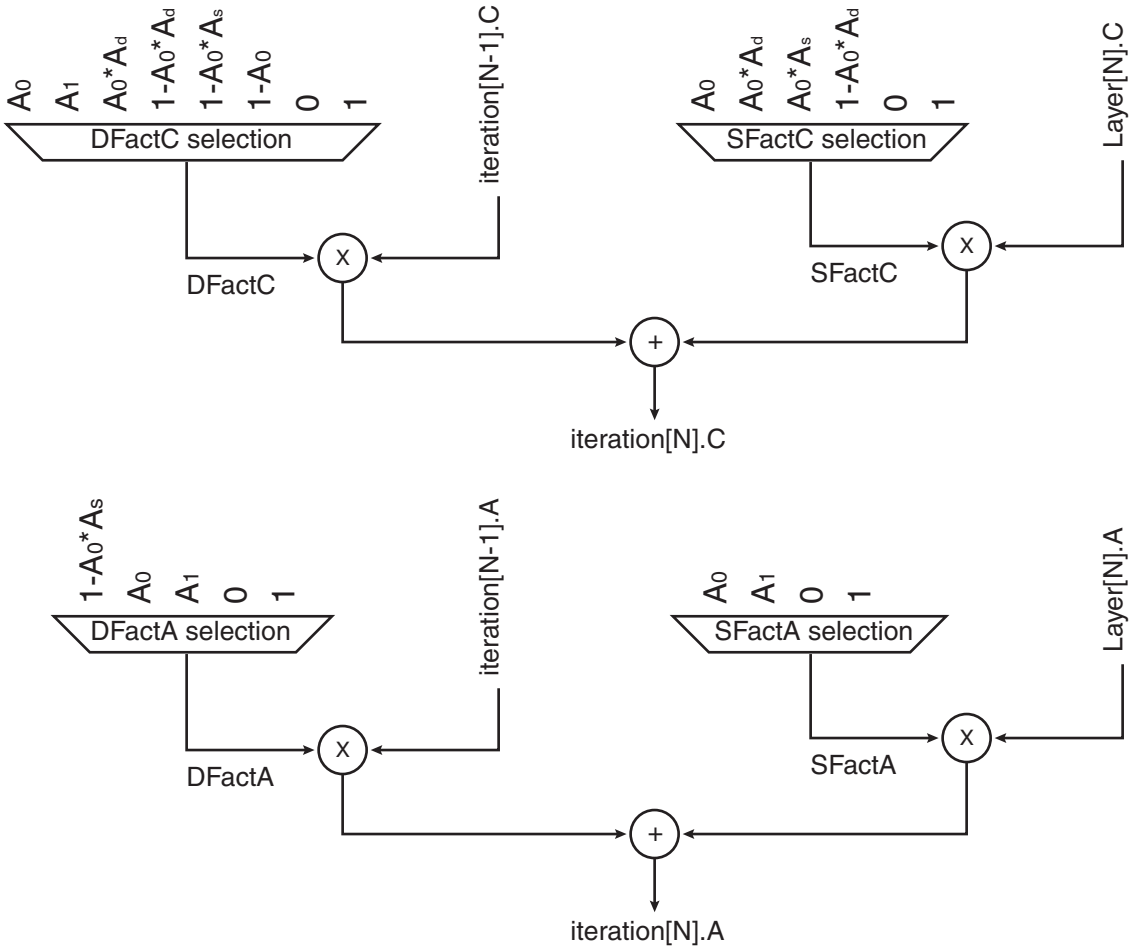


Figure 44.20. Blending Iteration Process Block Diagram



44.6.14.3.Window Blending

Figure 44.21. 256-level Alpha Blending

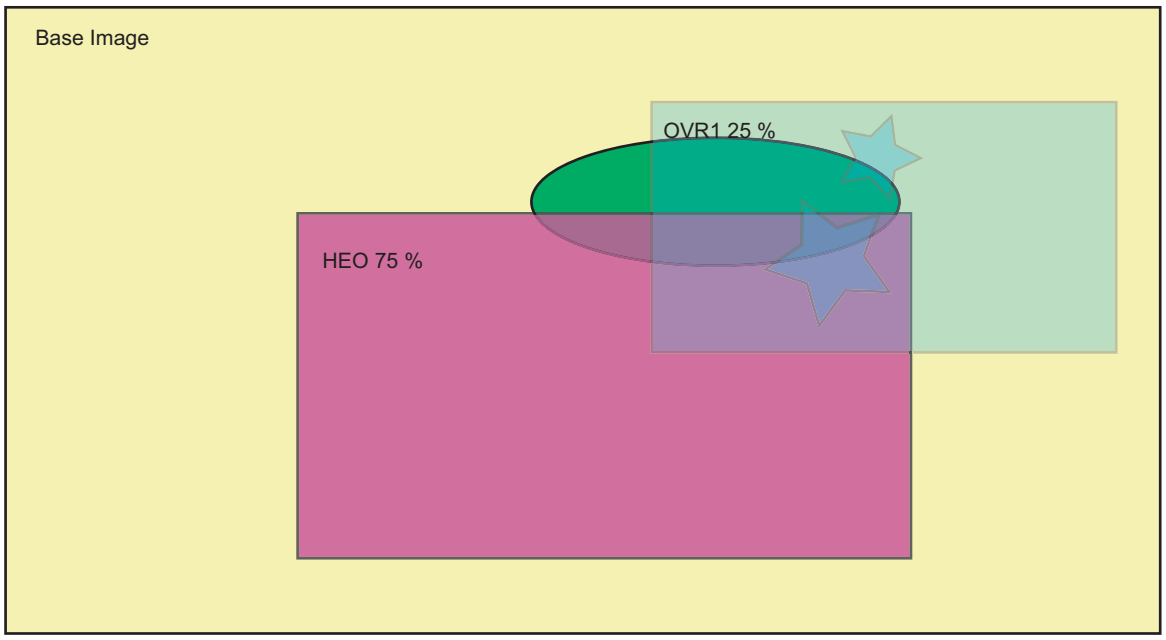
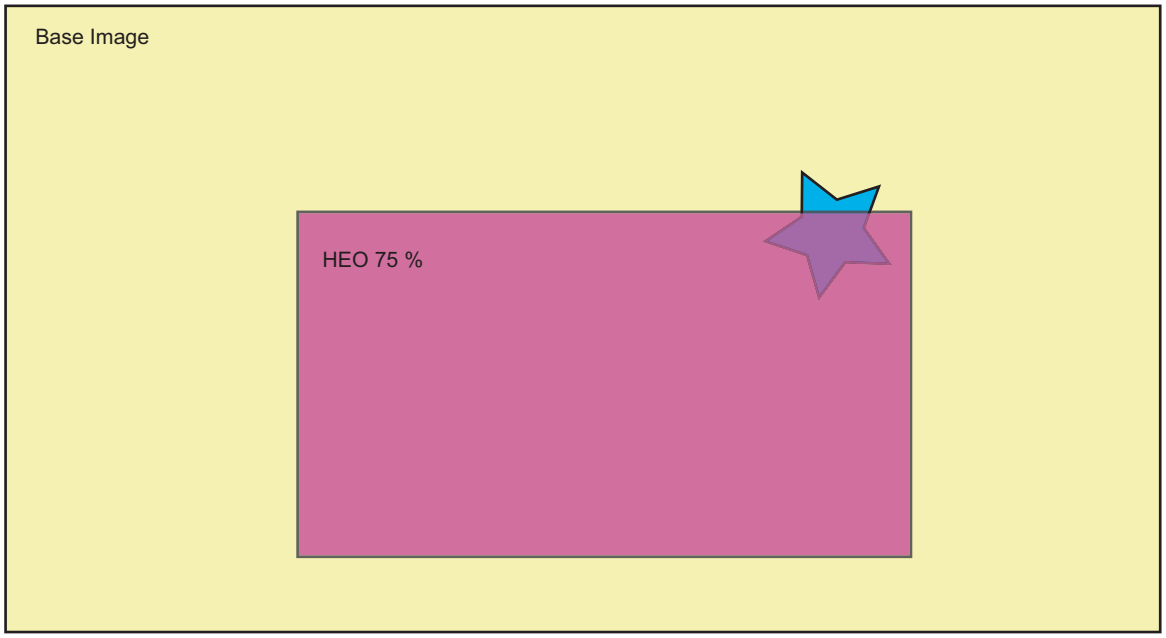


Figure 44.22. Figure 7-23. 256-level Alpha Blending



44.6.14.4.Color Keying

Color keying involves a method of bit-block image transfer (Blit). This entails blitting one image onto another where not all the pixels are copied. Blitting usually involves two bitmaps: a source bitmap and a destination bitmap. A raster operation (ROP) is performed to define whether the iterated color or the overlay color is to be visible or not.

44.6.14.4.1.Source Color Keying

If the masked overlay color matches the color key, the iterated color is selected and Source Color Keying is activated using the following configuration sequence:

1. Select the overlay to blit.
2. Write a zero to DSTKEY.
3. Activate Color Keying by writing a one to CRKEY.
4. Configure the Color Key by writing RKEY, GKEY and BKEY fields.
5. Configure the Color Mask by writing RKEY, GKEY and BKEY fields.

When the field RMASK, GMASK, or BMASK is configured to zero, the comparison is disabled and the raster operation is activated.

44.6.14.4.2.Destination Color Keying

If the iterated masked color matches the color key then the overlay color is selected, Destination Color Keying is activated using the following configuration sequence:

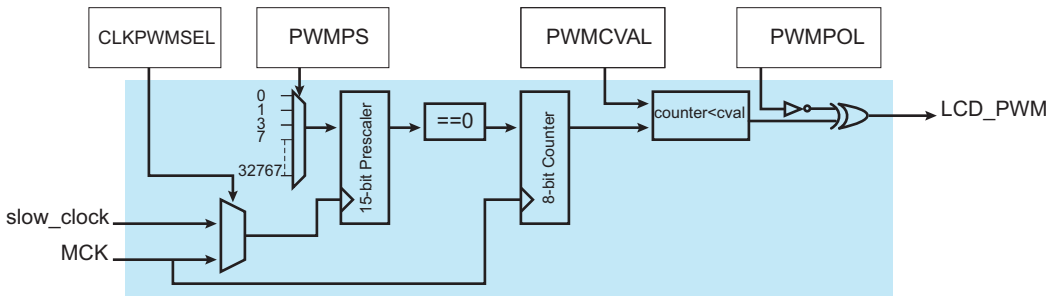
1. Select the overlay to blit.
2. Write a one to DSTKEY.
3. Activate Color Keying by writing a one to CRKEY bit
4. Configure the Color Key by writing RKEY, GKEY and BKEY fields.
5. Configure the Color Mask by writing RKEY, GKEY and BKEY fields.

When the field RMASK, GMASK, or BMASK is configured to zero, the comparison is disabled and the raster operation is activated.

44.6.15. LCDC PWM Controller

The LCDC integrates a Pulse Width Modulation (PWM) Controller.

Figure 44.23. PWM Controller Block Diagram



This block generates the LCD contrast control signal (LCD_PWM) that controls the display's contrast by software. LCDC_PWM is an 8-bit PWM signal that can be converted to an analog voltage with a simple passive filter.

The PWM module has a free-running counter whose value is compared against a compare register (LCDC_LCDCFG6.PWMCVAL). If the value in the counter is less than that in the register, LCDC_PWM is asserted with the value in the compare register (LCDC_LCDCFG6.PWMCVAL). Otherwise, the opposite value is output. Thus, a periodic waveform with a pulse width proportional to the value in the compare register is generated.

Due to the comparison mechanism, the output pulse has a width between 0 and 255 PWM counter cycles. Thus, by adding a simple passive filter outside the chip, an analog voltage between 0 and (255/256) × V_{DD} can be obtained for the positive polarity case, or between (1/256) × V_{DD} and V_{DD} for the negative polarity case. Other voltage values can be obtained by adding active external circuitry.

For PWM mode, the counter frequency can be adjusted to 16 different values using LCDC_LCDCFG6.PWMPS.

The PWM module can be fed with the slow clock or the peripheral clock (MCK), depending on LCDC_CFG0.CLKPWMSEL.

LCD display panels have different backlight specifications in terms of minimum/maximum values for PWM frequency. If the LCDC PWM frequency range does not match the LCD display panel, it is possible to use the standalone PWM Controller (refer to the section “Pulse Width Modulation Controller (PWM)”) to drive the backlight.

44.6.16. Register Write Protection

To prevent any single software error from corrupting LCDC behavior, certain registers in the address space can be write-protected by setting the bits WPCFGE, WPITE, WPCRE in the LCDC Write Protection Mode register (LCDC_WPMR).

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in the LCDC Write Protection Status register (LCDC_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 LCDC_WPSR.

44.6.17. LCD Overall Performance

44.6.17.1.Color Look-Up Table (CLUT)

Table 44.63. CLUT Pixel Performance

CLUT Mode	Pixels/Word	Pixel Striding (Rotation/Mirroring)	Scaling
1 bpp	32	Not supported	Supported
2 bpp	16	Not supported	Supported
4 bpp	8	Not supported	Supported
8 bpp	4	Supported	Supported

44.6.17.2.RGB Mode Fetch Performance

The table below gives the maximum theoretical bandwidth according to input modes in burst access. In case of random single pixel fetch (when PSTRIDE is not 0) in non-zero wait state memory, the actual bandwidth will be significantly reduced.

Table 44.64. RGB Mode Performance

RGB Mode	Pixels/Word Memory Burst Mode
12 bpp	2
16 bpp	2
18 bpp	1
18 bpp RGB PACKED	1.333
19 bpp	1
19 bpp PACKED	1.333
24 bpp	1
24 bpp PACKED	1.333
25 bpp	1
32 bpp	1

44.6.17.3.YCbCr Mode Fetch Performance

Table 44.65. YCbCr Planar Components Capacity

YCbCr Mode	Comp/Word Memory Burst Mode
16 bpp 4:2:2 semiplanar	4 Y, 2 UV
16 bpp 4:2:2 planar	4 Y, 4 U, 4 V
12 bpp 4:2:0 semiplanar	4 Y, 2 UV
12 bpp 4:2:0 planar	4 Y, 4 U, 4 V

Table 44.66. YCbCr Planar Overall Capacity

YCbCr Mode	Pix/Word Memory Burst Mode
16 bpp 4:2:2 semiplanar	2
16 bpp 4:2:2 planar	2
12 bpp 4:2:0 semiplanar	2.66
12 bpp 4:2:0 planar	2.66

44.6.18. Input FIFO

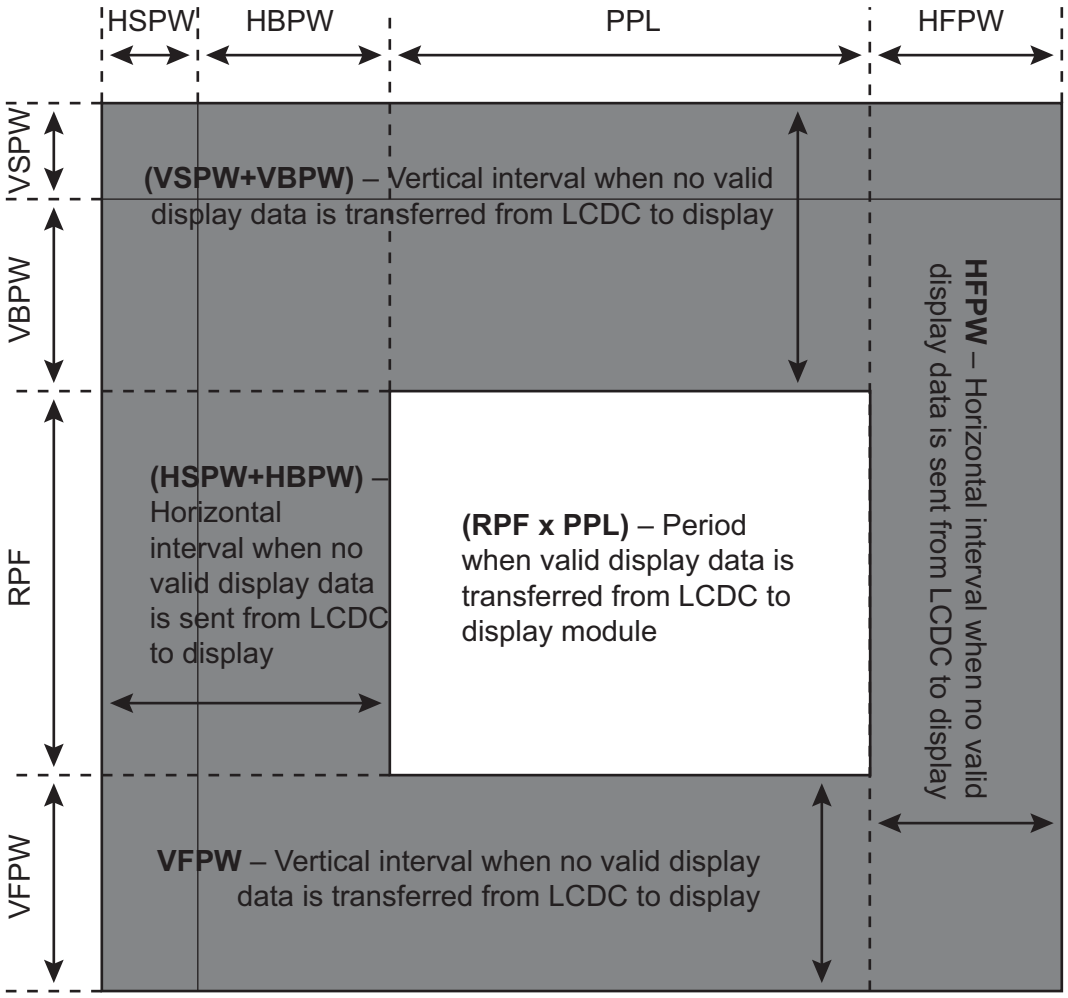
The LCDC includes one input FIFO per layer. These input FIFOs are used to buffer the system bus burst and serialize the stream of pixels.

44.6.19. Output FIFO

The LCDC includes one output FIFO that stores the blended pixel.

44.6.20. Output Timing Generation

Figure 44.24. Frame Transmission Overview



44.6.20.1.Active Display Timing Mode

Figure 44.25. Active Display Timing

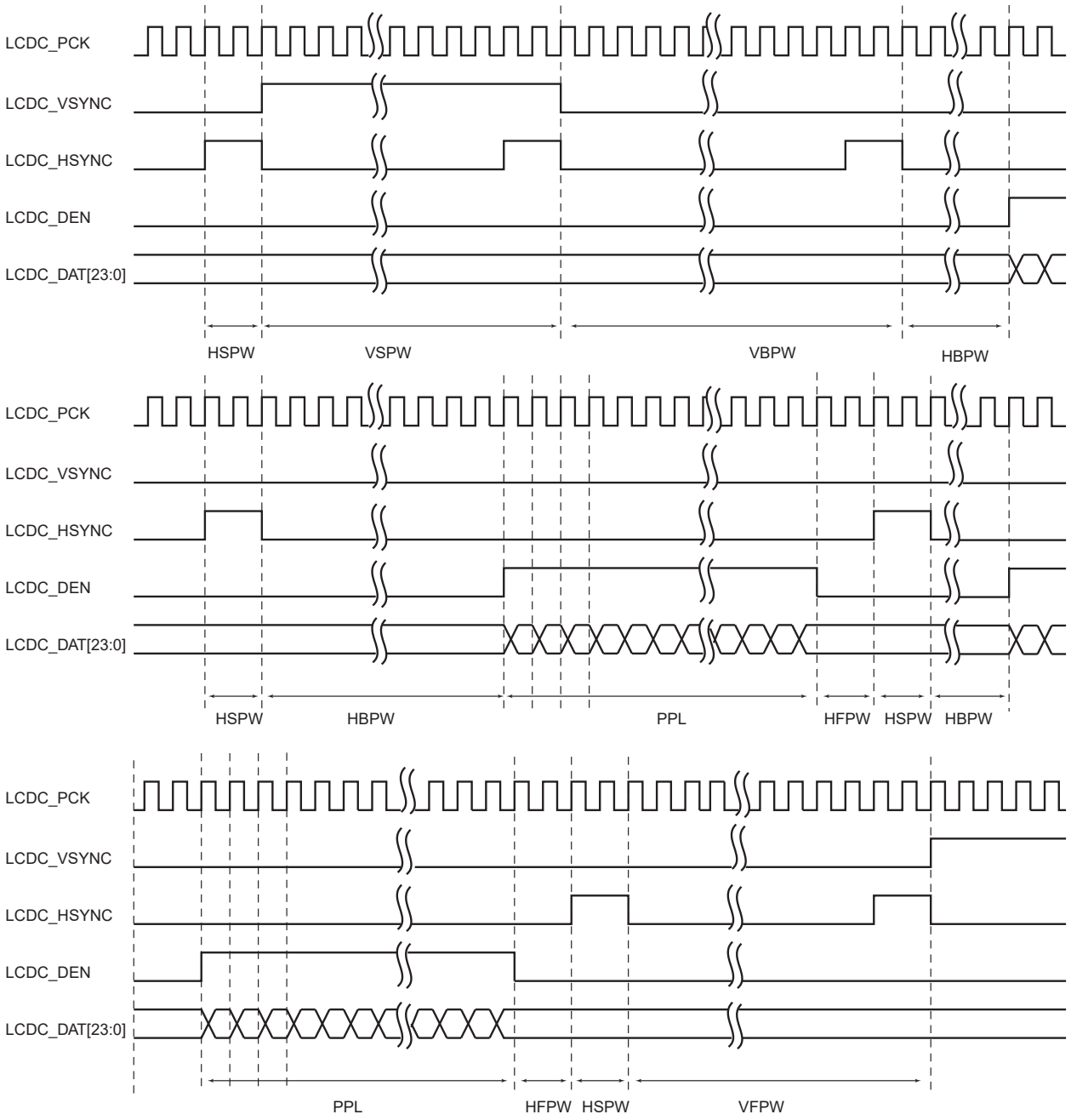


Figure 44.26. Vertical Synchronization Timing (part 1)

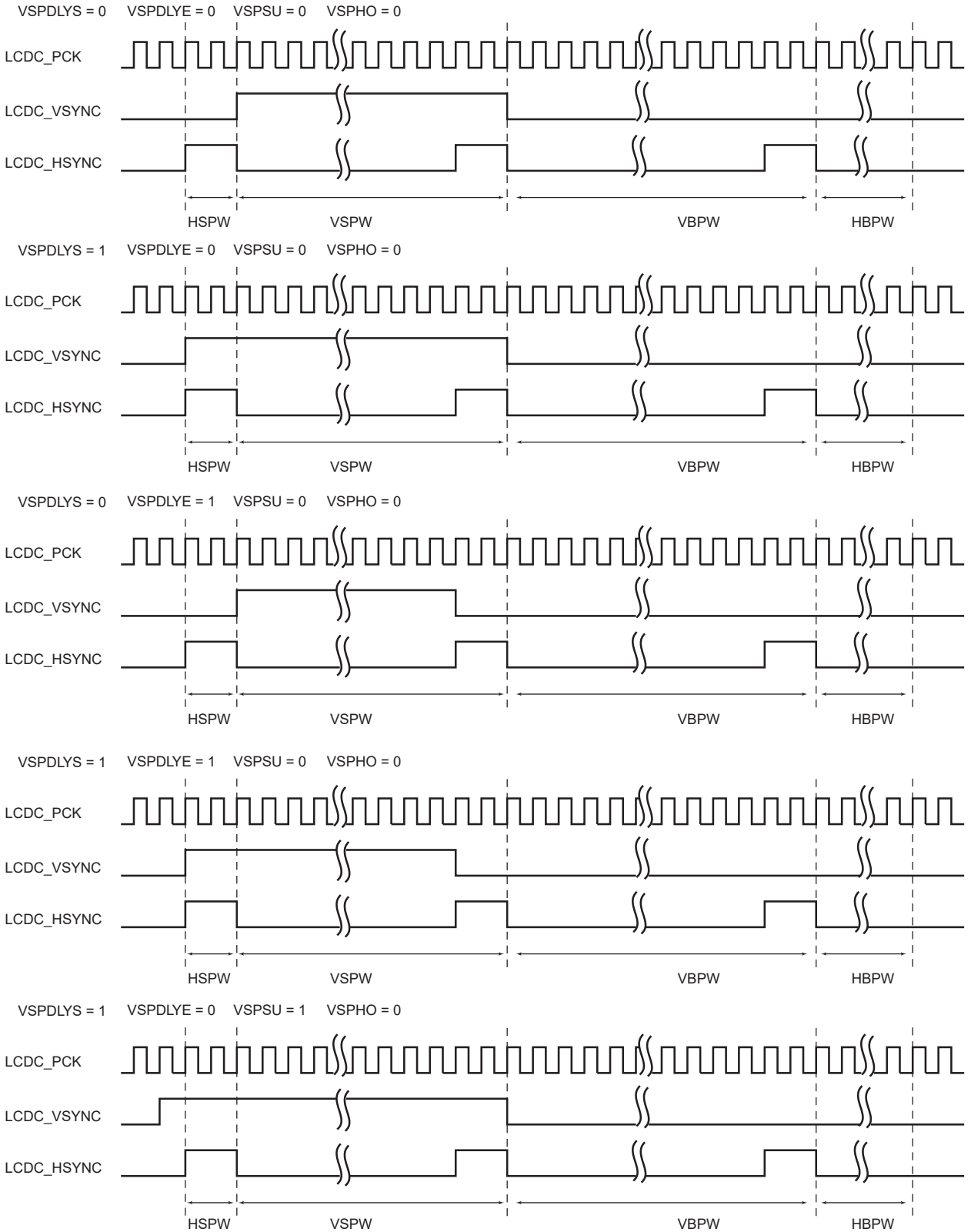


Figure 44.27. Vertical Synchronization Timing (part 2)

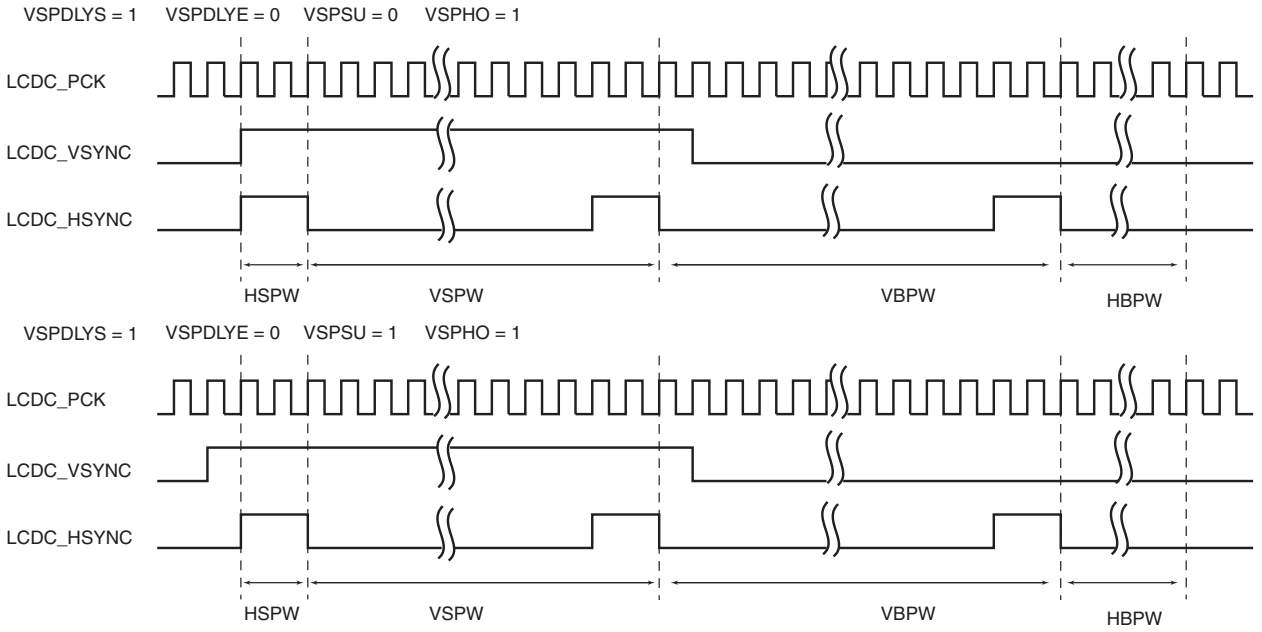
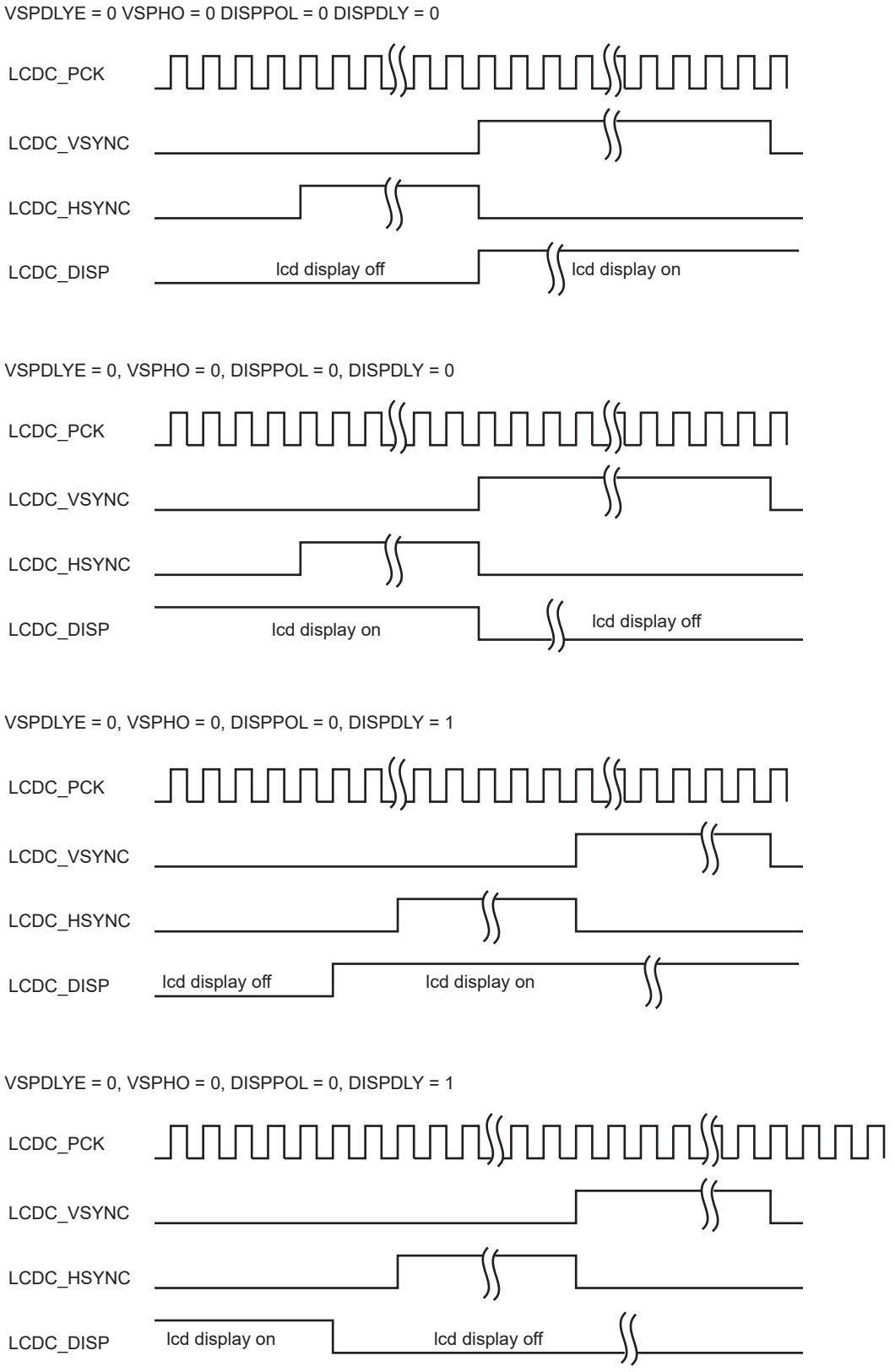


Figure 44.28. DISP Signal Timing Diagram



44.6.21. Output Format

44.6.21.1.Active Mode Output Pin Assignment

The LCDC supports classical and MIPI DPI standard RGB output modes. When operating on a 24-bit bus interface, if LCDC_LCDCFG5.DPI = 0, classical output modes are used (see Table 44.67). If LCDC_LCDCFG5.DPI = 1, DPI output modes are used (see Table 44.68).

Table 44.67. Classical Output Modes with 24-bit Bus Interface Configuration (LCDC_LCDCFG5.DPI = 0)

Pin ID	24-bit Output	18-bit Output	16-bit Output	12-bit Output
LCDC_DAT[23]	R[7]	R[5]	R[4]	R[3]
LCDC_DAT[22]	R[6]	R[4]	R[3]	R[2]
LCDC_DAT[21]	R[5]	R[3]	R[2]	R[1]
LCDC_DAT[20]	R[4]	R[2]	R[1]	R[0]
LCDC_DAT[19]	R[3]	R[1]	R[0]	–
LCDC_DAT[18]	R[2]	R[0]	–	–
LCDC_DAT[17]	R[1]	–	–	–
LCDC_DAT[16]	R[0]	–	–	–
LCDC_DAT[15]	G[7]	G[5]	G[5]	G[3]
LCDC_DAT[14]	G[6]	G[4]	G[4]	G[2]
LCDC_DAT[13]	G[5]	G[3]	G[3]	G[1]
LCDC_DAT[12]	G[4]	G[2]	G[2]	G[0]
LCDC_DAT[11]	G[3]	G[1]	G[1]	–
LCDC_DAT[10]	G[2]	G[0]	G[0]	–
LCDC_DAT[9]	G[1]	–	–	–
LCDC_DAT[8]	G[0]	–	–	–
LCDC_DAT[7]	B[7]	B[5]	B[4]	B[3]
LCDC_DAT[6]	B[6]	B[4]	B[3]	B[2]
LCDC_DAT[5]	B[5]	B[3]	B[2]	B[1]
LCDC_DAT[4]	B[4]	B[2]	B[1]	B[0]
LCDC_DAT[3]	B[3]	B[1]	B[0]	–
LCDC_DAT[2]	B[2]	B[0]	–	–
LCDC_DAT[1]	B[1]	–	–	–
LCDC_DAT[0]	B[0]	–	–	–

Table 44.68. DPI Ouput Modes with 24-bit Bus Interface Configuration (LCDC_LCDCFG5.DPI = 1)

Pin ID	16-bit Output			18-bit Output	18-bit Output	24-bit Output
	Config 1	Config 2	Config 3	Config 1	Config 2	
LCDC_DAT[23]	–	–	–	–	–	R[7]
LCDC_DAT[22]	–	–	–	–	–	R[6]
LCDC_DAT[21]	–	–	R[4]	–	R[5]	R[5]
LCDC_DAT[20]	–	R[4]	R[3]	–	R[4]	R[4]
LCDC_DAT[19]	–	R[3]	R[2]	–	R[3]	R[3]
LCDC_DAT[18]	–	R[2]	R[1]	–	R[2]	R[2]
LCDC_DAT[17]	–	R[1]	R[0]	R[5]	R[1]	R[1]
LCDC_DAT[16]	–	R[0]	–	R[4]	R[0]	R[0]
LCDC_DAT[15]	R[4]	–	–	R[3]	–	G[7]
LCDC_DAT[14]	R[3]	–	–	R[2]	–	G[6]
LCDC_DAT[13]	R[2]	G[5]	G[5]	R[1]	G[5]	G[5]
LCDC_DAT[12]	R[1]	G[4]	G[4]	R[0]	G[4]	G[4]

Table 44.68. DPI Ouput Modes with 24-bit Bus Interface Configuration (LCDC_LCDCFG5.DPI = 1) (continued)

Pin ID	16-bit Output			18-bit Output	18-bit Output	24-bit Output
	Config 1	Config 2	Config 3	Config 1	Config 2	
LCDC_DAT[11]	R[0]	G[3]	G[3]	G[5]	G[3]	G[3]
LCDC_DAT[10]	G[2]	G[2]	G[2]	G[4]	G[2]	G[2]
LCDC_DAT[9]	G[1]	G[1]	G[1]	G[3]	G[1]	G[1]
LCDC_DAT[8]	G[0]	G[0]	G[0]	G[2]	G[0]	G[0]
LCDC_DAT[7]	B[7]	–	–	G[1]	–	B[7]
LCDC_DAT[6]	B[6]	–	–	G[0]	–	B[6]
LCDC_DAT[5]	B[5]	–	B[4]	B[5]	B[5]	B[5]
LCDC_DAT[4]	B[4]	B[4]	B[3]	B[4]	B[4]	B[4]
LCDC_DAT[3]	B[3]	B[3]	B[2]	B[3]	B[3]	B[3]
LCDC_DAT[2]	B[2]	B[2]	B[1]	B[2]	B[2]	B[2]
LCDC_DAT[1]	B[1]	B[1]	B[0]	B[1]	B[1]	B[1]
LCDC_DAT[0]	B[0]	B[0]	–	B[0]	B[0]	B[0]

44.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	LCDC_LCDCFG0	31:24								
		23:16	CLKDIV[7:0]							
		15:8								
		7:0					CLKPWMSEL		CLKBYP	CLKPOL
0x04	LCDC_LCDCFG1	31:24							VSPW[9:8]	
		23:16	VSPW[7:0]							
		15:8							HSPW[9:8]	
		7:0	HSPW[7:0]							
0x08	LCDC_LCDCFG2	31:24							VBPW[9:8]	
		23:16	VBPW[7:0]							
		15:8							VFPW[9:8]	
		7:0	VFPW[7:0]							
0x0C	LCDC_LCDCFG3	31:24							HBPW[9:8]	
		23:16	HBPW[7:0]							
		15:8							HFPW[9:8]	
		7:0	HFPW[7:0]							
0x10	LCDC_LCDCFG4	31:24						RPF[10:8]		
		23:16	RPF[7:0]							
		15:8						PPL[10:8]		
		7:0	PPL[7:0]							
0x14	LCDC_LCDCFG5	31:24								
		23:16	GUARDTIME[7:0]							
		15:8			VSPHO	VSPSU	DPI	MODE[2:0]		
		7:0	DISPDLY	DITHER		DISPPOL	VSPDLYE	VSPDLYS	VSPOL	HSPOL
0x18	LCDC_LCDCFG6	31:24								
		23:16								
		15:8	PWMCVAL[7:0]							
		7:0				PWMPOL	PWMPS[3:0]			
0x1C	LCDC_LCDCFG7	31:24								
		23:16								
		15:8						ROW[10:8]		
		7:0	ROW[7:0]							
0x20	LCDC_LCDEN	31:24								
		23:16								
		15:8								
		7:0		CMEN	SDEN		PWMEN	DISPEN	SYNCEN	CLKEN
0x24	LCDC_LCDDIS	31:24								
		23:16								
		15:8					PWMRST	DISPRST	SYNCRST	CLKRST
		7:0		CMDIS	SDDIS		PWMDIS	DISPDIS	SYNCDIS	CLKDIS
0x28	LCDC_LCDSR	31:24								
		23:16								
		15:8								
		7:0		CMSTS	SDSTS	SIPSTS	PWMSTS	DISPSTS	LCDSTS	CLKSTS
0x2C	LCDC_LCDIER	31:24	WPIE							
		23:16								
		15:8					HEOIE	OVR2IE	OVR1IE	BASEIE
		7:0				FIFOERRIE	ROWIE	DISPIE	DISIE	SOFIE
0x30	LCDC_LCDIDR	31:24	WPID							
		23:16								
		15:8					HEOID	OVR2ID	OVR1ID	BASEID
		7:0				FIFOERRID	ROWID	DISPID	DISID	SOFID
0x34	LCDC_LCDIMR	31:24	WPIM							
		23:16								
		15:8					HEOIM	OVR2IM	OVR1IM	BASEIM
		7:0				FIFOERRIM	ROWIM	DISPIM	DISIM	SOFIM

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	LCDC_LCDISR	31:24	WP							
		23:16								
		15:8					HEO	OVR2	OVR1	BASE
		7:0				FIFOERR	ROW	DISP	DIS	SOF
0x3C	LCDC_ATTRE	31:24								
		23:16								
		15:8					HEOCL	OVR2CL	OVR1CL	BASECL
		7:0					HEO	OVR2	OVR1	BASE
0x40	LCDC_ATTRS	31:24	SIP							
		23:16								
		15:8					HEOCL	OVR2CL	OVR1CL	BASECL
		7:0					HEO	OVR2	OVR1	BASE
0x44 ... 0x5F	Reserved									
0x60	LCDC_BASEIER	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x64	LCDC_BASEIDR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x68	LCDC_BASEIMR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x6C	LCDC_BASEISR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x70	LCDC_BASEEN	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x74	LCDC_BASECLA	31:24	CLA[29:22]							
		23:16	CLA[21:14]							
		15:8	CLA[13:6]							
		7:0	CLA[5:0]							
0x78	LCDC_BASEFBA	31:24	FBA[29:22]							
		23:16	FBA[21:14]							
		15:8	FBA[13:6]							
		7:0	FBA[5:0]							
0x7C	LCDC_BASECFG0	31:24								
		23:16								
		15:8								
		7:0		BLEN[2:0]						
0x80	LCDC_BASECFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]					GAM		CLUTEN
0x84	LCDC_BASECFG2	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x88	LCDC_BASECFG3	31:24								
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x8C	LCDC_BASECFG4	31:24								
		23:16								
		15:8								
		7:0				DISCEN			REP	DMA
0x90	LCDC_BASECFG5	31:24						DISCYPOS[10:8]		
		23:16	DISCYPOS[7:0]							
		15:8						DISCXPOS[10:8]		
		7:0	DISCXPOS[7:0]							
0x94	LCDC_BASECFG6	31:24						DISCYSIZE[10:8]		
		23:16	DISCYSIZE[7:0]							
		15:8						DISCXSIZE[10:8]		
		7:0	DISCXSIZE[7:0]							
0x98 ... 0xE3	Reserved									
0xE4	LCDC_WPMR	31:24	WPKEY[7:0]							
		23:16								
		15:8		HEWPCRE	HEWPITE	HEWPCFGE	O2WPCRE	O2WPITE	O2WPCFGE	O1WPCRE
		7:0	O1WPITE	O1WPCFGE	BWPCRE	BWPITE	BWPCFGE	WPCRE	WPITE	WPCFGE
0xE8	LCDC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0						SEQE	CGD	WPVS
0xEC ... 0x015F	Reserved									
0x0160	LCDC_OVR1IER	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0164	LCDC_OVR1IDR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0168	LCDC_OVR1IMR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x016C	LCDC_OVR1ISR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0170	LCDC_OVR1EN	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x0174	LCDC_OVR1CLA	31:24	CLA[29:22]							
		23:16	CLA[21:14]							
		15:8	CLA[13:6]							
		7:0	CLA[5:0]							
0x0178	LCDC_OVR1FBA	31:24	FBA[31:24]							
		23:16	FBA[23:16]							
		15:8	FBA[15:8]							
		7:0	FBA[7:0]							
0x017C	LCDC_OVR1CFG0	31:24								
		23:16								
		15:8								
		7:0		BLEN[2:0]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0180	LCDC_OVR1CFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]					GAM		CLUTEN
0x0184	LCDC_OVR1CFG2	31:24						YPOS[10:8]		
		23:16	YPOS[7:0]							
		15:8						XPOS[10:8]		
		7:0	XPOS[7:0]							
0x0188	LCDC_OVR1CFG3	31:24						YSIZE[10:8]		
		23:16	YSIZE[7:0]							
		15:8						XSIZE[10:8]		
		7:0	XSIZE[7:0]							
0x018C	LCDC_OVR1CFG4	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x0190	LCDC_OVR1CFG5	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							
0x0194	LCDC_OVR1CFG6	31:24	ADEF[7:0]							
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x0198	LCDC_OVR1CFG7	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x019C	LCDC_OVR1CFG8	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							
0x01A0	LCDC_OVR1CFG9	31:24	A1[7:0]							
		23:16	A0[7:0]							
		15:8	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[2]
		7:0	SFACTC[1:0]				DSTKEY	CRKEY	REP	DMA
0x01A4 ... 0x025F	Reserved									
0x0260	LCDC_OVR2IER	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0264	LCDC_OVR2IDR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0268	LCDC_OVR2IMR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x026C	LCDC_OVR2ISR	31:24								
		23:16								
		15:8								
		7:0						OVF	ERROR	END
0x0270	LCDC_OVR2EN	31:24								
		23:16								
		15:8								
		7:0								ENABLE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0274	LCDC_OVR2CLA	31:24	CLA[29:22]							
		23:16	CLA[21:14]							
		15:8	CLA[13:6]							
		7:0	CLA[5:0]							
0x0278	LCDC_OVR2FBA	31:24	FBA[31:24]							
		23:16	FBA[23:16]							
		15:8	FBA[15:8]							
		7:0	FBA[7:0]							
0x027C	LCDC_OVR2CFG0	31:24								
		23:16								
		15:8								
		7:0		BLEN[2:0]						
0x0280	LCDC_OVR2CFG1	31:24								
		23:16								
		15:8							CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]					GAM		CLUTEN
0x0284	LCDC_OVR2CFG2	31:24							YPOS[10:8]	
		23:16	YPOS[7:0]							
		15:8							XPOS[10:8]	
		7:0	XPOS[7:0]							
0x0288	LCDC_OVR2CFG3	31:24							YSIZE[10:8]	
		23:16	YSIZE[7:0]							
		15:8							XSIZE[10:8]	
		7:0	XSIZE[7:0]							
0x028C	LCDC_OVR2CFG4	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x0290	LCDC_OVR2CFG5	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							
0x0294	LCDC_OVR2CFG6	31:24	ADEF[7:0]							
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x0298	LCDC_OVR2CFG7	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x029C	LCDC_OVR2CFG8	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							
0x02A0	LCDC_OVR2CFG9	31:24	A1[7:0]							
		23:16	A0[7:0]							
		15:8	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[2]
		7:0	SFACTC[1:0]				DSTKEY	CRKEY	REP	DMA
0x02A4 ... 0x035F	Reserved									
0x0360	LCDC_HEOIER	31:24								
		23:16						CROVF	CRERROR	CREND
		15:8						CBOVF	CBERROR	CBEND
		7:0						OVF	ERROR	END
0x0364	LCDC_HEOIDR	31:24								
		23:16						CROVF	CRERROR	CREND
		15:8						CBOVF	CBERROR	CBEND
		7:0						OVF	ERROR	END

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0368	LCDC_HEOIMR	31:24								
		23:16						CROVF	CRERROR	CREND
		15:8						CBOVF	CBERROR	CBEND
		7:0						OVF	ERROR	END
0x036C	LCDC_HEOISR	31:24								
		23:16						CROVF	CRERROR	CREND
		15:8						CBOVF	CBERROR	CBEND
		7:0						OVF	ERROR	END
0x0370	LCDC_HEOEN	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x0374	LCDC_HEOCLA	31:24	CLA[29:22]							
		23:16	CLA[21:14]							
		15:8	CLA[13:6]							
		7:0	CLA[5:0]							
0x0378	LCDC_HEOYFBA0	31:24	FBA[31:24]							
		23:16	FBA[23:16]							
		15:8	FBA[15:8]							
		7:0	FBA[7:0]							
0x037C	LCDC_HEOCBFBA0	31:24	CBFBA[31:24]							
		23:16	CBFBA[23:16]							
		15:8	CBFBA[15:8]							
		7:0	CBFBA[7:0]							
0x0380	LCDC_HEOCRFBA0	31:24	CRFBA[31:24]							
		23:16	CRFBA[23:16]							
		15:8	CRFBA[15:8]							
		7:0	CRFBA[7:0]							
0x0384	LCDC_HEOYFBA1	31:24	FBA[31:24]							
		23:16	FBA[23:16]							
		15:8	FBA[15:8]							
		7:0	FBA[7:0]							
0x0388	LCDC_HEOCBFBA1	31:24	CBFBA[31:24]							
		23:16	CBFBA[23:16]							
		15:8	CBFBA[15:8]							
		7:0	CBFBA[7:0]							
0x038C	LCDC_HEOCRFBA1	31:24	CRFBA[31:24]							
		23:16	CRFBA[23:16]							
		15:8	CRFBA[15:8]							
		7:0	CRFBA[7:0]							
0x0390	LCDC_HEOCFG0	31:24								
		23:16								
		15:8						BLENCC[2:0]		
		7:0		BLEN[2:0]						
0x0394	LCDC_HEOCFG1	31:24								ILD
		23:16								YCC422ROT
		15:8	YCCMODE[3:0]						CLUTMODE[1:0]	
		7:0	RGBMODE[3:0]					GAM	YCCEN	CLUTEN
0x0398	LCDC_HEOCFG2	31:24						YPOS[10:8]		
		23:16	YPOS[7:0]							
		15:8						XPOS[10:8]		
		7:0	XPOS[7:0]							
0x039C	LCDC_HEOCFG3	31:24						YSIZE[10:8]		
		23:16	YSIZE[7:0]							
		15:8						XSIZE[10:8]		
		7:0	XSIZE[7:0]							
0x03A0	LCDC_HEOCFG4	31:24						YMEMSIZE[10:8]		
		23:16	YMEMSIZE[7:0]							
		15:8						XMEMSIZE[10:8]		
		7:0	XMEMSIZE[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03A4	LCDC_HEOCFG5	31:24	XSTRIDE[31:24]							
		23:16	XSTRIDE[23:16]							
		15:8	XSTRIDE[15:8]							
		7:0	XSTRIDE[7:0]							
0x03A8	LCDC_HEOCFG6	31:24	PSTRIDE[31:24]							
		23:16	PSTRIDE[23:16]							
		15:8	PSTRIDE[15:8]							
		7:0	PSTRIDE[7:0]							
0x03AC	LCDC_HEOCFG7	31:24	CCXSTRIDE[31:24]							
		23:16	CCXSTRIDE[23:16]							
		15:8	CCXSTRIDE[15:8]							
		7:0	CCXSTRIDE[7:0]							
0x03B0	LCDC_HEOCFG8	31:24	CCPSTRIDE[31:24]							
		23:16	CCPSTRIDE[23:16]							
		15:8	CCPSTRIDE[15:8]							
		7:0	CCPSTRIDE[7:0]							
0x03B4	LCDC_HEOCFG9	31:24	ADEF[7:0]							
		23:16	RDEF[7:0]							
		15:8	GDEF[7:0]							
		7:0	BDEF[7:0]							
0x03B8	LCDC_HEOCFG10	31:24								
		23:16	RKEY[7:0]							
		15:8	GKEY[7:0]							
		7:0	BKEY[7:0]							
0x03BC	LCDC_HEOCFG11	31:24								
		23:16	RMASK[7:0]							
		15:8	GMASK[7:0]							
		7:0	BMASK[7:0]							
0x03C0	LCDC_HEOCFG12	31:24	A1[7:0]							
		23:16	A0[7:0]							
		15:8	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[2]
		7:0	SFACTC[1:0]		VIDPRI		DSTKEY	CRKEY	REP	DMA
0x03C4	LCDC_HEOCFG13	31:24								
		23:16								
		15:8								
		7:0				BL				ENABLE
0x03C8	LCDC_HEOCFG14	31:24					CONT[11:8]			
		23:16	CONT[7:0]							
		15:8						BRIGHT[10:8]		
		7:0	BRIGHT[7:0]							
0x03CC	LCDC_HEOCFG15	31:24					SAT[11:8]			
		23:16	SAT[7:0]							
		15:8								HUE[8]
		7:0	HUE[7:0]							
0x03D0	LCDC_HEOCFG16	31:24				RCBGAIN[12:8]				
		23:16	RCBGAIN[7:0]							
		15:8				RYGAIN[12:8]				
		7:0	RYGAIN[7:0]							
0x03D4	LCDC_HEOCFG17	31:24								
		23:16								
		15:8				RCRGAIN[12:8]				
		7:0	RCRGAIN[7:0]							
0x03D8	LCDC_HEOCFG18	31:24				GCBGAIN[12:8]				
		23:16	GCBGAIN[7:0]							
		15:8				GYGAIN[12:8]				
		7:0	GYGAIN[7:0]							
0x03DC	LCDC_HEOCFG19	31:24								
		23:16								
		15:8				GCRGAIN[12:8]				
		7:0	GCRGAIN[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x03E0	LCDC_HEOCFG20	31:24				BCBGAIN[12:8]				
		23:16	BCBGAIN[7:0]							
		15:8				BYGAIN[12:8]				
		7:0	BYGAIN[7:0]							
0x03E4	LCDC_HEOCFG21	31:24								
		23:16								
		15:8				BCRGAIN[12:8]				
		7:0	BCRGAIN[7:0]							
0x03E8	LCDC_HEOCFG22	31:24								
		23:16								
		15:8								
		7:0						CROFF	CBOFF	YOFF
0x03EC	LCDC_HEOCFG23	31:24								
		23:16								
		15:8								
		7:0			HXSCEN	HXSYEN			VXSCEN	VXSYEN
0x03F0	LCDC_HEOCFG24	31:24								
		23:16	VXSYFACT[23:16]							
		15:8	VXSYFACT[15:8]							
		7:0	VXSYFACT[7:0]							
0x03F4	LCDC_HEOCFG25	31:24								
		23:16	VXSCFACT[23:16]							
		15:8	VXSCFACT[15:8]							
		7:0	VXSCFACT[7:0]							
0x03F8	LCDC_HEOCFG26	31:24								
		23:16	HXSYFACT[23:16]							
		15:8	HXSYFACT[15:8]							
		7:0	HXSYFACT[7:0]							
0x03FC	LCDC_HEOCFG27	31:24								
		23:16	HXSCFACT[23:16]							
		15:8	HXSCFACT[15:8]							
		7:0	HXSCFACT[7:0]							
0x0400	LCDC_HEOCFG28	31:24					VXSCOFF1[3:0]			
		23:16					VXSCOFF[3:0]			
		15:8					VXSYOFF1[3:0]			
		7:0					VXSYOFF[3:0]			
0x0404	LCDC_HEOCFG29	31:24								
		23:16					HXSCOFF[3:0]			
		15:8								
		7:0					HXSYOFF[3:0]			
0x0408	LCDC_HEOCFG30	31:24							VXSCCFG1[1:0]	
		23:16		VXSCBICU	VXSC1201N	VXSCTAP2			VXSCCFG[1:0]	
		15:8							VXSYCFG1[1:0]	
		7:0		VXSYBICU	VXSY1201N	VXSYTAP2			VXSYCFG[1:0]	
0x040C	LCDC_HEOCFG31	31:24								
		23:16			HXSCBICU	HXSCTAP2			HXSCCFG[1:0]	
		15:8								
		7:0			HXSYBICU	HXSYTAP2			HXSYCFG[1:0]	
0x0410	LCDC_HEOVTAP10P0	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							
0x0414	LCDC_HEOVTAP32P0	31:24				TAP3[12:8]				
		23:16	TAP3[7:0]							
		15:8				TAP2[12:8]				
		7:0	TAP2[7:0]							
0x0418	LCDC_HEOVTAP10P1	31:24				TAP1[12:8]				
		23:16	TAP1[7:0]							
		15:8				TAP0[12:8]				
		7:0	TAP0[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x041C	LCDC_HEOVTAP32P1	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0420	LCDC_HEOVTAP10P2	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x0424	LCDC_HEOVTAP32P2	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0428	LCDC_HEOVTAP10P3	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x042C	LCDC_HEOVTAP32P3	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0430	LCDC_HEOVTAP10P4	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x0434	LCDC_HEOVTAP32P4	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0438	LCDC_HEOVTAP10P5	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x043C	LCDC_HEOVTAP32P5	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0440	LCDC_HEOVTAP10P6	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x0444	LCDC_HEOVTAP32P6	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0448	LCDC_HEOVTAP10P7	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x044C	LCDC_HEOVTAP32P7	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0450	LCDC_HEOVTAP10P8	31:24						TAP1[12:8]		
		23:16	TAP1[7:0]							
		15:8						TAP0[12:8]		
		7:0	TAP0[7:0]							
0x0454	LCDC_HEOVTAP32P8	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0458	LCDC_HEOVTAP10P9	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x045C	LCDC_HEOVTAP32P9	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0460	LCDC_HEOVTAP10P10	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x0464	LCDC_HEOVTAP32P10	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0468	LCDC_HEOVTAP10P11	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x046C	LCDC_HEOVTAP32P11	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0470	LCDC_HEOVTAP10P12	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x0474	LCDC_HEOVTAP32P12	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0478	LCDC_HEOVTAP10P13	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x047C	LCDC_HEOVTAP32P13	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0480	LCDC_HEOVTAP10P14	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x0484	LCDC_HEOVTAP32P14	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0488	LCDC_HEOVTAP10P15	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x048C	LCDC_HEOVTAP32P15	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0490	LCDC_HEOHTAP10P0	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0494	LCDC_HEOHTAP32P0	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0498	LCDC_HEOHTAP10P1	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x049C	LCDC_HEOHTAP32P1	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04A0	LCDC_HEOHTAP10P2	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04A4	LCDC_HEOHTAP32P2	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04A8	LCDC_HEOHTAP10P3	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04AC	LCDC_HEOHTAP32P3	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04B0	LCDC_HEOHTAP10P4	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04B4	LCDC_HEOHTAP32P4	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04B8	LCDC_HEOHTAP10P5	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04BC	LCDC_HEOHTAP32P5	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04C0	LCDC_HEOHTAP10P6	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04C4	LCDC_HEOHTAP32P6	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04C8	LCDC_HEOHTAP10P7	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04CC	LCDC_HEOHTAP32P7	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x04D0	LCDC_HEOHTAP10P8	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04D4	LCDC_HEOHTAP32P8	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04D8	LCDC_HEOHTAP10P9	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04DC	LCDC_HEOHTAP32P9	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04E0	LCDC_HEOHTAP10P10	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04E4	LCDC_HEOHTAP32P10	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04E8	LCDC_HEOHTAP10P11	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04EC	LCDC_HEOHTAP32P11	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04F0	LCDC_HEOHTAP10P12	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04F4	LCDC_HEOHTAP32P12	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x04F8	LCDC_HEOHTAP10P13	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x04FC	LCDC_HEOHTAP32P13	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0500	LCDC_HEOHTAP10P14	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			
0x0504	LCDC_HEOHTAP32P14	31:24						TAP3[12:8]		
		23:16					TAP3[7:0]			
		15:8						TAP2[12:8]		
		7:0					TAP2[7:0]			
0x0508	LCDC_HEOHTAP10P15	31:24						TAP1[12:8]		
		23:16					TAP1[7:0]			
		15:8						TAP0[12:8]		
		7:0					TAP0[7:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x050C	LCDC_HEOHTAP32P15	31:24						TAP3[12:8]		
		23:16	TAP3[7:0]							
		15:8						TAP2[12:8]		
		7:0	TAP2[7:0]							
0x0510 ... 0x06FF	Reserved									
0x0700	LCDC_BASECLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x0AFC	LCDC_BASECLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x0B00	LCDC_OVR1CLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x0EFC	LCDC_OVR1CLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x0F00	LCDC_OVR2CLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x12FC	LCDC_OVR2CLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
0x1300	LCDC_HEOCLUT0	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							
...										
0x16FC	LCDC_HEOCLUT255	31:24	ACLUT[7:0]							
		23:16	RCLUT[7:0]							
		15:8	GCLUT[7:0]							
		7:0	BCLUT[7:0]							

44.7.1. LCDC Configuration Register 0

Name: LCDC_LCDCFG0
Offset: 0x00000000
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	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	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CLKPWMSEL		CLKBYP	CLKPOL
Access					R/W		R/W	R/W
Reset					0		0	0

Bits 23:16 – CLKDIV[7:0] LCDC Clock Divider
8-bit width clock divider for pixel clock (LCDC_PCK). If CLKBYP = 0, the pixel clock frequency formula is:
LCDC_PCK_freq = GCLK_freq / (CLKDIV+2)

Bit 3 – CLKPWMSEL LCDC PWM Clock Source Selection

Value	Description
0	The slow clock is selected and feeds the PWM module.
1	The peripheral clock (MCK) is selected and feeds the PWM module.

Bit 1 – CLKBYP LCDC Pixel Clock Divider Bypass

Value	Description
0	Pixel clock divider is not bypassed. LCDC_PCK is defined with CLKDIV parameter.
1	Pixel clock divider is bypassed. LCDC_PCK = GCLK clock

Bit 0 – CLKPOL LCDC Clock Polarity

Value	Description
0	Data/Control signals are launched on the rising edge of the pixel clock.
1	Data/Control signals are launched on the falling edge of the pixel clock.

44.7.2. LCDC Configuration Register 1

Name: LCDC_LCDCFG1
Offset: 0x00000004
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
							VSPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VSPW[7:0]							
Access	R/W	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
							HSPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	HSPW[7:0]							
Access	R/W	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 – VSPW[9:0] Vertical Synchronization Pulse Width
Width of the LCDC_VSYNC pulse, given in number of lines. Width is (VSPW+1) lines.

Bits 9:0 – HSPW[9:0] Horizontal Synchronization Pulse Width
Width of the LCDC_HSYNC pulse, given in pixel clock cycles. Width is (HSPW+1) LCDC_PCK cycles.

44.7.3. LCDC Configuration Register 2

Name: LCDC_LCDCFG2
Offset: 0x00000008
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
							VBPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VBPW[7:0]							
Access	R/W	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
							VFPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	VFPW[7:0]							
Access	R/W	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 – VBPW[9:0] Vertical Back Porch Width
Indicates the number of lines at the beginning of the frame. The blanking interval is equal to (VBPW+1) lines.

Bits 9:0 – VFPW[9:0] Vertical Front Porch Width
Indicates the number of lines at the end of the frame. The blanking interval is equal to (VFPW+1) lines.

44.7.4. LCDC Configuration Register 3

Name: LCDC_LCDCFG3
Offset: 0x0000000C
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
							HBPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	HBPW[7:0]							
Access	R/W	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
							HFPW[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	HFPW[7:0]							
Access	R/W	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 – HBPW[9:0] Horizontal Back Porch Width
Number of pixel clock cycles inserted at the beginning of the line. The interval is equal to (HBPW+1) LCDC_PCK cycles.

Bits 9:0 – HFPW[9:0] Horizontal Front Porch Width
Number of pixel clock cycles inserted at the end of the active line. The interval is equal to (HFPW+1) LCDC_PCK cycles.

44.7.5. LCDC Configuration Register 4

Name: LCDC_LCDCFG4
Offset: 0x00000010
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
						RPF[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	RPF[7:0]							
Access	R/W	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
						PPL[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	PPL[7:0]							
Access	R/W	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 – RPF[10:0] Number of Active Row Per Frame
Number of active lines in the frame. The frame height is equal to (RPF+1) lines.

Bits 10:0 – PPL[10:0] Number of Pixels Per Line
Number of pixels in the frame. The number of active pixels in the frame is equal to (PPL+1) pixels.

44.7.6. LCDC Configuration Register 5

Name: LCDC_LCDCFG5
Offset: 0x00000014
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this register.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	GUARDTIME[7:0]							
Access	R/W	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
			VSPHO	VSPSU	DPI	MODE[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
	DISPDLY	DITHER		DISPPOL	VSPDLYE	VSPDLYS	VSPOL	HSPOL
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

Bits 23:16 – GUARDTIME[7:0] LCD Display Guard Time
Number of frames inserted during start-up before LCDC_DISP signal is enabled, and after LCDC_DISP signal is disabled.

Bit 13 – VSPHO LCDC Vertical Synchronization Pulse Hold Configuration

Value	Description
0	The vertical synchronization pulse is asserted synchronously with horizontal pulse edge.
1	The vertical synchronization pulse is held active one pixel clock cycle after the horizontal pulse.

Bit 12 – VSPSU LCDC Vertical Synchronization Pulse Setup Configuration

Value	Description
0	The vertical synchronization pulse is asserted synchronously with horizontal pulse edge.
1	The vertical synchronization pulse is asserted one pixel clock cycle before the horizontal pulse.

Bit 11 – DPI Display Pixel Interface Compatible Mode

Value	Description
0	Legacy pixel mapping.
1	Activates the DPI compliant pixel stream. See field MODE: LCDC Output Mode.

Bits 10:8 – MODE[2:0] LCDC Output Mode
When DPI = 0:

Value	Name	Description
0	OUTPUT_12BPP	LCD Output mode is set to 12 bits per pixel
1	OUTPUT_16BPP	LCD Output mode is set to 16 bits per pixel
2	OUTPUT_18BPP	LCD Output mode is set to 18 bits per pixel
3	OUTPUT_24BPP	LCD Output mode is set to 24 bits per pixel

When DPI = 1:

Value	Name	Description
0	OUTPUT_DPI_16BPPCFG1	LCD Output mode is set to 16 bits per pixel Configuration 1
1	OUTPUT_DPI_16BPPCFG2	LCD Output mode is set to 1 bits per pixel Configuration 2
2	OUTPUT_DPI_16BPPCFG3	LCD Output mode is set to 16 bits per pixel Configuration 3
3	OUTPUT_DPI_18BPPCFG1	LCD Output mode is set to 18 bits per pixel Configuration 1
4	OUTPUT_DPI_18BPPCFG2	LCD Output mode is set to 18 bits per pixel Configuration 2
5	OUTPUT_DPI_24BPP	LCD Output mode is set to 24 bits per pixel

Bit 7 – DISPDLY LCDC Display Power Signal Synchronization

Value	Description
0	The LCDC_DISP signal is asserted synchronously with the second active edge of the horizontal pulse.
1	The LCDC_DISP signal is asserted asynchronously with both edges of the horizontal pulse.

Bit 6 – DITHER LCDC Dithering

Value	Description
0	Dithering logical unit is disabled.
1	Dithering logical unit is activated.

Bit 4 – DISPPOL Display Signal Polarity

Value	Description
0	Active high.
1	Active low.

Bit 3 – VSPDLYE Vertical Synchronization Pulse End

Value	Description
0	The second active edge of the vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.
1	The second active edge of the vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

Bit 2 – VSPDLYS Vertical Synchronization Pulse Start

Value	Description
0	The first active edge of the vertical synchronization pulse is synchronous with the second edge of the horizontal pulse.
1	The first active edge of the vertical synchronization pulse is synchronous with the first edge of the horizontal pulse.

Bit 1 – VSPOL Vertical Synchronization Pulse Polarity

Value	Description
0	Active high.
1	Active low.

Bit 0 – HSPOL Horizontal Synchronization Pulse Polarity

Value	Description
0	Active high.
1	Active low.

44.7.7. LCDC Configuration Register 6

Name: LCDC_LCDCFG6
Offset: 0x00000018
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this 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
	PWMCVAL[7:0]							
Access	R/W	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
				PWMPOL	PWMPS[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bits 15:8 – PWMCVAL[7:0] LCDC PWM Compare Value
PWM compare value. Used to adjust the analog value obtained after an external filter to control the contrast of the display.

Bit 4 – PWMPOL LCDC PWM Signal Polarity
Defines the polarity of the PWM output signal.

Value	Description
0	The output pulses are low level.
1	The output pulses are high level (the output is high whenever the value in the counter is less than the value PWMCVAL).

Bits 3:0 – PWMPS[3:0] PWM Clock Prescaler
Selects the configuration of the counter prescaler module.

Value	Name	Description
0	DIV_1	The counter advances at a rate of $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK}$
1	DIV_2	The counter advances at a rate of $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/2}$
2	DIV_4	The counter advances at a rate of $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/4}$
3	DIV_8	The counter advances at a rate of $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/8}$
4	DIV_16	The counter advances at a rate of $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/16}$
5	DIV_32	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/32}$
6	DIV_64	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/64}$
7	DIV_128	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/128}$
8	DIV_256	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/256}$

Value	Name	Description
9	DIV_512	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/512}$
10	DIV_1024	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/1024}$
11	DIV_2048	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/2048}$
12	DIV_4096	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/4096}$
13	DIV_8192	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/8192}$
14	DIV_16384	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/16384}$
15	DIV_32768	The counter advances at a of rate $f_{COUNTER} = f_{PWM\_SELECTED\_CLOCK/32768}$

44.7.8. LCDC Configuration Register 7

Name: LCDC_LCDCFG7
Offset: 0x0000001C
Reset: 0x00000000
Property: Read/Write

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

GCLK must be running before writing in this 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
						ROW[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	ROW[7:0]							
Access	R/W	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 – ROW[10:0] Row Identifier For Row Interrupt Signal
When the LCDC timing engine row pointer reaches the field ROW, an interrupt is triggered.
Indicates a line in reverse order, i.e., ROW0 is the last line and ROW height-1 the first line displayed.

44.7.9. LCDC Enable Register

Name: LCDC_LCDEN
Offset: 0x00000020
Reset: –
Property: Write-only

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

GCLK must be running before writing in this 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
		CMEN	SDEN		PWMEN	DISPEN	SYNCEN	CLKEN
Access		W	W		W	W	W	W
Reset		–	–		–	–	–	–

Bit 6 – CMEN Color Mode Signal Enable

Value	Description
0	No effect.
1	Sets the color mode signal (lcd_cm) to one. If a rising edge is generated, signals the MIPI DSI host to send a “Color Mode On” command to the LCD screen when the MIPI output interface is selected.

Bit 5 – SDEN Shutdown Signal Enable

Value	Description
0	No effect.
1	Sets the shutdown signal (lcd_sd) to zero (turns on the display).

Bit 3 – PWMEN LCDC Pulse Width Modulation Enable

Value	Description
0	No effect.
1	PWM is enabled.

Bit 2 – DISPEN LCDC DISP Signal Enable

Value	Description
0	No effect.
1	LCDC_DISP signal is generated.

Bit 1 – SYNCEN LCDC Horizontal and Vertical Synchronization Enable

Value	Description
0	No effect.
1	Both horizontal and vertical synchronization (LCD_VSYNC and LCD_HSYNC) signals are generated.

Bit 0 – CLKEN LCDC Pixel Clock Enable

Value	Description
0	No effect.
1	Pixel clock logical unit is activated.

44.7.10. LCDC Disable Register

Name: LCDC_LCDDIS
Offset: 0x00000024
Reset: –
Property: Write-only

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

GCLK must be running before writing in this 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					PWMRST	DISPRST	SYNCRST	CLKRST
Reset					W	W	W	W
					–	–	–	–

Bit	7	6	5	4	3	2	1	0
Access		CMDIS	SDDIS		PWMDIS	DISPDIS	SYNCDIS	CLKDIS
Reset		W	W		W	W	W	W
		–	–		–	–	–	–

Bit 11 – PWMRST LCDC PWM Reset

Value	Description
0	No effect.
1	Resets the PWM module immediately. The duty cycle may be violated.

Bit 10 – DISPRST LCDC DISP Signal Reset

Value	Description
0	No effect.
1	Resets the DISP signal immediately.

Bit 9 – SYNCRST LCDC Horizontal and Vertical Synchronization Reset

Value	Description
0	No effect.
1	Resets the timing engine immediately. The horizontal and vertical pulse widths are both violated.

Bit 8 – CLKRST LCDC Clock Reset

Value	Description
0	No effect.
1	Resets the pixel clock generator module immediately. The pixel clock duty cycle may be violated.

Bit 6 – CMDIS Color Mode Signal Disable

Value	Description
0	No effect.
1	Sets the color mode signa (lcd_cm) to one. If a falling edge is generated, signals the MIPI DSI host to send a “Color Mode Off” command to the LCD screen when the MIPI output interface is selected).

Bit 5 – SDDIS Shutdown Signal Disable

Value	Description
0	No effect.
1	Sets the shutdown signal (lcd_sd) to one (turns off the display).

Bit 3 – PWMDIS LCDC Pulse Width Modulation Disable

Value	Description
0	No effect.
1	Disables the pulse width modulation signal after the end of the frame.

Bit 2 – DISPDIS LCDC DISP Signal Disable

Value	Description
0	No effect.
1	Disables the DISP signal after the end of the frame.

Bit 1 – SYNCDIS LCDC Horizontal and Vertical Synchronization Disable

Value	Description
0	No effect.
1	Disables the synchronization signals after the end of the frame.

Bit 0 – CLKDIS LCDC Pixel Clock Disable

Value	Description
0	No effect.
1	Disables the pixel clock after the end of the frame.

44.7.11. LCDC Status Register

Name: LCDC_LCDSR
Offset: 0x00000028
Reset: 0x00000020
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
		CMSTS	SDSTS	SIPSTS	PWMSTS	DISPSTS	LCDSTS	CLKSTS
Access		R	R	R	R	R	R	R
Reset		0	1	0	0	0	0	0

Bit 6 – CMSTS Color Mode Signal Status

Value	Description
0	Color mode signal output is zero.
1	Color mode signal output is one.

Bit 5 – SDSTS Shutdown Signal Status

Value	Description
0	Shutdown signal output is zero.
1	Shutdown signal output is one.

Bit 4 – SIPSTS Synchronization In Progress

Value	Description
0	Clock domain synchronization is terminated.
1	Synchronization is in progress. Access to the registers LCDC_LCDCCFG[0..7], LCDC_LCDEN and LCDC_LCDDIS has no effect.

Bit 3 – PWMSTS LCDC PWM Signal Status

Value	Description
0	PWM is disabled.
1	PWM signal is activated.

Bit 2 – DISPSTS LCDC DISP Signal Status

Value	Description
0	DISP is disabled.

Value	Description
1	DISP signal is activated.

Bit 1 – LCDSTS LCDC Synchronization Status

Value	Description
0	Timing engine is disabled.
1	Timing engine is running.

Bit 0 – CLKSTS Clock Status

Value	Description
0	Pixel clock is disabled.
1	Pixel clock is running.

44.7.12. LCDC Interrupt Enable Register

Name: LCDC_LCDIER
Offset: 0x0000002C
Reset: –
Property: Write-only

This register can only be written if WPITE is cleared in the LCDC_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
	WPIE							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HEOIE	OVR2IE	OVR1IE	BASEIE
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
				FIFOERRIE	ROWIE	DISPIE	DISIE	SOFIE
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 31 – WPIE Write Protection Interrupt Enable

Bit 11 – HEOIE High-End Overlay Interrupt Enable

Bit 10 – OVR2IE Overlay 2 Interrupt Enable

Bit 9 – OVR1IE Overlay 1 Interrupt Enable

Bit 8 – BASEIE Base Layer Interrupt Enable

Bit 4 – FIFOERRIE Output FIFO Error Interrupt Enable

Bit 3 – ROWIE Row Interrupt Enable

Bit 2 – DISPIE Power-up/Power-down Sequence Terminated Interrupt Enable

Bit 1 – DISIE LCD Disable Interrupt Enable

Bit 0 – SOFIE Start of Frame Interrupt Enable

44.7.13. LCDC Interrupt Disable Register

Name: LCDC_LCDIDR
Offset: 0x00000030
Reset: –
Property: Write-only

This register can only be written if WPITE is cleared in the LCDC_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
	WPID							
Access	W							
Reset	–							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HEOID	OVR2ID	OVR1ID	BASEID
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
				FIFOERRID	ROWID	DISPID	DISID	SOFID
Access				W	W	W	W	W
Reset				–	–	–	–	–

Bit 31 – WPID Write Protection Interrupt Disable

Bit 11 – HEOID High-End Overlay Interrupt Disable

Bit 10 – OVR2ID Overlay 2 Interrupt Disable

Bit 9 – OVR1ID Overlay 1 Interrupt Disable

Bit 8 – BASEID Base Layer Interrupt Disable

Bit 4 – FIFOERRID Output FIFO Error Interrupt Disable

Bit 3 – ROWID Row Interrupt Disable

Bit 2 – DISPID Power-up/Power-down Sequence Terminated Interrupt Disable

Bit 1 – DISID LCD Disable Interrupt Disable

Bit 0 – SOFID Start of Frame Interrupt Disable

44.7.14. LCDC Interrupt Mask Register

Name: LCDC_LCDIMR
Offset: 0x00000034
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
	WPIM							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HEOIM	OVR2IM	OVR1IM	BASEIM
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIFOERRIM	ROWIM	DISPIM	DISIM	SOFIM
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – WPIM Write Protection Interrupt Mask

Bit 11 – HEOIM High-End Overlay Interrupt Mask

Bit 10 – OVR2IM Overlay 2 Interrupt Mask

Bit 9 – OVR1IM Overlay 1 Interrupt Mask

Bit 8 – BASEIM Base Layer Interrupt Mask

Bit 4 – FIFOERRIM Output FIFO Error Interrupt Mask

Bit 3 – ROWIM Row Interrupt Mask

Bit 2 – DISPIM Power-up/Power-down Sequence Terminated Interrupt Mask

Bit 1 – DISIM LCD Disable Interrupt Mask

Bit 0 – SOFIM Start of Frame Interrupt Mask

44.7.15. LCDC Interrupt Status Register

Name: LCDC_LCDISR
Offset: 0x00000038
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	WP							
Access	R							
Reset	0							

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					HEO	OVR2	OVR1	BASE
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
				FIFOERR	ROW	DISP	DIS	SOF
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 31 – WP Write Protection Interrupt Status

Value	Description
0	No Write Protection interrupt detected since last read of LCDC_LCDISR.
1	Indicates that a Write Protection error has been detected. This flag is reset after a read operation.

Bit 11 – HEO High-End Overlay Raw Interrupt Status

Value	Description
0	No High-End layer interrupt detected since last read of LCDC_HEOISR.
1	Indicates that a High-End layer interrupt is pending. This flag is reset as soon as the LCDC_HEOISR is read.

Bit 10 – OVR2 Overlay 2 Raw Interrupt Status

Value	Description
0	No Overlay 2 layer interrupt detected since last read of LCDC_OVR2ISR.
1	Indicates that an Overlay 2 layer interrupt is pending. This flag is reset as soon as the LCDC_OVR2ISR is read.

Bit 9 – OVR1 Overlay 1 Raw Interrupt Status

Value	Description
0	No Overlay 1 layer interrupt detected since last read of LCDC_OVR1ISR.
1	Indicates that an Overlay 1 layer interrupt is pending. This flag is reset as soon as the LCDC_OVR1ISR is read.

Bit 8 – BASE Base Layer Raw Interrupt Status

Value	Description
0	No base layer interrupt detected since last read of LCDC_BASEISR.

Value	Description
1	Indicates that a base layer interrupt is pending. This flag is reset as soon as the LCDC_BASEISR is read.

Bit 4 – FIFOERR Output FIFO Error

Value	Description
0	No underflow has occurred in the output FIFO since last read of LCDC_LCDISR.
1	Indicates that an underflow has occurred in the output FIFO. This flag is reset after a read operation.

Bit 3 – ROW Row Interrupt Status

Value	Description
0	No detection since last read of the LCDC_LCDCISR.
1	Indicates that a row event has been detected. This flag is reset after a read operation.

Bit 2 – DISP Power-up/Power-down Sequence Terminated Interrupt Status

Value	Description
0	Power-up sequence or power-down sequence has not yet terminated.
1	Indicates the power-up sequence or power-down sequence has terminated. This flag is reset after a read operation.

Bit 1 – DIS LCD Disable Interrupt Status

Value	Description
0	Horizontal and vertical timing generator has not yet been disabled.
1	Indicates that the horizontal and vertical timing generator has been disabled. This flag is reset after a read operation.

Bit 0 – SOF Start of Frame Interrupt Status

Value	Description
0	No detection since last read of LCDC_LCDISR.
1	Indicates that a start of frame event has been detected. This flag is reset after a read operation.

44.7.16. LCDC Attribute Enable Register

Name: LCDC_ATTRE
Offset: 0x0000003C
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
					HEOCL	OVR2CL	OVR1CL	BASECL
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
					HEO	OVR2	OVR1	BASE
Access					W	W	W	W
Reset					–	–	–	–

Bit 11 – HEOCL High-End Overlay Update Color LUT

Value	Description
0	No effect.
1	DMA update request for the HEO layer CLUT content.

Bit 10 – OVR2CL Overlay 2 Update Color LUT

Value	Description
0	No effect.
1	DMA update request for the OVR2 layer CLUT content.

Bit 9 – OVR1CL Overlay 1 Update Color LUT

Value	Description
0	No effect.
1	DMA update request for the OVR1 layer CLUT content.

Bit 8 – BASECL Base Layer Update Color LUT

Value	Description
0	No effect.
1	DMA update request for the base layer CLUT content.

Bit 3 – HEO High-End Overlay Update Attribute

Value	Description
0	No effect.

Value	Description
1	Updates the HEO window attribute.

Bit 2 – OVR2 Overlay 2 Update Attribute

Value	Description
0	No effect.
1	Updates the OVR2 window attribute.

Bit 1 – OVR1 Overlay 1 Update Attribute

Value	Description
0	No effect.
1	Updates the OVR1 window attribute.

Bit 0 – BASE Base Layer Update Attribute

Value	Description
0	No effect.
1	Updates the base window attributes.

44.7.17. LCDC Attribute Status Register

Name: LCDC_ATTRS
Offset: 0x00000040
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
					HEOCL	OVR2CL	OVR1CL	BASECL
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
					HEO	OVR2	OVR1	BASE
Access					R	R	R	R
Reset					0	0	0	0

Bit 31 – SIP Synchronization In Progress

Value	Description
0	Clock domain synchronization is terminated.
1	Synchronization is in progress. Access to LCDC_ATTRE has no effect.

Bit 11 – HEOCL High-End Overlay Color Table Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the High-end Overlay CLUT.

Bit 10 – OVR2CL Overlay 2 Color Table Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the Overlay 2 CLUT.

Bit 9 – OVR1CL Overlay 1 Color Table Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the Overlay 1 CLUT.

Bit 8 – BASECL Base Layer Color Table Status

Value	Description
0	No effect.

Value	Description
1	Indicates that an update request is pending for the base CLUT.

Bit 3 – HEO High-End Overlay Update Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for High-end Overlay.

Bit 2 – OVR2 Overlay 2 Update Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the Overlay 2 layer.

Bit 1 – OVR1 Overlay 1 Update Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the Overlay 1 layer.

Bit 0 – BASE Base Layer Update Status

Value	Description
0	No effect.
1	Indicates that an update request is pending for the base layer.

44.7.18. Base Layer Interrupt Enable Register

Name: LCDC_BASEIER
Offset: 0x00000060
Reset: –
Property: Write-only

This register can only be written if BWPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Enable

Bit 1 – ERROR Transfer Error Detected Interrupt Enable

Bit 0 – END End of Frame DMA Transfer Interrupt Enable

44.7.19. Base Layer Interrupt Disable Register

Name: LCDC_BASEIDR
Offset: 0x00000064
Reset: –
Property: Write-only

This register can only be written if BWPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Disable

Bit 1 – ERROR Transfer Error Interrupt Disable

Bit 0 – END End of Frame DMA Transfer Interrupt Disable

44.7.20. Base Layer Interrupt Mask Register

Name: LCDC_BASEIMR
Offset: 0x00000068
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt source is disabled.

1: The corresponding interrupt source 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
						OVF	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Interrupt Mask

Bit 1 – ERROR Transfer Error Interrupt Mask

Bit 0 – END End of Frame DMA Transfer Interrupt Mask

44.7.21. Base Layer Interrupt Status Register

Name: LCDC_BASEISR
Offset: 0x0000006C
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	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_BASEISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation.

Bit 1 – ERROR Transfer Error Detected

Value	Description
0	No system bus error has been detected since the last read of LCDC_BASEISR.
1	A system bus error has been detected. This flag is reset after a read operation.

Bit 0 – END End of Frame DMA Transfer

Value	Description
0	No end of frame DMA transfer has been detected since last read of LCDC_BASEISR.
1	End of transfer has been detected. This flag is reset after a read operation.

44.7.22. Base Layer Enable Register

Name: LCDC_BASEEN
Offset: 0x00000070
Reset: 0x00000000
Property: Read/Write

This register can only be written if BWPCRE is cleared in the LCDC 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
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Base Layer Enable

Value	Description
0	The layer is disabled.
1	The layer is enabled.

44.7.23. Base Layer Color Look-Up Table Address Register

Name: LCDC_BASECLA
Offset: 0x00000074
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CLA[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
	CLA[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
	CLA[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
	CLA[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 – CLA[29:0] Base Layer CLUT Address
The address register is 32-bit aligned.

44.7.24. Base Layer Frame Buffer Address Register

Name: LCDC_BASEFBA
Offset: 0x00000078
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	FBA[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
	FBA[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
	FBA[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
	FBA[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 – FBA[29:0] Frame Buffer Address
The address register is 32-bit aligned.

44.7.25. Base Layer Configuration Register 0

Name: LCDC_BASECFG0
Offset: 0x0000007C
Reset: 0x00000000
Property: Read/Write

This register can only be written if BWPCFGE is cleared in the LCDC 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
		BLEN[2:0]						
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bits 6:4 – BLEN[2:0] System Bus Burst Length

Value	Name	Description
0	INCR1	System bus access is started as soon as there is enough space in the FIFO to store one data.
1	INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used by default. INCR1 is used for bursts less than 4.
2	INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used by default. INCR4 bursts are used for bursts of 4 beats.INCR1 is used for bursts less than 4.
3	INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used by default. INCR8 and INCR4 bursts are respectively used for bursts of 8 and 4 beats.INCR1 is used for bursts less than 4.
4	INCR32	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 32 data. A system bus INCR32 Burst is used by default. INCR16, INCR8 and INCR4 bursts are respectively used for bursts of 16, 8 and 4 beats. INCR1 is used for bursts less than 4.
5–7	RESERVED	–

44.7.26. Base Layer Configuration Register 1

Name: LCDC_BASECFG1
Offset: 0x00000080
Reset: 0x00000000
Property: Read/Write

This register can only be written if BWPCFGE is cleared in the LCDC 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
							CLUTMODE[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:8 – CLUTMODE[1:0] CLUT Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	CLUT mode set to 1 bit per pixel
1	CLUT_2BPP	CLUT mode set to 2 bits per pixel
2	CLUT_4BPP	CLUT mode set to 4 bits per pixel
3	CLUT_8BPP	CLUT mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_ARGB_1555	16 bpp ARGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_ARGB_1666	19 bpp ARGB 1666
8	19BPP_ARGB_PACKED	19 bpp ARGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_ARGB_1888	25 bpp ARGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 2 – GAM Gamma Correction

When GAM = 1, writing in LCDC_BASECLUT0 to LCDC_BASECLUT255 has no effect.

Value	Description
0	Gamma correction is disabled
1	Gamma correction is activated

Bit 0 – CLUTEN CLUT Mode Enable

When CLUTEN = 1, writing in LCDC_BASECLUT0 to LCDC_BASECLUT255 has no effect.

Value	Description
0	RGB mode is selected.
1	CLUT mode is selected.

44.7.27. Base Layer Configuration Register 2

Name: LCDC_BASECFG2
Offset: 0x00000084
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[7:0]							
Access	R/W	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 – XSTRIDE[31:0] Horizontal Stride
Memory offset, in bytes, between two rows of the image memory.

44.7.28. Base Layer Configuration Register 3

Name: LCDC_BASECFG3
Offset: 0x00000088
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RDEF[7:0]							
Access	R/W	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
	GDEF[7:0]							
Access	R/W	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
	BDEF[7:0]							
Access	R/W	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 – RDEF[7:0] Red Default
Default red color when the base DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default green color when the base DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default blue color when the base DMA channel is disabled.

44.7.29. Base Layer Configuration Register 4

Name: LCDC_BASECFG4
Offset: 0x0000008C
Reset: 0x00000000
Property: Read/Write

This register can only be written if BWPCFGE is cleared in the LCDC 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
				DISCEN			REP	DMA
Access				R/W			R/W	R/W
Reset				0			0	0

Bit 4 – DISCEN Discard Area Enable

Value	Description
0	The whole frame is retrieved from memory.
1	The DMA channel discards the area located at screen coordinates {DISCXPOS, DISCYPOS}.

Bit 1 – REP Use Replication Logic to Expand RGB Color to 24 Bits
Alpha component is also affected by the replication logic.
In all ARGB formats with one transparency bit, REP affects the A field interpretation when A=1. If REP=0, then A=1 will be interpreted as Alpha = 0x80 (half transparent). If REP=1, then A=1 will be interpreted as Alpha = 0xFF (full opaque)

Value	Description
0	When the selected pixel depth is less than 24 bpp, the pixel is shifted and LSBs are set to 0.
1	When the selected pixel depth is less than 24 bpp, the pixel is shifted and the LSB replicates the MSB.

Bit 0 – DMA Use DMA Data Path

Value	Description
0	The default color is used on the base layer.
1	The DMA channel retrieves the pixel stream from the memory.

44.7.30. Base Layer Configuration Register 5

Name: LCDC_BASECFG5
Offset: 0x00000090
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
						DISCYPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DISCYPOS[7:0]							
Access	R/W	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
						DISCXPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	DISCXPOS[7:0]							
Access	R/W	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 – DISCYPOS[10:0] Discard Area Vertical Coordinate
Vertical position of the discard area.

Bits 10:0 – DISCXPOS[10:0] Discard Area Horizontal Coordinate
Horizontal position of the discard area.

44.7.31. Base Layer Configuration Register 6

Name: LCDC_BASECFG6
Offset: 0x00000094
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
						DISCYSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	DISCYSIZE[7:0]							
Access	R/W	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
						DISCXSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	DISCXSIZE[7:0]							
Access	R/W	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 – DISCYSIZE[10:0] Discard Area Vertical Size
Discard vertical size in pixels. The discard size is set to (DISCYSIZE + 1) pixels vertically.

Bits 10:0 – DISCXSIZE[10:0] Discard Area Horizontal Size
Discard horizontal size in pixels. The discard size is set to (DISCXSIZE + 1) pixels horizontally.

44.7.32. LCDC Write Protection Mode Register

Name: LCDC_WPMR
Offset: 0x000000E4
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	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	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		HEWPCRE	HEWPITE	HEWPCFGE	O2WPCRE	O2WPITE	O2WPCFGE	O1WPCRE
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
	O1WPITE	O1WPCFGE	BWPCRE	BWPITE	BWPCFGE	WPCRE	WPITE	WPCFGE
Access	R/W	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 – WPKEY[7:0] Write Protection Key Password

Value	Name	Description
0x58	PASSWD	Writing any other value in this field aborts the write operation of the WPCFGE, WPITE, WPCRE bits. Always reads as 0.

Bit 14 – HEWPCRE High-End Overlay Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of High-End Overlay Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of High-End Overlay Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 13 – HEWPITE High-End Overlay Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of High-End Overlay Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of High-End Overlay Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 12 – HEWPCFGE High-End Overlay Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of High-End Overlay Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of High-End Overlay Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 11 – O2WPCRE Overlay 2 Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of Overlay 2 Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 2 Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 10 – O2WPITE Overlay 2 Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of Overlay 2 Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 2 Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 9 – O2WPCFGE Overlay 2 Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of Overlay 2 Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 2 Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 8 – O1WPCRE Overlay 1 Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of Overlay 1 Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 1 Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 7 – O1WPITE Overlay 1 Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of Overlay 1 Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 1 Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 6 – O1WPCFGE Overlay 1 Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of Overlay 1 Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of Overlay 1 Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 5 – BWPCRE Base Layer Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of base layer Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of base layer Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 4 – BWPITE Base Layer Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of base layer Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of base layer Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 3 – BWPCFGE Base Layer Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of base layer Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of base layer Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 2 – WPCRE Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of LCDC Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of LCDC Control registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 1 – WPITE Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of LCDC Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of LCDC Interrupt Enable/Disable registers if WPKEY corresponds to 0x58 (“X” in ASCII).

Bit 0 – WPCFGE Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of LCDC Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).
1	Enables the write protection of LCDC Configuration registers if WPKEY corresponds to 0x58 (“X” in ASCII).

44.7.33. LCDC Write Protection Status Register

Name: LCDC_WPSR
Offset: 0x000000E8
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
						SEQE	CGD	WPVS
Access						R	R	R
Reset						0	0	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 2 – SEQE Internal Sequencer Error (cleared on read)

Value	Name
0	No peripheral internal sequencer error has occurred since the last read of LCDC_WPSR.
1	A peripheral internal sequencer error has occurred since the last read of LCDC_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 LCDC_WPSR. Under normal operating conditions, this bit is always cleared.
1	The clock monitoring circuitry has been corrupted since the last read of LCDC_WPSR. This flag can only be set in case of an abnormal clock signal waveform (glitch).

Bit 0 – WPVS Write Protection Violation Status

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

44.7.34. Overlay 1 Interrupt Enable Register

Name: LCDC_OVR1IER
Offset: 0x00000160
Reset: –
Property: Write-only

This register can only be written if O1WPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Enable

Bit 1 – ERROR Transfer Error Detected Interrupt Enable

Bit 0 – END End of Frame DMA Transfer Interrupt Enable

44.7.35. Overlay 1 Interrupt Disable Register

Name: LCDC_OVR1IDR
Offset: 0x00000164
Reset: –
Property: Write-only

This register can only be written if O1WPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Disable

Bit 1 – ERROR Error Interrupt Disable

Bit 0 – END End of Frame DMA Transfer Interrupt Disable

44.7.36. Overlay 1 Interrupt Mask Register

Name: LCDC_OVR1IMR
Offset: 0x00000168
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
						OVF	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Interrupt Mask

Bit 1 – ERROR Bus Error Detected Interrupt Mask

Bit 0 – END End of Frame DMA Transfer Interrupt Mask

44.7.37. Overlay 1 Interrupt Status Register

Name: LCDC_OVR1ISR
Offset: 0x0000016C
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	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_OVR1ISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation

Bit 1 – ERROR Bus Error Detected

Value	Description
0	No system bus error has been detected since the last read of LCDC_OVR1ISR.
1	A system bus error has been detected. This flag is reset after a read operation

Bit 0 – END End of Frame DMA Transfer

Value	Description
0	No end of transfer has been detected since last read of LCDC_OVR1ISR.
1	End of transfer has been detected. This flag is reset after a read operation.

44.7.38. Overlay 1 Enable Register

Name: LCDC_OVR1EN
Offset: 0x00000170
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCRE is cleared in the LCDC 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
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Layer Enable

Value	Description
0	The layer is disabled.
1	The layer is enabled.

44.7.39. Overlay 1 Color Look-Up Table Address Register

Name: LCDC_OVR1CLA
Offset: 0x00000174
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CLA[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
	CLA[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
	CLA[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
	CLA[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 – CLA[29:0] Overlay 1 CLUT Address
The address register is 32-bit aligned.

44.7.40. Overlay 1 Frame Buffer Address Register

Name: LCDC_OVR1FBA
Offset: 0x00000178
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FBA[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
	FBA[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
	FBA[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
	FBA[7:0]							
Access	R/W	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 – FBA[31:0] Frame Buffer Address

44.7.41. Overlay 1 Configuration Register 0

Name: LCDC_OVR1CFG0
Offset: 0x0000017C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC 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
		BLEN[2:0]						
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bits 6:4 – BLEN[2:0] System Bus Burst Length

Value	Name	Description
0	INCR1	System bus access is started as soon as there is enough space in the FIFO to store one data.
1	INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used by default. INCR1 is used for bursts less than 4.
2	INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used by default. INCR4 bursts are used for burst of 4 beats. INCR1 is used for bursts less than 4.
3	INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used by default. INCR8 and INCR4 bursts are respectively used for burst of 8 and 4 beats. INCR1 is used for bursts less than 4.
4	INCR32	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 32 data. A system bus INCR32 Burst is used by default. INCR16, INCR8 and INCR4 bursts are respectively used for burst of 16, 8 and 4 beats. INCR1 is used for bursts less than 4.
5-7	–	RESERVED

44.7.42. Overlay 1 Configuration Register 1

Name: LCDC_OVR1CFG1
Offset: 0x00000180
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC 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							R/W	R/W
Reset							0	0

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

Bits 9:8 – CLUTMODE[1:0] CLUT Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	CLUT mode set to 1 bit per pixel
1	CLUT_2BPP	CLUT mode set to 2 bits per pixel
2	CLUT_4BPP	CLUT mode set to 4 bits per pixel
3	CLUT_8BPP	CLUT mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_ARGB_1555	16 bpp ARGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_ARGB_1666	19 bpp ARGB 1666
8	19BPP_ARGB_PACKED	19 bpp ARGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_ARGB_1888	25 bpp ARGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 2 – GAM Gamma Correction

When GAM = 1, writing in LCDC_OVR1CLUT[0..255] has no effect.

Value	Description
0	Gamma correction is disabled
1	Gamma correction is enabled

Bit 0 – CLUTEN CLUT Mode Enable

When CLUTEN = 1, writing in LCDC_OVR1CLUT[0..255] has no effect.

Value	Description
0	RGB mode is selected.
1	CLUT mode is selected.

44.7.43. Overlay 1 Configuration Register 2

Name: LCDC_OVR1CFG2
Offset: 0x00000184
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	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
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	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 – YPOS[10:0] Vertical Window Position
Overlay 1 vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
Overlay 1 horizontal window position.

44.7.44. Overlay 1 Configuration Register 3

Name: LCDC_OVR1CFG3
Offset: 0x00000188
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	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
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	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 – YSIZE[10:0] Vertical Window Size
Overlay 1 window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
Overlay 1 window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

44.7.45. Overlay 1 Configuration Register 4

Name: LCDC_OVR1CFG4
Offset: 0x0000018C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[7:0]							
Access	R/W	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 – XSTRIDE[31:0] Horizontal Stride
Memory offset, in bytes, between two rows of the image memory.

44.7.46. Overlay 1 Configuration Register 5

Name: LCDC_OVR1CFG5
Offset: 0x00000190
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[7:0]							
Access	R/W	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 – PSTRIDE[31:0] Pixel Stride
Memory offset, in bytes, between two pixels of the image.

44.7.47. Overlay 1 Configuration Register 6

Name: LCDC_OVR1CFG6
Offset: 0x00000194
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADEF[7:0]							
Access	R/W	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
	RDEF[7:0]							
Access	R/W	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
	GDEF[7:0]							
Access	R/W	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
	BDEF[7:0]							
Access	R/W	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 – ADEF[7:0] Alpha Default
Default Alpha value when the Overlay 1 DMA channel is disabled (only for post-processing usage).

Bits 23:16 – RDEF[7:0] Red Default
Default red color when the Overlay 1 DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default green color when the Overlay 1 DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default blue color when the Overlay 1 DMA channel is disabled.

44.7.48. Overlay 1 Configuration Register 7

Name: LCDC_OVR1CFG7
Offset: 0x00000198
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Image components are MSB-aligned with their respective KEY field, so that if selected input mode involves components less than 8 bits, LSBs of KEY fields are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	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
	GKEY[7:0]							
Access	R/W	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
	BKEY[7:0]							
Access	R/W	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 – RKEY[7:0] Red Color Component Chroma Key
Reference red chroma key used to match the red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference green chroma key used to match the green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference blue chroma key used to match the blue color of the current overlay.

44.7.49. Overlay 1 Configuration Register 8

Name: LCDC_OVR1CFG8
Offset: 0x0000019C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Image components are MSB-aligned with their respective MASK field, so that if selected input mode involves components less than 8 bits, the LSBs of MASK fields are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	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
	GMASK[7:0]							
Access	R/W	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
	BMASK[7:0]							
Access	R/W	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 – RMASK[7:0] Red Color Component Chroma Key Mask
Red mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue mask used when the compare function is used. If a bit is set, then this bit is compared.

44.7.50. Overlay 1 Configuration Register 9

Name: LCDC_OVR1CFG9
Offset: 0x000001A0
Reset: 0x00000000
Property: Read/Write

This register can only be written if O1WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	A1[7:0]							
Access	R/W	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
	A0[7:0]							
Access	R/W	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
	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[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
	SFACTC[1:0]				DSTKEY	CRKEY	REP	DMA
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 31:24 – A1[7:0] Alpha 1 Value
Operand used in the Alpha blending process depending on factor value.

Bits 23:16 – A0[7:0] Alpha 0 Value
Operand used in the Alpha blending process depending on factor value.

Bits 15:14 – DFACTA[1:0] Destination Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	M_A0_MUL_AS	Computes A0 multiplied by source alpha, then sets the factor to one minus the result.
3	A1	Sets the factor to A1/255.

Bits 13:11 – DFACTC[2:0] Destination Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.
4	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
5	M_A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha, then sets the factor to one minus the result.
6	M_A0_MUL_AS	Computes A0 multiplied by source alpha0, then sets the factor to one minus the result.

Value	Name	Description
7	M_A0	Computes one minus A0, then sets the factor to one minus the result.

Bits 10:9 – SFACTA[1:0] Source Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.

Bits 8:6 – SFACTC[2:0] Source Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ONE	Sets the factor to 1.0.
1	ZERO	Sets the factor to 0.0.
2	A0	Sets the factor to A0/255.
3	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
4	A0_MUL_AS	Sets the factor to A0 multiplied by Source Alpha.
5	M_A0_MUL_AD	Computes A0 multiplied by Destination Alpha, then sets the factor to minus the result .

Bit 3 – DSTKEY Destination Color Keying

Value	Description
0	When CRKEY is enabled, color key is applied on OVR1 pixels, before the blending operation.
1	When CRKEY is enabled, color key is applied on OVR1 pixels, after the blending operation.

Bit 2 – CRKEY Chroma Keying Enable

Bit 1 – REP Replication Logic

The alpha component is also affected by the replication logic.
In all ARGB formats with one transparency bit, the REP configuration affects the A field interpretation when A=1. If REP=0, then A=1 is interpreted as Alpha = 0x80 (half transparent). If REP=1, then A=1 is interpreted as Alpha = 0xFF (full opaque).

Value	Description
0	When the selected pixel depth is less than 24 bpp, the pixel is shifted and LSBs are set to 0.
1	When the selected pixel depth is less than 24 bpp, the pixel is shifted and the LSB replicates the MSB.

Bit 0 – DMA DMA Enable

Value	Description
0	The pixel for the current layer is retrieved from the default color register.
1	The pixel stream is retrieved from the memory.

44.7.51. Overlay 2 Interrupt Enable Register

Name: LCDC_OVR2IER
Offset: 0x00000260
Reset: –
Property: Write-only

This register can only be written if O2WPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Enable

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Enable

Bit 0 – END End of Frame DMA Transfer Interrupt Enable

44.7.52. Overlay 2 Interrupt Disable Register

Name: LCDC_OVR2IDR
Offset: 0x00000264
Reset: –
Property: Write-only

This register can only be written if O2WPITE is cleared in the LCDC 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
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 2 – OVF Overflow Interrupt Disable

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Disable

Bit 0 – END End of Frame DMA Transfer Interrupt Disable

44.7.53. Overlay 2 Interrupt Mask Register

Name: LCDC_OVR2IMR
Offset: 0x00000268
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
						OVF	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Interrupt Mask

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Mask

Bit 0 – END End of Frame DMA Transfer Interrupt Mask

44.7.54. Overlay 2 Interrupt Status Register

Name: LCDC_OVR2ISR
Offset: 0x0000026C
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	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 2 – OVF Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_OVR2ISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation.

Bit 1 – ERROR Bus Error Detected

Value	Description
0	No system bus error has been detected since the last read of LCDC_OVR2ISR.
1	A system bus error has been detected. This flag is reset after a read operation.

Bit 0 – END End of Frame DMA Transfer

Value	Description
0	No end of transfer has been detected since last read of LCDC_OVR2ISR.
1	End of transfer has been detected. This flag is reset after a read operation.

44.7.55. Overlay 2 Enable Register

Name: LCDC_OVR2EN
Offset: 0x00000270
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCRE is cleared in the LCDC 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
Access								R/W
Reset								0

Bit 0 – ENABLE Layer Enable

Value	Description
0	The layer is disabled.
1	The layer is enabled.

44.7.56. Overlay 2 Color Look-Up Table Register

Name: LCDC_OVR2CLA
Offset: 0x00000274
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	CLA[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
	CLA[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
	CLA[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
	CLA[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 – CLA[29:0] Overlay 2 CLUT Address
The address register is 32-bit aligned.

44.7.57. Overlay 2 Frame Buffer Address Register

Name: LCDC_OVR2FBA
Offset: 0x00000278
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	FBA[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
	FBA[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
	FBA[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
	FBA[7:0]							
Access	R/W	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 – FBA[31:0] Frame Buffer Address

44.7.58. Overlay 2 Configuration Register 0

Name: LCDC_OVR2CFG0
Offset: 0x0000027C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC 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
		BLEN[2:0]						
Access		R/W	R/W	R/W				
Reset		0	0	0				

Bits 6:4 – BLEN[2:0] System Bus Burst Length

Value	Name	Description
0	INCR1	System bus access is started as soon as there is enough space in the FIFO to store one data.
1	INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used by default. INCR1 is used for bursts less than 4.
2	INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used by default. INCR4 bursts are used for bursts of 4 beats. INCR1 is used for bursts less than 4.
3	INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used by default. INCR8 and INCR4 bursts are respectively used for bursts of 8 and 4 beats. INCR1 is used for bursts less than 4.
4	INCR32	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 32 data. A system bus INCR32 Burst is used by default. INCR16, INCR8 and INCR4 bursts are respectively used for bursts of 16, 8 and 4 beats. INCR1 is used for bursts less than 4.
5-7	–	RESERVED

44.7.59. Overlay 2 Configuration Register 1

Name: LCDC_OVR2CFG1
Offset: 0x00000280
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC 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
							CLUTMODE[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:8 – CLUTMODE[1:0] CLUT Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	CLUT mode set to 1 bit per pixel
1	CLUT_2BPP	CLUT mode set to 2 bits per pixel
2	CLUT_4BPP	CLUT mode set to 4 bits per pixel
3	CLUT_8BPP	CLUT mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_ARGB_1555	16 bpp ARGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_ARGB_1666	19 bpp ARGB 1666
8	19BPP_ARGB_PACKED	19 bpp ARGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_ARGB_1888	25 bpp ARGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 2 – GAM Gamma Correction

When GAM = 1, writing in LCDC_OVR2CLUT[0..255] has no effect.

Value	Description
0	Gamma correction is disabled
1	Gamma correction is enabled

Bit 0 – CLUTEN CLUT Mode Enable

When CLUTEN = 1, writing in LCDC_OVR2CLUT[0..255] has no effect.

Value	Description
0	RGB mode is selected.
1	CLUT mode is selected.

44.7.60. Overlay 2 Configuration Register 2

Name: LCDC_OVR2CFG2
Offset: 0x00000284
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	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
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	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 – YPOS[10:0] Vertical Window Position
Overlay 2 vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
Overlay 2 horizontal window position.

44.7.61. Overlay 2 Configuration Register 3

Name: LCDC_OVR2CFG3
Offset: 0x00000288
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	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
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	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 – YSIZE[10:0] Vertical Window Size
Overlay 2 window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
Overlay 2 window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

44.7.62. Overlay 2 Configuration Register 4

Name: LCDC_OVR2CFG4
Offset: 0x0000028C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[7:0]							
Access	R/W	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 – XSTRIDE[31:0] Horizontal Stride
Memory offset, in bytes, between two rows of the image memory.

44.7.63. Overlay 2 Configuration Register 5

Name: LCDC_OVR2CFG5
Offset: 0x00000290
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[7:0]							
Access	R/W	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 – PSTRIDE[31:0] Pixel Stride
Memory offset, in bytes, between two pixels of the image memory.

44.7.64. Overlay 2 Configuration Register 6

Name: LCDC_OVR2CFG6
Offset: 0x00000294
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	ADEF[7:0]							
Access	R/W	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
	RDEF[7:0]							
Access	R/W	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
	GDEF[7:0]							
Access	R/W	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
	BDEF[7:0]							
Access	R/W	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 – ADEF[7:0] Alpha Default
Default Alpha value when the Overlay 2 DMA channel is disabled (only for post-processing usage).

Bits 23:16 – RDEF[7:0] Red Default
Default red color when the Overlay 2 DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default green color when the Overlay 2 DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default blue color when the Overlay 2 DMA channel is disabled.

44.7.65. Overlay 2 Configuration Register 7

Name: LCDC_OVR2CFG7
Offset: 0x00000298
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Image components are MSB-aligned with their respective KEY field, so that if selected input mode involves components less than 8 bits, KEY field LSBs are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	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
	GKEY[7:0]							
Access	R/W	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
	BKEY[7:0]							
Access	R/W	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 – RKEY[7:0] Red Color Component Chroma Key
Reference red chroma key used to match the red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference green chroma key used to match the green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference blue chroma key used to match the blue color of the current overlay.

44.7.66. Overlay 2 Configuration Register 8

Name: LCDC_OVR2CFG8
Offset: 0x0000029C
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Image components are MSB-aligned with their respective MASK field, so that if selected input mode involves components less than 8 bits, MASK field LSBs are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	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
	GMASK[7:0]							
Access	R/W	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
	BMASK[7:0]							
Access	R/W	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 – RMASK[7:0] Red Color Component Chroma Key Mask
Red mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue mask used when the compare function is used. If a bit is set, then this bit is compared.

44.7.67. Overlay 2 Configuration Register 9

Name: LCDC_OVR2CFG9
Offset: 0x000002A0
Reset: 0x00000000
Property: Read/Write

This register can only be written if O2WPCFGE is cleared in the LCDC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	A1[7:0]							
Access	R/W	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
	A0[7:0]							
Access	R/W	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
	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[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
	SFACTC[1:0]				DSTKEY	CRKEY	REP	DMA
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bits 31:24 – A1[7:0] Alpha 1 Value
Operand used in Alpha blending process depending on the factor value.

Bits 23:16 – A0[7:0] Alpha 0 Value
Operand used in Alpha blending process depending on the factor value.

Bits 15:14 – DFACTA[1:0] Destination Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	M_A0_MUL_AS	Computes A0 multiplied by source alpha, then sets the factor to one minus the result.
3	A1	Sets the factor to A1/255.

Bits 13:11 – DFACTC[2:0] Destination Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.
4	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
5	M_A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha, then set the factor one minus the result.
6	M_A0_MUL_AS	Computes A0 multiplied by source alpha0, then sets the factor to one minus the result.
7	M_A0	Computes one minus A0, then sets the factor to one minus the result.

Bits 10:9 – SFACTA[1:0] Source Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.

Bits 8:6 – SFACTC[2:0] Source Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ONE	Sets the factor to 1.0.
1	ZERO	Sets the factor to 0.0.
2	A0	Sets the factor to A0/255.
3	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
4	A0_MUL_AS	Sets the factor to A0 multiplied by Source Alpha.
5	M_A0_MUL_AD	Computes A0 multiplied by Destination Alpha, then sets the factor to minus the result.

Bit 3 – DSTKEY Destination Color Keying

Value	Description
0	When CRKEY is enabled, color key is applied on OVR2 pixels, before the blending operation.
1	When CRKEY is enabled, color key is applied on OVR2 pixels, after the blending operation.

Bit 2 – CRKEY Chroma Keying

Bit 1 – REP Replication Logic

Alpha component is also affected by the replication logic.
In all ARGB formats with 1 transparency bit, REP configuration affects the A field interpretation when A=1. If REP=0, then A=1 will be interpreted as a Alpha = 0x80 (half transparent). If REP=1, then A=1 will be interpreted as a Alpha = 0xFF (full opaque).

Value	Description
0	When the selected pixel depth is less than 24 bpp, the pixel is shifted and LSBs are set to 0.
1	When the selected pixel depth is less than 24 bpp, the pixel is shifted and the LSB replicates the MSB.

Bit 0 – DMA DMA Enable

Value	Description
0	The pixel for the current layer is retrieved from the default color register.
1	The pixel stream is retrieved from the memory.

44.7.68. High-End Overlay Interrupt Enable Register

Name: LCDC_HEOIER
Offset: 0x00000360
Reset: –
Property: Write-only

This register can only be written if HEWPITE is cleared in the LCDC 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
						CROVF	CRERROR	CREND
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
						CBOVF	CBERROR	CBEND
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 18 – CROVF Overflow for Cr Chroma Plane Interrupt Enable

Bit 17 – CRERROR Bus Transfer Error Detected for Cr Chroma Plane Interrupt Enable

Bit 16 – CREND End of Frame DMA for Cr Chroma Plane Transfer Interrupt Enable

Bit 10 – CBOVF Overflow for Cb or CbCr Chroma Plane Interrupt Enable

Bit 9 – CBERROR Bus Transfer Error Detected for Cb or CbCr Chroma Plane Interrupt Enable

Bit 8 – CBEND End of Frame DMA Transfer for Cb or CbCr Chroma Plane Interrupt Enable

Bit 2 – OVF Overflow Interrupt Enable

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Enable

Bit 0 – END End of Frame DMA Transfer Interrupt Enable

44.7.69. High-End Overlay Interrupt Disable Register

Name: LCDC_HEOIDR
Offset: 0x00000364
Reset: –
Property: Write-only

This register can only be written if HEWPITE is cleared in the LCDC 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
						CROVF	CRERROR	CREND
Access						W	W	W
Reset						–	–	–

Bit	15	14	13	12	11	10	9	8
						CBOVF	CBERROR	CBEND
Access						W	W	W
Reset						–	–	–

Bit	7	6	5	4	3	2	1	0
						OVF	ERROR	END
Access						W	W	W
Reset						–	–	–

Bit 18 – CROVF Overflow for Cr Chroma Plane Interrupt Disable

Bit 17 – CRERROR Bus Transfer Error Detected for Cr Chroma Plane Interrupt Disable

Bit 16 – CREND End of Frame DMA Transfer for Cr Chroma Plane Interrupt Disable

Bit 10 – CBOVF Overflow for Cb or CbCr Chroma Plane Interrupt Disable

Bit 9 – CBERROR Bus Transfer Error Detected for Cb or CbCr Chroma Plane Interrupt Disable

Bit 8 – CBEND End of Frame DMA Transfer for Cb or CbCr Chroma Plane Interrupt Disable

Bit 2 – OVF Overflow Interrupt Disable

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Disable

Bit 0 – END End of Frame DMA Transfer Interrupt Disable

44.7.70. High-End Overlay Interrupt Mask Register

Name: LCDC_HEOIMR
Offset: 0x00000368
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
						CROVF	CRERROR	CREND
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
						CBOVF	CBERROR	CBEND
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						OVF	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 18 – CROVF Overflow for Cr Chroma Plane Interrupt Mask

Bit 17 – CRERROR Bus Transfer Error Detected for Cr Chroma Plane Interrupt Mask

Bit 16 – CREND End of Frame DMA Transfer for Cr Chroma Plane Interrupt Mask

Bit 10 – CBOVF Overflow for Cb or CbCr Chroma Plane Interrupt Mask

Bit 9 – CBERROR Bus Transfer Error Detected for Cb or CbCr Chroma Plane Interrupt Mask

Bit 8 – CBEND End of Frame DMA Transfer for Cb or CbCr Chroma Plane Interrupt Mask

Bit 2 – OVF Overflow Interrupt Mask

Bit 1 – ERROR Bus Transfer Error Detected Interrupt Mask

Bit 0 – END End of Frame DMA Transfer Interrupt Mask

44.7.71. High-End Overlay Interrupt Status Register

Name: LCDC_HEOISR
Offset: 0x0000036C
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
						CROVF	CRERROR	CREND
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
						CBOVF	CBERROR	CBEND
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
						OVF	ERROR	END
Access						R	R	R
Reset						0	0	0

Bit 18 – CROVF Overflow Detected for Cr plane

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation.

Bit 17 – CRERROR Bus Transfer Error detected for Cr plane

Value	Description
0	No system bus error has been detected since the last read of LCDC_HEOISR.
1	A system bus error has been detected. This flag is reset after a read operation.

Bit 16 – CREND End of Frame DMA Transfer for Cr plane

Value	Description
0	No end of transfer has been detected since last read of LCDC_HEOISR.
1	End of transfer has been detected. This flag is reset after a read operation.

Bit 10 – CBOVF Overflow Detected for Cb or CbCr plane

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation.

Bit 9 – CBERROR Bus Transfer Error Detected for Cb or CbCr plane

Value	Description
0	No system bus error has been detected since the last read of LCDC_HEOISR.
1	A system bus error has been detected. This flag is reset after a read operation.

Bit 8 – CBEND End of Frame DMA Transfer for Cb or CbCr plane

Value	Description
0	No end of transfer has been detected since last read of LCDC_HEOISR.
1	End of transfer has been detected. This flag is reset after a read operation.

Bit 2 – OVF Overflow Detected

Value	Description
0	No overflow occurred since last read of LCDC_HEOISR.
1	An overflow occurred, at least one DMA transfer is still running at the End Of Frame. This flag is reset after a read operation

Bit 1 – ERROR Bus Error Detected

Value	Description
0	No system bus error has been detected since the last read of LCDC_HEOISR.
1	A system bus error has been detected. This flag is reset after a read operation.

Bit 0 – END End of Frame DMA Transfer

Value	Description
0	No end of transfer has been detected since last read of LCDC_HEOISR.
1	End of transfer has been detected. This flag is reset after a read operation.

44.7.72. High-End Overlay Enable Register

Name: LCDC_HEOEN
Offset: 0x00000370
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCRE is cleared in the LCDC 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
Access								R/W
Reset								0

Bit 0 – ENABLE Overlay Enable

Value	Description
0	The layer is disabled.
1	The layer is enabled.

44.7.73. High-End Overlay Color Look-Up Table Address Register

Name: LCDC_HEOCLA
Offset: 0x00000374
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CLA[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
	CLA[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
	CLA[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
	CLA[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 – CLA[29:0] High-End Overlay CLUT Address
The address register is 32-bit aligned.

44.7.74. High-End Overlay Frame Buffer Address Register

Name: LCDC_HEOYFBA0
Offset: 0x00000378
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	FBA[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
	FBA[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
	FBA[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
	FBA[7:0]							
Access	R/W	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 – FBA[31:0] Frame Buffer Address

44.7.75. High-End Overlay Cb Plane Frame Buffer Address Register

Name: LCDC_HEOCBFBA0
Offset: 0x0000037C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CBFBA[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
	CBFBA[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
	CBFBA[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
	CBFBA[7:0]							
Access	R/W	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 – CBFBA[31:0] Cb or CbCr Plane Frame Buffer Address

44.7.76. High-End Overlay Cr Plane Frame Buffer Address Register

Name: LCDC_HEOCRFBA0
Offset: 0x00000380
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CRFBA[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
	CRFBA[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
	CRFBA[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
	CRFBA[7:0]							
Access	R/W	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 – CRFBA[31:0] Cr Chroma Plane Frame Buffer Address

44.7.77. High-End Overlay Frame Buffer 1 Address Register

Name: LCDC_HEOYFBA1
Offset: 0x00000384
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	FBA[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
	FBA[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
	FBA[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
	FBA[7:0]							
Access	R/W	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 – FBA[31:0] Frame Buffer Address for Field 1 (used only for Interlaced frame content)

44.7.78. High-End Overlay Cb Plane Frame Buffer 1 Address Register

Name: LCDC_HEOCBFBA1
Offset: 0x00000388
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CBFBA[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
	CBFBA[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
	CBFBA[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
	CBFBA[7:0]							
Access	R/W	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 – CBFBA[31:0] Cb Plane Frame Buffer Address for Field 1 (used only for Interlaced frame content)

44.7.79. High-End Overlay Cr Plane Frame Buffer 1 Address Register

Name: LCDC_HEOCRFBA1
Offset: 0x0000038C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CRFBA[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
	CRFBA[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
	CRFBA[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
	CRFBA[7:0]							
Access	R/W	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 – CRFBA[31:0] Cr Plane Frame Buffer Address for Field 1 (used only for Interlaced frame content)

44.7.80. High-End Overlay Configuration Register 0

Name: LCDC_HEOCFG0
Offset: 0x00000390
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
						BLENCC[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

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

Bits 10:8 – BLENCC[2:0] System Bus Burst Length for Cb-Cr Channel

Value	Name	Description
0	INCR1	System bus access is started as soon as there is enough space in the FIFO to store one data.
1	INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used by default. INCR1 is used for bursts less than 4.
2	INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used by default. INCR4 bursts are used for bursts of 4 beats. INCR1 is used for bursts less than 4.
3	INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used by default. INCR8 and INCR4 bursts are respectively used for bursts of 8 and 4 beats. INCR1 is used for bursts less than 4.
4	INCR32	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 32 data. A system bus INCR32 Burst is used by default. INCR16, INCR8 and INCR4 bursts are respectively used for bursts of 16, 8 and 4 beats. INCR1 is used for bursts less than 4.

Bits 6:4 – BLEN[2:0] System Bus Burst Length
This field is applicable only for YCrCb, planar or semi-planar modes.

Value	Name	Description
0	INCR1	System bus access is started as soon as there is enough space in the FIFO to store one data.
1	INCR4	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 4 data. A system bus INCR4 Burst is used by default. INCR1 is used for bursts less than 4.
2	INCR8	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 8 data. A system bus INCR8 Burst is used by default. INCR4 bursts are used for bursts of 4 beats. INCR1 is used for bursts less than 4.

Value	Name	Description
3	INCR16	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 16 data. A system bus INCR16 Burst is used by default. INCR8 and INCR4 bursts are respectively used for bursts of 8 and 4 beats. INCR1 is used for bursts less than 4.
4	INCR32	System bus access is started as soon as there is enough space in the FIFO to store a total amount of 32 data. A system bus INCR32 Burst is used by default. INCR16, INCR8 and INCR4 bursts are respectively used for bursts of 16, 8 and 4 beats. INCR1 is used for bursts less than 4.

44.7.81. High-End Overlay Configuration Register 1

Name: LCDC_HEOCFG1
Offset: 0x00000394
Reset: 0x00000000
Property: Read/Write

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

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

Bit	23	22	21	20	19	18	17	16
								YCC422ROT
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	YCCMODE[3:0]						CLUTMODE[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
	RGBMODE[3:0]					GAM	YCCEN	CLUTEN
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 – ILD Interlaced Content

Value	Description
0	Progressive scan is used.
1	Interlaced Frame content is used and deinterlacing process is activated.

Bit 16 – YCC422ROT YCbCr 4:2:2 Rotation

Value	Description
0	Chroma upsampling kernel is configured to use 0 and 180 degrees algorithm.
1	Indicates that the chroma upsampling kernel is configured to use the 4:2:2 rotation algorithm. This bit is relevant only when a rotation angle of 90 degrees or 270 degrees is used.

Bits 15:12 – YCCMODE[3:0] YCbCr Mode Input Selection

Value	Name	Description
0	32BPP_AYCBCR	32 bpp AYCbCr 444
1	16BPP_YCBCR_MODE0	16 bpp Cr(n)Y(n+1)Cb(n)Y(n) 4:2:2
2	16BPP_YCBCR_MODE1	16 bpp Y(n+1)Cr(n)Y(n)Cb(n) 4:2:2
3	16BPP_YCBCR_MODE2	16 bpp Cb(n)Y(+1)Cr(n)Y(n) 4:2:2
4	16BPP_YCBCR_MODE3	16 bpp Y(n+1)Cb(n)Y(n)Cr(n) 4:2:2
5	16BPP_YCBCR_SEMIPLANAR	16 bpp Semiplanar 4:2:2 YCbCr
6	16BPP_YCBCR_PLANAR	16 bpp Planar 4:2:2 YCbCr
7	12BPP_YCBCR_SEMIPLANAR	12 bpp Semiplanar 4:2:0 YCbCr
8	12BPP_YCBCR_PLANAR	12 bpp Planar 4:2:0 YCbCr

Bits 9:8 – CLUTMODE[1:0] CLUT Mode Input Selection

Value	Name	Description
0	CLUT_1BPP	CLUT mode set to 1 bit per pixel
1	CLUT_2BPP	CLUT mode set to 2 bits per pixel
2	CLUT_4BPP	CLUT mode set to 4 bits per pixel
3	CLUT_8BPP	CLUT mode set to 8 bits per pixel

Bits 7:4 – RGBMODE[3:0] RGB Mode Input Selection

Value	Name	Description
0	12BPP_RGB_444	12 bpp RGB 444
1	16BPP_ARGB_4444	16 bpp ARGB 4444
2	16BPP_RGBA_4444	16 bpp RGBA 4444
3	16BPP_RGB_565	16 bpp RGB 565
4	16BPP_ARGB_1555	16 bpp ARGB 1555
5	18BPP_RGB_666	18 bpp RGB 666
6	18BPP_RGB_666PACKED	18 bpp RGB 666 PACKED
7	19BPP_ARGB_1666	19 bpp ARGB 1666
8	19BPP_ARGB_PACKED	19 bpp ARGB 1666 PACKED
9	24BPP_RGB_888	24 bpp RGB 888
10	24BPP_RGB_888_PACKED	24 bpp RGB 888 PACKED
11	25BPP_ARGB_1888	25 bpp ARGB 1888
12	32BPP_ARGB_8888	32 bpp ARGB 8888
13	32BPP_RGBA_8888	32 bpp RGBA 8888

Bit 2 – GAM Gamma Correction

When GAM = 1, writing in LCDC_HEOCLUT[0..255] has no effect.

Value	Description
0	Gamma correction is disabled.
1	Gamma correction is enabled.

Bit 1 – YCCEN YCbCr Color Space Enable

Value	Description
0	Color space is RGB.
1	Color space is YCbCr.

Bit 0 – CLUTEN CLUT Mode Enable

When CLUTEN = 1, writing in LCDC_HEOCLUT[0..255] has no effect.

Value	Description
0	RGB mode is selected.
1	CLUT mode is selected.

44.7.82. High-End Overlay Configuration Register 2

Name: LCDC_HEOCFG2
Offset: 0x00000398
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
						YPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YPOS[7:0]							
Access	R/W	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
						XPOS[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XPOS[7:0]							
Access	R/W	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 – YPOS[10:0] Vertical Window Position
High-end overlay vertical window position.

Bits 10:0 – XPOS[10:0] Horizontal Window Position
High-end overlay horizontal window position.

44.7.83. High-End Overlay Configuration Register 3

Name: LCDC_HEOCFG3
Offset: 0x0000039C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
						YSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YSIZE[7:0]							
Access	R/W	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
						XSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XSIZE[7:0]							
Access	R/W	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 – YSIZE[10:0] Vertical Window Size
High-end overlay window height in pixels. The window height is set to (YSIZE + 1).
The following constraint must be met: YPOS + YSIZE ≤ RPF

Bits 10:0 – XSIZE[10:0] Horizontal Window Size
High-end overlay window width in pixels. The window width is set to (XSIZE + 1).
The following constraint must be met: XPOS + XSIZE ≤ PPL

44.7.84. High-End Overlay Configuration Register 4

Name: LCDC_HEOCFG4
Offset: 0x000003A0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
						YMEMSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	23	22	21	20	19	18	17	16
	YMEMSIZE[7:0]							
Access	R/W	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
						XMEMSIZE[10:8]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	XMEMSIZE[7:0]							
Access	R/W	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 – YMEMSIZE[10:0] Vertical image Size in Memory
High-end overlay image height in pixels. The image height is set to (YMEMSIZE + 1).

Bits 10:0 – XMEMSIZE[10:0] Horizontal image Size in Memory
High-end overlay image width in pixels. The image width is set to (XMEMSIZE + 1).

44.7.85. High-End Overlay Configuration Register 5

Name: LCDC_HEOCFG5
Offset: 0x000003A4
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[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
	XSTRIDE[7:0]							
Access	R/W	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 – XSTRIDE[31:0] Horizontal Stride
Memory offset, in bytes, between two rows of the image memory.

44.7.86. High-End Overlay Configuration Register 6

Name: LCDC_HEOCFG6
Offset: 0x000003A8
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[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
	PSTRIDE[7:0]							
Access	R/W	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 – PSTRIDE[31:0] Pixel Stride
Memory offset, in bytes, between two pixels of the image memory.

44.7.87. High-End Overlay Configuration Register 7

Name: LCDC_HEOCFG7
Offset: 0x000003AC
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CCXSTRIDE[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
	CCXSTRIDE[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
	CCXSTRIDE[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
	CCXSTRIDE[7:0]							
Access	R/W	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 – CCXSTRIDE[31:0] CbCr Horizontal Stride
Memory offset, in bytes, between two rows of the image memory.

44.7.88. High-End Overlay Configuration Register 8

Name: LCDC_HEOCFG8
Offset: 0x000003B0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	CCPSTRIDE[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
	CCPSTRIDE[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
	CCPSTRIDE[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
	CCPSTRIDE[7:0]							
Access	R/W	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 – CCPSTRIDE[31:0] CbCr Pixel Stride
Memory offset, in bytes, between two pixels of the image memory.

44.7.89. High-End Overlay Configuration Register 9

Name: LCDC_HEOCFG9
Offset: 0x000003B4
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	ADEF[7:0]							
Access	R/W	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
	RDEF[7:0]							
Access	R/W	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
	GDEF[7:0]							
Access	R/W	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
	BDEF[7:0]							
Access	R/W	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 – ADEF[7:0] Alpha Default
Default Alpha value when the high-end overlay DMA channel is disabled.

Bits 23:16 – RDEF[7:0] Red Default
Default red color when the high-end overlay DMA channel is disabled.

Bits 15:8 – GDEF[7:0] Green Default
Default green color when the high-end overlay DMA channel is disabled.

Bits 7:0 – BDEF[7:0] Blue Default
Default blue color when the high-end overlay DMA channel is disabled.

44.7.90. High-End Overlay Configuration Register 10

Name: LCDC_HEOCFG10
Offset: 0x000003B8
Reset: 0x00000000
Property: Read/Write

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

Image components are MSB-aligned with their respective KEY field, so that if selected input mode involves components less than 8 bits, KEY field LSBs are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RKEY[7:0]							
Access	R/W	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
	GKEY[7:0]							
Access	R/W	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
	BKEY[7:0]							
Access	R/W	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 – RKEY[7:0] Red Color Component Chroma Key
Reference red chroma key used to match the red color of the current overlay.

Bits 15:8 – GKEY[7:0] Green Color Component Chroma Key
Reference green chroma key used to match the green color of the current overlay.

Bits 7:0 – BKEY[7:0] Blue Color Component Chroma Key
Reference blue chroma key used to match the blue color of the current overlay.

44.7.91. High-End Overlay Configuration Register 11

Name: LCDC_HEOCFG11
Offset: 0x000003BC
Reset: 0x00000000
Property: Read/Write

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

Image components are MSB-aligned with their respective MASK field, so that if selected input mode involves components less than 8 bits, MASK field LSBs are ignored.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RMASK[7:0]							
Access	R/W	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
	GMASK[7:0]							
Access	R/W	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
	BMASK[7:0]							
Access	R/W	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 – RMASK[7:0] Red Color Component Chroma Key Mask
Red mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 15:8 – GMASK[7:0] Green Color Component Chroma Key Mask
Green mask used when the compare function is used. If a bit is set, then this bit is compared.

Bits 7:0 – BMASK[7:0] Blue Color Component Chroma Key Mask
Blue mask used when the compare function is used. If a bit is set, then this bit is compared.

44.7.92. High-End Overlay Configuration Register 12

Name: LCDC_HEOCFG12
Offset: 0x000003C0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	A1[7:0]							
Access	R/W	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
	A0[7:0]							
Access	R/W	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
	DFACTA[1:0]		DFACTC[2:0]			SFACTA[1:0]		SFACTC[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
	SFACTC[1:0]		VIDPRI		DSTKEY	CRKEY	REP	DMA
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 – A1[7:0] Alpha 1 Value
Operand used in Alpha blending process depending on factor value.

Bits 23:16 – A0[7:0] Alpha 0 Value
Operand used in Alpha blending process depending on factor value.

Bits 15:14 – DFACTA[1:0] Destination Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	M_A0_MUL_AS	Computes A0 multiplied by source alpha, then set the factor to one minus the result.
3	A1	Sets the factor to A1/255.

Bits 13:11 – DFACTC[2:0] Destination Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.
4	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
5	M_A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha, then sets the factor one minus the result.
6	M_A0_MUL_AS	Computes A0 multiplied by source alpha0, then set the factor to one minus the result.
7	M_A0	Computes one minus A0, then set the factor to one minus the result.

Bits 10:9 – SFACTA[1:0] Source Factor for the Blending Equation of the Alpha Component

Value	Name	Description
0	ZERO	Sets the factor to 0.0.
1	ONE	Sets the factor to 1.0.
2	A0	Sets the factor to A0/255.
3	A1	Sets the factor to A1/255.

Bits 8:6 – SFACTC[2:0] Source Factor for the Blending Equation of the Color Component

Value	Name	Description
0	ONE	Sets the factor to 1.0.
1	ZERO	Sets the factor to 0.0.
2	A0	Sets the factor to A0/255.
3	A0_MULT_AD	Sets the factor to A0 multiplied by Destination Alpha.
4	A0_MUL_AS	Sets the factor to A0 multiplied by Source Alpha.
5	M_A0_MUL_AD	Computes A0 multiplied by Destination Alpha, then sets the factor to minus the result.

Bit 5 – VIDPRI Video Priority Mode

Value	Description
0	HEO is located below OVR1.
1	HEO is located above OVR1.

Bit 3 – DSTKEY Destination Color Keying

Value	Description
0	When CRKEY is enabled, color key is applied on HEO pixels, before blending operation.
1	When CRKEY is enabled, color key is applied on HEO pixels, after blending operation.

Bit 2 – CRKEY Chroma Keying

Bit 1 – REP Replication Logic

Alpha component is also affected by the replication logic.
In all ARGB formats with 1 transparency bit, the REP configuration affects the A field interpretation when A=1. If REP=0, then A=1 will be interpreted as Alpha = 0x80 (half transparent). If REP=1, then A=1 will be interpreted as Alpha = 0xFF (full opaque)

Value	Description
0	When the selected pixel depth is less than 24 bpp, the pixel is shifted and LSBs are set to 0.
1	When the selected pixel depth is less than 24 bpp, the pixel is shifted and the LSB replicates the MSB.

Bit 0 – DMA DMA Enable

Value	Description
0	The pixel for the current layer is retrieved from the default color register.
1	The pixel stream is retrieved from the memory.

44.7.93. High-End Overlay Configuration Register 13

Name: LCDC_HEOCFG13
Offset: 0x000003C4
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
				BL				ENABLE
Access				R/W				R/W
Reset				0				0

Bit 4 – BL Black Level

Value	Description
0	Black level offset is disabled.
1	Black level offset is enabled.

Bit 0 – ENABLE Contrast, Brightness Hue and Saturation Enable

Value	Description
0	CBHS is disabled.
1	CBHS is enabled.

44.7.94. High-End Overlay Configuration Register 14

Name: LCDC_HEOCFG14
Offset: 0x000003C8
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
					CONT[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
	CONT[7:0]							
Access	R/W	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
						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 27:16 – CONT[11:0] Contrast Value
The contrast value mapping is 0:8:4 (unsigned fixed point with 8 bits for integer part and 4 bits for fractional part).

Bits 10:0 – BRIGHT[10:0] Brightness Value
The brightness value mapping is 1:10:0 (signed 2's complement fixed point with 1 sign bit and 10 bits for integer part).

44.7.95. High-End Overlay Configuration Register 15

Name: LCDC_HEOCFG15
Offset: 0x000003CC
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
					SAT[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
	SAT[7:0]							
Access	R/W	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
								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 27:16 – SAT[11:0] Saturation Value
The saturation value mapping is 0:8:4 (unsigned fixed point with 8 bits for integer part and 4 bits for fractional part).

Bits 8:0 – HUE[8:0] Hue Value in Degrees
The hue is an unsigned integer value expressed in degrees (from 0 to 360).

44.7.96. High-End Overlay Configuration Register 16

Name: LCDC_HEOCFG16
Offset: 0x000003D0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
				RCBGAIN[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
	RCBGAIN[7:0]							
Access	R/W	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
				RYGAIN[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
	RYGAIN[7:0]							
Access	R/W	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 – RCBGAIN[12:0] Color Space Conversion CB Coefficient for Red Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

Bits 12:0 – RYGAIN[12:0] Color Space Conversion Y Coefficient for Red Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.97. High-End Overlay Configuration Register 17

Name: LCDC_HEOCFG17
Offset: 0x000003D4
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
				RCRGAIN[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
	RCRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – RCRGAIN[12:0] Color Space Conversion CR Coefficient for Red Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.98. High-End Overlay Configuration Register 18

Name: LCDC_HEOCFG18
Offset: 0x000003D8
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
				GCBGAIN[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
	GCBGAIN[7:0]							
Access	R/W	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
				GYGAIN[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
	GYGAIN[7:0]							
Access	R/W	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 – GCBGAIN[12:0] Color Space Conversion CB Coefficient for Green Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

Bits 12:0 – GYGAIN[12:0] Color Space Conversion Y Coefficient for Green Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.99. High-End Overlay Configuration Register 19

Name: LCDC_HEOCFG19
Offset: 0x000003DC
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
				GCRGAIN[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
	GCRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – GCRGAIN[12:0] Color Space Conversion CR Coefficient for Green Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.100.High-End Overlay Configuration Register 20

Name: LCDC_HEOCFG20
Offset: 0x000003E0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
				BCBGAIN[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
	BCBGAIN[7:0]							
Access	R/W	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
				BYGAIN[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
	BYGAIN[7:0]							
Access	R/W	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 – BCBGAIN[12:0] Color Space Conversion CB Coefficient for Blue Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

Bits 12:0 – BYGAIN[12:0] Color Space Conversion Y Coefficient for Blue Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.101.High-End Overlay Configuration Register 21

Name: LCDC_HEOCFG21
Offset: 0x000003E4
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
				BCRGAIN[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
	BCRGAIN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – BCRGAIN[12:0] Color Space Conversion Y Coefficient for Blue Component 1:2:10 Format
Color space conversion coefficient format is 1 sign bit, 2 bits for integer part and 10 bits for fractional part.

44.7.102.High-End Overlay Configuration Register 22

Name: LCDC_HEOCFG22
Offset: 0x000003E8
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
						CROFF	CBOFF	YOFF
Access						R/W	R/W	R/W
Reset						0	0	0

Bit 2 – CROFF Red Chroma Offset

Bit 1 – CBOFF Blue Chroma Offset

Bit 0 – YOFF Luma Offset

44.7.103.High-End Overlay Configuration Register 23

Name: LCDC_HEOCFG23
Offset: 0x000003EC
Reset: 0x00000000
Property: Read/Write

This register can only be written if HEWPCFGE is cleared in the LCDC 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
			HXSCEN	HXSYEN			VXSCEN	VXSYEN
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 5 – HXSCEN Horizontal Scaler Chroma Component Enable

Bit 4 – HXSYEN Horizontal Scaler Luma Component Enable

Bit 1 – VXSCEN Vertical Scaler Chroma Component Enable

Bit 0 – VXSYEN Vertical Scaler Luma Component Enable

44.7.104.High-End Overlay Configuration Register 24

Name: LCDC_HEOCFG24
Offset: 0x000003F0
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	VXSYFACT[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
	VXSYFACT[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
	VXSYFACT[7:0]							
Access	R/W	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 – VXSYFACT[23:0] Vertical Luma Scaling Factor

44.7.105.High-End Overlay Configuration Register 25

Name: LCDC_HEOCFG25
Offset: 0x000003F4
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	VXSCFACT[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
	VXSCFACT[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
	VXSCFACT[7:0]							
Access	R/W	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 – VXSCFACT[23:0] Vertical Chroma Scaling Factor

44.7.106.High-End Overlay Configuration Register 26

Name: LCDC_HEOCFG26
Offset: 0x000003F8
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HXSYFACT[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
	HXSYFACT[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
	HXSYFACT[7:0]							
Access	R/W	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 – HXSYFACT[23:0] Horizontal Luma Scaling factor

44.7.107.High-End Overlay Configuration Register 27

Name: LCDC_HEOCFG27
Offset: 0x000003FC
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HXSCFACT[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
	HXSCFACT[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
	HXSCFACT[7:0]							
Access	R/W	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 – HXSCFACT[23:0] Horizontal Chroma Scaling Factor

44.7.108.High-End Overlay Configuration Register 28

Name: LCDC_HEOCFG28
Offset: 0x00000400
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
					VXSCOFF1[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
					VXSCOFF[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
					VXSYOFF1[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
					VXSYOFF[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 27:24 – VXSCOFF1[3:0] Vertical Scaler Chroma Default Phase for Field 1

Bits 19:16 – VXSCOFF[3:0] Vertical Scaler Chroma Default Phase for Field 0 or Progressive Scan

Bits 11:8 – VXSYOFF1[3:0] Vertical Scaler Luma Default Phase for Field 1

Bits 3:0 – VXSYOFF[3:0] Vertical Scaler Luma Default Phase for Field 0 or Progressive Scan

44.7.109.High-End Overlay Configuration Register 29

Name: LCDC_HEOCFG29
Offset: 0x00000404
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bits 19:16 – HXSCOFF[3:0] Hozontal Scaler Chroma Offset

Bits 3:0 – HXSYOFF[3:0] Horizontal Scaler Luma Offset

44.7.110.High-End Overlay Configuration Register 30

Name: LCDC_HEOCFG30
Offset: 0x00000408
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
							VXSCCFG1[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
		VXSCBICU	VXSC1201N	VXSCTAP2			VXSCCFG[1:0]	
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
							VXSYCFG1[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
		VXSYBICU	VXSY1201N	VXSYTAP2			VXSYCFG[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 25:24 – VXSCCFG1[1:0] Vertical Scaler Chroma Configuration for Field 1(only used for YCrCb Interlaced Frame Content)

Filter taps shift for chroma components. The value is from 0 to 3 when VXSYTAP2 = 0 (4 coefficients used) and from 0 to 1 when when VXSYTAP2 = 1 (2 coefficients used).

Note: When using packed formats, VXSCCFG1 and VXSYCFG1 must have the same value.

Bit 22 – VXSCBICU Vertical Scaler Predefined Bicubic Taps Selected for Chroma/GB Component

Value	Description
0	Bicubic taps values are not selected.
1	Bicubic taps values are selected.

Bit 21 – VXSC1201N Vertical Scaler Bilinear 2 Taps Position among 4 Taps Filter for Chroma/GB Component

Value	Description
0	When selected, bilinear taps values are located on 0 and 1 positions (left aligned).
1	When selected, bilinear taps values are located on 0 and 1 positions (center aligned).

Bit 20 – VXSCTAP2 Vertical Scaler Predefined Bilinear 2 Taps Selected for Chroma/GB Component

Value	Description
0	Bilinear 2 taps value not selected.
1	Bilinear 2 taps value selected.

Bits 17:16 – VXSCCFG[1:0] Vertical Scaler Chroma/GB Configuration for Field 0 or Progressive Scan

Filter taps shift for chroma components. The value is from 0 to 3 when VXSYTAP2 = 0 (4 coefficients used) and from 0 to 1 when when VXSYTAP2 = 1 (2 coefficients used).

Note: When using packed formats, VXSCCFG and VXSYCFG must have the same value.

Bits 9:8 – VXSYCFG1[1:0] Vertical Scaler Chroma/GB Configuration for Field 1 (only used for interlaced frame content)

Filter taps shift for chroma components. The value is from 0 to 3 when VXSYTAP2 = 0 (4 coefficients used) and from 0 to 1 when when VXSYTAP2 = 1 (2 coefficients used).

Note: When using packed formats, VXSYCFG1 and VXSCCFG1 must have the same value.

Bit 6 – VXSYBICU Vertical Scaler Predefined Bicubic Taps Selected for Luma/AR Component

Value	Description
0	Bicubic taps values are not selected.
1	Bicubic taps values are selected.

Bit 5 – VXSY1201N Vertical Scaler Bilinear 2 Taps Position among 4 Filter Taps for Luma/AR Component

Value	Description
0	When selected, bilinear tap values are located on 0 and 1 positions (left-aligned).
1	When selected, bilinear tap values are located on 0 and 1 positions (center-aligned).

Bit 4 – VXSYTAP2 Vertical Scaler Predefined Bilinear 2 Taps Selected for Luma/AR Component

Value	Description
0	Bilinear 2 taps value not selected.
1	Bilinear 2 taps value selected.

Bits 1:0 – VXSYCFG[1:0] Vertical Scaler Luma/AR Configuration for Field 0 or Progressive Scan

Filter taps shift for luma component. The value is from 0 to 3 when VXSYTAP2=0 (4 coefficients used) and from 0 to 1 when VXSYTAP2=1 (2 coefficients used).

Note: When using packed formats, VXSYCFG and VXSCCFG must have the same value.

44.7.111.High-End Overlay Configuration Register 31

Name: LCDC_HEOCFG31
Offset: 0x0000040C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			HXSCBICU	HXSCTAP2			HXSCCFG[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
			HXSYBICU	HXSYTAP2			HXSYCFG[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bit 21 – HXSCBICU Horizontal Scaler Bicubic Coefficients Selected for Chroma Component

Value	Description
0	Bicubic taps values are not selected.
1	Bicubic taps values are selected.

Bit 20 – HXSCTAP2 Horizontal Scaler Predefined Bilinear 2 Taps Selected for Chroma Component

Value	Description
0	Bilinear 2 taps value not selected.
1	Bilinear 2 taps value selected.

Bits 17:16 – HXSCCFG[1:0] Horizontal Scaler Chroma/GB Configuration
Filter taps shift for chroma components. The value is from 0 to 3 when HXSYTAP2 = 0 (4 coefficients used) and from 0 to 1 when when HXSYTAP2 = 1 (2 coefficients used).
When using packed formats, HXSCCFG and HXSYCFG must have the same value.

Bit 5 – HXSYBICU Horizontal Scaler Predefined Bicubic Taps Selected for Luma/AR Component

Value	Description
0	Bicubic taps values are not selected.
1	Bicubic taps values are selected.

Bit 4 – HXSYTAP2 Horizontal Scaler Predefined Bilinear 2 Taps Selected for Luma/AR Component

Value	Description
0	Bilinear 2 taps value not selected.

Value	Description
1	Bilinear 2 taps value selected.

Bits 1:0 – HXSYCFG[1:0] Horizontal Scaler Luma Configuration
Filter taps shift for luma component. The value is from 0 to 3 when HXSYTAP2 = 0 (4 coefficients used) and from 0 to 1 when when HXSYTAP2 = 1 (2 coefficients used).
When using packed formats, HXSCCFG and HXSYCFG must have the same value.

44.7.112.High-End Overlay Vertical Scaler Taps 1 and 0 Phase x Register

Name: LCDC_HEOVTAP10Px
Offset: 0x0410 + 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 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

Bits 12:0 – TAP0[12:0] Vertical Filter Tap 0 Coefficient
Filter coefficient for TAP 0, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

44.7.113.High-End Overlay Vertical Scaler Taps 3 and 2 Phase x Register

Name: LCDC_HEOVTAP32Px
Offset: 0x0414 + 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 1, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

Bits 12:0 – TAP2[12:0] Vertical Filter Tap 2 Coefficient
Filter coefficient for TAP 0, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

44.7.114.High-End Overlay Horizontal Scaler Taps 1 and 0 Phase x Register

Name: LCDC_HEOHTAP10Px
Offset: 0x0490 + 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] Horitzontal Filter Tap 1 Coefficient
Filter coefficient for TAP 1, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

Bits 12:0 – TAP0[12:0] Horizontal Filter Tap 0 Coefficient
Filter coefficient for TAP 0, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

44.7.115.High-End Overlay Horizontal Scaler Taps 3 and 2 Phase x Register

Name: LCDC_HEOHTAP32Px
Offset: 0x0494 + 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] Horizontal Filter Tap 3 Coefficient
Filter coefficient for TAP 3, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

Bits 12:0 – TAP2[12:0] Horizontal Filter Tap 2 Coefficient
Filter coefficient for TAP 2, the format is 1:2:10 (signed 2's complement fixed point with 1 sign bit, 2 bits for integer part and 10 bits for fractional part).

44.7.116.Base Layer CLUT Register x

Name: LCDC_BASECLUTx
Offset: 0x0700 + x*0x04 [x=0..255]
Reset: 0x–
Property: Read/Write

Notes:

1. The reset value is undefined because the CLUT registers are located in the embedded RAM.
2. BASECFG1.CLUTEN and BASECFG1.GAM must be disabled in order to read/write in the CLUT through the User Interface.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
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
	RCLUT[7:0]							
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
	GCLUT[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
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – ACLUT[7:0] Alpha Component Entry
Indicates the 8-bit width Alpha component of the CLUT.

Bits 23:16 – RCLUT[7:0] Red Color Entry
Indicates the 8-bit width red color of the CLUT.

Bits 15:8 – GCLUT[7:0] Green Color Entry
Indicates the 8-bit width green color of the CLUT.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
Indicates the 8-bit width blue color of the CLUT.

44.7.117.Overlay 1 CLUT Register x

Name: LCDC_OVR1CLUTx
Offset: 0x0B00 + x*0x04 [x=0..255]
Reset: 0x–
Property: Read/Write

Notes:

1. The reset value is undefined because the CLUT registers are located in the embedded RAM.
2. OVR1CFG1.CLUTEN and OVR1CFG1.GAM must be disabled in order to read/write in the CLUT through the User Interface.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
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
	RCLUT[7:0]							
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
	GCLUT[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
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
Indicates the 8-bit width Alpha component of the CLUT.

Bits 23:16 – RCLUT[7:0] Red Color Entry
Indicates the 8-bit width red color of the CLUT.

Bits 15:8 – GCLUT[7:0] Green Color Entry
Indicates the 8-bit width green color of the CLUT.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
Indicates the 8-bit width blue color of the CLUT.

44.7.118.Overlay 2 CLUT Register x

Name: LCDC_OVR2CLUTx
Offset: 0x0F00 + x*0x04 [x=0..255]
Reset: 0x–
Property: Read/Write

Notes:

1. The reset value is undefined because the CLUT registers are located in the embedded RAM.
2. OVR2CFG1.CLUTEN and OVR2CFG1.GAM must be disabled in order to read/write in the CLUT through the User Interface.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
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
	RCLUT[7:0]							
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
	GCLUT[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
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
Indicates the 8-bit width Alpha component of the CLUT.

Bits 23:16 – RCLUT[7:0] Red Color Entry
Indicates the 8-bit width red color of the CLUT.

Bits 15:8 – GCLUT[7:0] Green Color Entry
Indicates the 8-bit width green color of the CLUT.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
Indicates the 8-bit width blue color of the CLUT.

44.7.119.High-End Overlay CLUT Register x

Name: LCDC_HEOCLUTx
Offset: 0x1300 + x*0x04 [x=0..255]
Reset: 0x–
Property: Read/Write

Notes:

1. The reset value is undefined because the CLUT registers are located in the embedded RAM.
2. HEOCFG1.CLUTEN and HEOCFG1.GAM have to be disabled in order to read/write in the CLUT through the User Interface.

Bit	31	30	29	28	27	26	25	24
	ACLUT[7:0]							
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
	RCLUT[7:0]							
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
	GCLUT[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
	BCLUT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	–	–	–	–	–

Bits 31:24 – ACLUT[7:0] Alpha Color Entry
Indicates the 8-bit width Alpha component of the CLUT.

Bits 23:16 – RCLUT[7:0] Red Color Entry
Indicates the 8-bit width red color of the CLUT.

Bits 15:8 – GCLUT[7:0] Green Color Entry
Indicates the 8-bit width green color of the CLUT.

Bits 7:0 – BCLUT[7:0] Blue Color Entry
Indicates the 8-bit width blue color of the CLUT.

45. Low Voltage Differential Signaling Controller (LVDSC)

45.1. Description

The Low Voltage Differential Signaling Controller (LVDSC) manages data format conversion from the LCD Controller internal DPI bus to OpenLDI LVDS output signals.

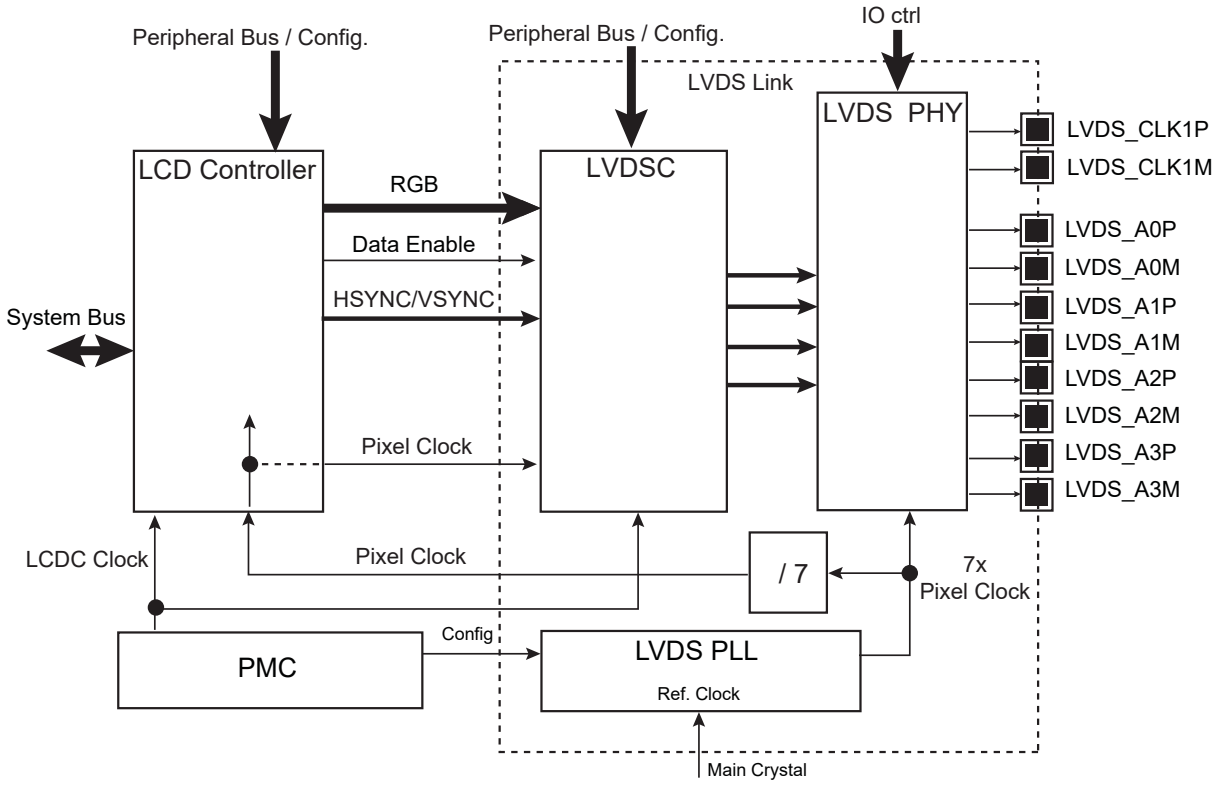
LVDSC functions include bit mapping, balanced mode management, and serializer.

45.2. Embedded Characteristics

- 18-/24-bit DPI Input Format
- VESA/JEIDA Formats
- DC-balanced Mode Allowing AC-coupling LVDS Receivers
- Single Channel, 5 LVDS Lanes (4 Data Lanes + 1 Clock Lane)

45.3. Block Diagram

Figure 45.1. LVDSC Block Diagram



45.4. I/O Lines Description

Table 45.1. I/O Lines Description

Signal Name	Description	Type
LVDS_CLK1P	Differential LVDS clock line transceiver output	Output
LVDS_CLK1M		

Table 45.1. I/O Lines Description (continued)

Signal Name	Description	Type
LVDS_A0P	Differential LVDS data 0 line transceiver output	Output
LVDS_A0M		
LVDS_A1P	Differential LVDS data 1 line transceiver output	Output
LVDS_A1M		
LVDS_A2P	Differential LVDS data 2 line transceiver output	Output
LVDS_A2M		
LVDS_A3P	Differential LVDS data 3 line transceiver output	Output
LVDS_A3M		

45.5. Product Dependencies

45.5.1. I/O Lines

The LVDS outputs are multiplexed with digital functions on the I/O lines. If the LVDS interface is not used by the application (LVDSC_CR.SER_EN), the associated I/O lines can be used for other purposes (refer to section "Peripheral Signal Multiplexing on I/O Lines").

The I/O line LVDS_A3P/3N can be used by digital peripherals when the LVDSC is enabled and configured for 18-bit pixel data (LVDSC_CFGR.LCDC_PIXSIZE=1).

Note that if the LVDS interface is used, the I/O lines are associated to it automatically without any PIO configuration.

45.5.2. Power Management

The pixel clock is not continuously provided to the LVDSC. Before using the LVDSC, the programmer must first enable the LVDS-dedicated PLL and the LCDC peripheral clock in the Power Management Controller (PMC). Then the LCDC must be configured to generate the pixel clock.

Once the LVDS PLL is enabled, the pixel clock is used as the clock for LCDC, so its GCLK is no longer needed.

45.6. Functional Description

45.6.1. Introduction

The LVDSC is driven by the LCD Controller and drives the LVDS physical interface (LVDS PHY). The LVDSC embeds the bit mapping circuitry and configures the serializers located in the LVDS PHY.

The LCDC peripheral clock and pixel clock must be enabled to configure the LVDSC.

The operating modes of the LVDSC are configured in the Configuration register (LVDSC_CFGR). The configuration status bit CS in the Status register (LVDSC_SR) must be read at 0 prior to writing a new value in LVDSC_CFGR and in the User Control Bits register (LVDSC_UCBR).

When the configuration is complete, the LVDSC must be enabled by writing a ‘1’ to the bit SER_EN in the Control register (LVDSC_CR).

LVDSC_CFGR and LVDSC_UCBR cannot be modified when LVDSC_CR.SER_EN=1.

The reserved bit of the LVDS frame can be configured in LVDSC_UCBR.

45.6.2. Unbalanced Mode Mapping Format

The LVDS Link supports VESA (OpenLDI) and JEIDA formats.

Table 45.2. SPWG/PSWG/VESA 18/24 bpp Data Mapping

Serializer Input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
A0	G0	R5	R4	R3	R2	R1	R0
A1	B1	B0	G5	G4	G3	G2	G1
A2	Data En	VSYNC	HSYNC	B5	B4	B3	B2
A3	Reserved	B7	B6	G7	G6	R7	R6

Table 45.3. JEIDA 24 bpp Data Mapping

Serializer Input	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6
A0	G2	R7	R6	R5	R4	R3	R2
A1	B3	B2	G7	G6	G5	G4	G3
A2	Data en	VSYNC	HSYNC	B7	B6	B5	B4
A3	Reserved	B1	B0	G1	G0	R1	R0

45.6.3. Supported Modes

LVDSC supports the following modes:

- 24-bit, Single Channel, DC-Unbalanced
- 18-bit, Single Channel, DC-Unbalanced
- 24-bit, Single Channel, DC-Balanced
- 18-bit, Single Channel, DC-Balanced

Figure 45.2. VESA Format, 24-bit, Single Channel, DC-Unbalanced

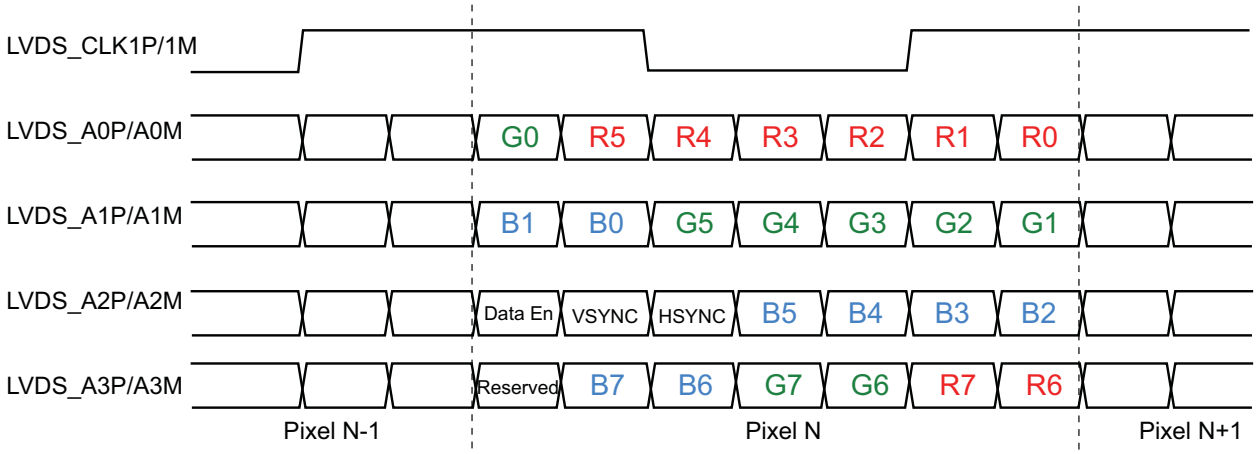
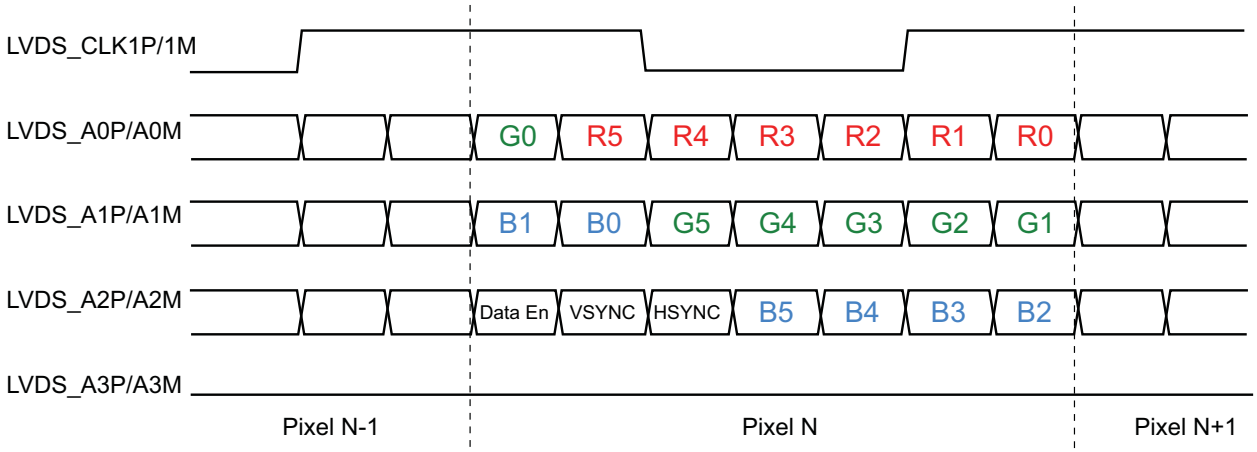


Figure 45.3. VESA Format, 18-bit, Single Channel, DC-Unbalanced



The following figures illustrate balanced operating modes. In addition to pixel and control information, an additional bit, DCBAL, is transmitted on every signal line during each cycle. DCBAL bit minimizes the short- and long-term DC bias on the signal lines. The LVDSC sends the pixel data either unmodified or inverted. The value of the DCBAL bit is 0 when the data is sent unmodified and 1 when the data is sent inverted.

Figure 45.4. 24-bit, Single Channel, DC-Balanced

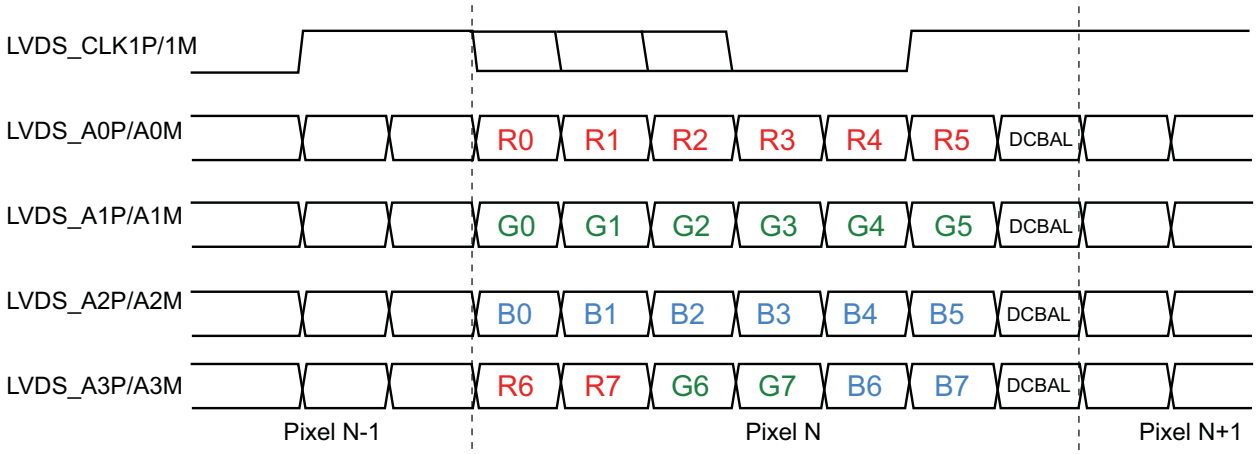
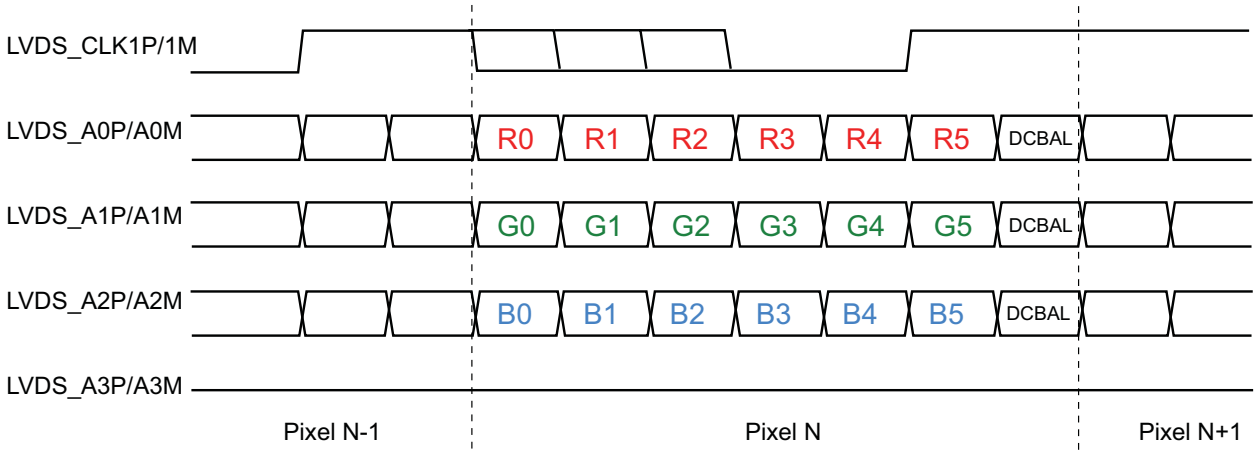


Figure 45.5. 18-bit, Single Channel, DC-Balanced



45.6.4. Register Write Protection

To prevent any single software error from corrupting LVDSC behavior, certain registers in the address space can be write-protected by setting the bit WPEN in the LVDSC Write Protection Status Register (LVDSC_WPMR).

The following registers are write-protected when LVDSC_WPMR.WPEN is set:

- LVDSC Control Register
- LVDSC Configuration Register
- LVDSC User Control Bits Register

To enhance safety, the write protection can be locked until the next system reset by configuring LCKWPEN=1 (WPEN must be 1 in the same access). When LCKWPEN=1, any further write access to the WPEN bit has no effect. If LCKWPEN=1 and WPEN=0, the lock is not activated.

45.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	LVDSC_CR	31:24								
		23:16								
		15:8								
		7:0								SER_EN
0x04	LVDSC_CFGR	31:24								
		23:16								
		15:8								
		7:0		MAPPING	DC_BAL					LCDC_PIXSIZE
0x08	LVDSC_UCBR	31:24								
		23:16								
		15:8								
		7:0						RESA3		
0x0C	LVDSC_SR	31:24								
		23:16								
		15:8								
		7:0								CS
0x10 ... 0x13	Reserved									
0x14	LVDSC_ACR	31:24						PREEMP_CLK1[2:0]		
		23:16		PREEMP_A3[2:0]				PREEMP_A2[2:0]		
		15:8		PREEMP_A1[2:0]				PREEMP_A0[2:0]		
		7:0				DCBIAS[4:0]				
0x18 ... 0xE3	Reserved									
0xE4	LVDSC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0				LCKWPEN				WPEN
0xE8	LVDSC_WPSR	31:24							SWETYP[1:0]	
		23:16								
		15:8	WPVSRC[7:0]							
		7:0					SWE			WPVS

45.7.1. LVDSC Control Register

Name: LVDSC_CR
Offset: 0x00
Reset: 0x00000000
Property: Write-only

This register can only be written if WPEN is cleared in LVDSC_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
								SER_EN
Access								R/W
Reset								0

Bit 0 – SER_EN LVDS Serializer Enable

Value	Description
0	Disables the serializer of the LVDS.
1	Enables the serializer of the LVDS.

45.7.2. LVDSC Configuration Register

Name: LVDSC_CFGR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPEN is cleared in LVDSC_WPMR.

LVDSC_CR.SER_EN must be cleared to write in LVDSC_CFGR.

Bit	31	30	29	28	27	26	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
		MAPPING	DC_BAL					LCDC_PIXSIZE
Access		R/W	R/W					R/W
Reset		0	0					0

Bit 6 – MAPPING LVDS Mapping Format (Unbalanced mode only)

Value	Name	Description
0	VESA	Maps LVDS lanes on VESA format.
1	JEIDA	Maps LVDS lanes on JEIDA format.

Bit 5 – DC_BAL DC Mode

Value	Name	Description
0	UNBALANCED	LVDS lane is DC-unbalanced.
1	BALANCED	LVDS lane is DC-balanced.

Bit 0 – LCDC_PIXSIZE LCD Controller Pixel Size

Value	Name	Description
0	24BITS	LCD controller provides 24 bits per pixel.
1	18BITS	LCD controller provides 18 bits per pixel.

45.7.3. LVDSC User Control Bits Register

Name: LVDSC_UCBR
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPEN is cleared in LVDSC_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								

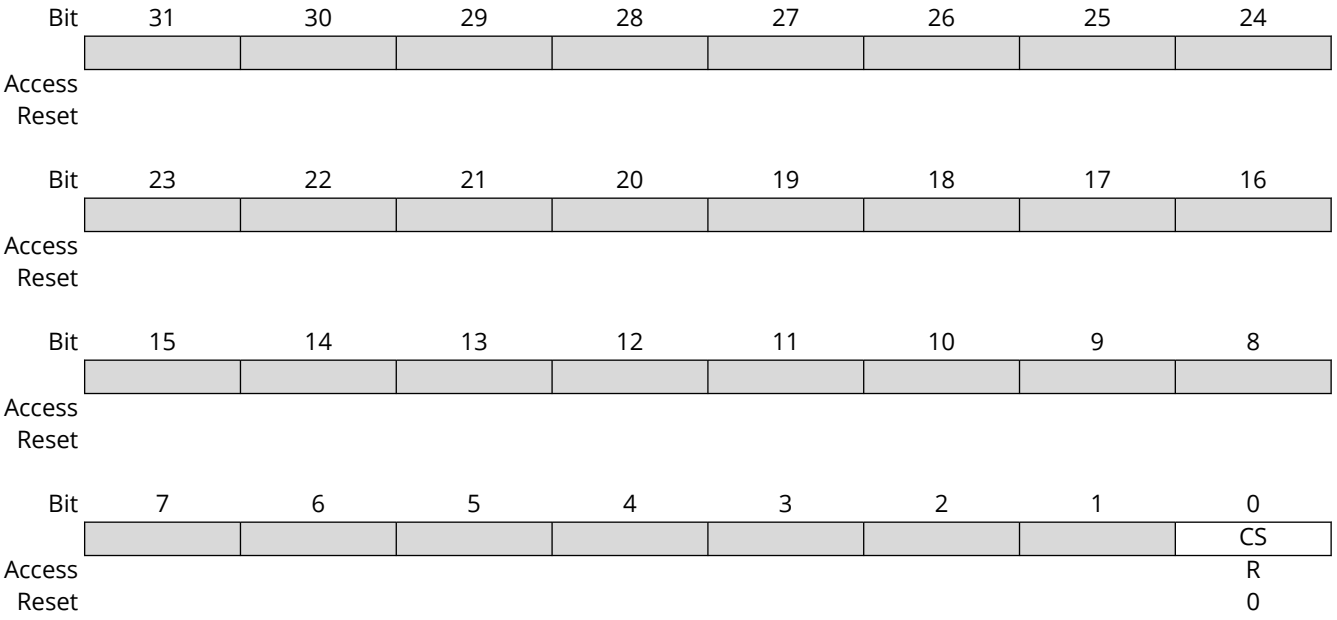
Bit	7	6	5	4	3	2	1	0
						RESA3		
Access						R/W		
Reset						0		

Bit 2 – RESA3 Lane A3 Reserved Bit Value

Value	Description
0	Asserts a logical 0 on lane A3 reserved bit location.
1	Asserts a logical 1 on lane A3 reserved bit location.

45.7.4. LVDSC Status Register

Name: LVDSC_SR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only



Bit 0 – CS Configuration Status

Value	Description
0	The LVDSC_CFGR and LVDSC_UCBR registers can be modified.
1	The LVDSC_CFGR and LVDSC_UCBR registers must not be modified.

45.7.5. LVDSC Analog Control Register

Name: LVDSC_ACR
Offset: 0x14
Reset: 0x0000000B
Property: Read/Write

This register can only be written if WPEN is cleared in LVDSC_WPMR.

Bit	31	30	29	28	27	26	25	24
						PREEMP_CLK1[2:0]		
Access						R/W	R/W	R/W
Reset								

Bit	23	22	21	20	19	18	17	16
		PREEMP_A3[2:0]				PREEMP_A2[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	15	14	13	12	11	10	9	8
		PREEMP_A1[2:0]				PREEMP_A0[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

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

Bits 26:24 – PREEMP_CLK1[2:0] Pre-Emphasis Control for CLK LVDS Lane

PREEMP_CLK	C_{load} (pF)	G_{preemp}
100	5	1.2
101		2.4
110		3.6
111		4.8

Bits 8:10, 12:14, 16:18, 20:22 – PREEMP_Ax Pre-Emphasis Control for Ax LVDS Lane

PREEMP_Ax	C_{load} (pF)	G_{preemp}
100	5	1.2
101		2.4
110		3.6
111		4.8

Bits 4:0 – DCBIAS[4:0] Common DC Bias Control for LVDS Lanes
The nominal trimming to program is LVDSC_ACR.DCBIAS = 9 to reach a typical ±350 mV DC differential output amplitude for a 100-ohm differential load.

45.7.6. LVDSC Write Protection Mode Register

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

See 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
				LCKWPEN				WPEN
Access				R/W				R/W
Reset				0				0

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

Value	Name	Description
0x4C5644	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 4 – LCKWPEN Lock Write Protection Enable Until Next System Reset

Value	Description
0	No effect.
1	Locks the WPEN bit to 1 until the next system reset if on the same access WPEN=1 and WPKEY corresponds to 0x4C5644 (“LVD” in ASCII).

Bit 0 – WPEN Write Protection Enable

Value	Description
0	Disables write protection if WPKEY corresponds to 0x4C5644 (“LVD” in ASCII).
1	Enables write protection if WPKEY corresponds to 0x4C5644 (“LVD” in ASCII).

45.7.7. LVDSC Write Protection Status Register

Name: LVDSC_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
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			WPVS
Access					R			R
Reset					0			0

Bits 25:24 – SWETYP[1:0] Software Error Type (cleared on read)

Value	Name	Description
1	WRITE_RO	The LVDSC is enabled and a write access has been performed on a read-only register.
2	UNDEF_RW	Access to an undefined address.
3	WEIRD_ACTION	Attempt to modify LVDSC_CFGR when LVDSC_CR.SER_EN=1.

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 3 – SWE Software Control Error (cleared on read)

Value	Description
0	No software error has occurred since the last read of LVDSC_WPSR.
1	A software error has occurred since the last read of LVDSC_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 0 – WPVS Write Protection Violation Status (cleared on read)

Value	Description
0	No write protect violation has occurred since the last read of LVDSC_WPSR.
1	A write protect violation has occurred since the last read of LVDSC_WPSR. The address offset of the violated register is reported into field WPVSRC.

46. 2D Graphics Engine (GFX2D)

46.1. Description

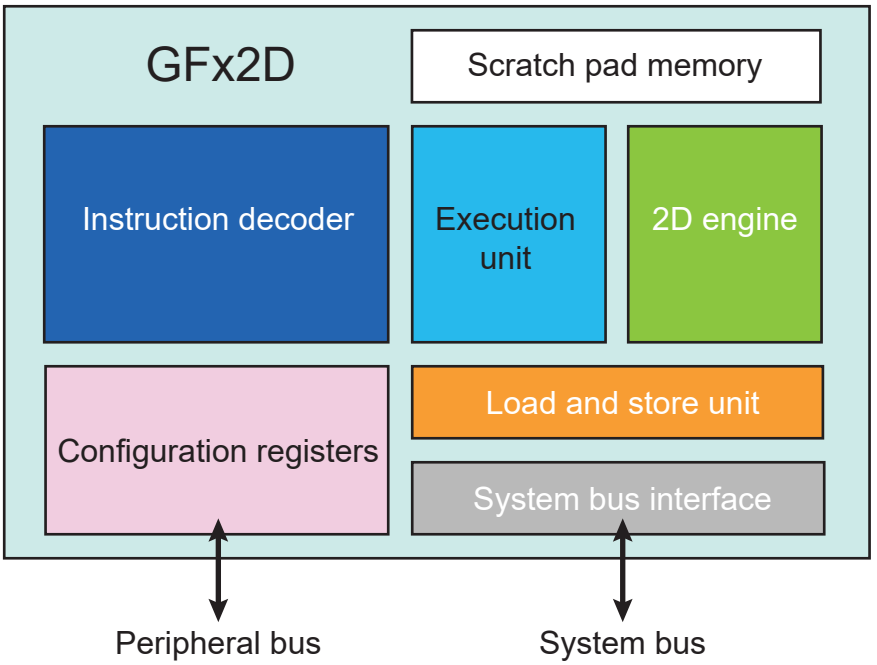
The 2D Graphics Engine (GFX2D) is a drawing engine performing memory data move operations via its system bus host port.

46.2. Embedded Characteristics

- 32-bit System Bus Host Interface
- Ring Buffer Command Interface
- FILL, COPY, BLEND and ROP Instructions
- Classic and Special Blending Functions Available
- Alpha Channel, Indexed Color and True Color Supported
- Linear-to-Cartesian Coordinate Transformation
- Automatic Transfer Regulation

46.3. Block Diagram

Figure 46.1. GFX2D Block Diagram



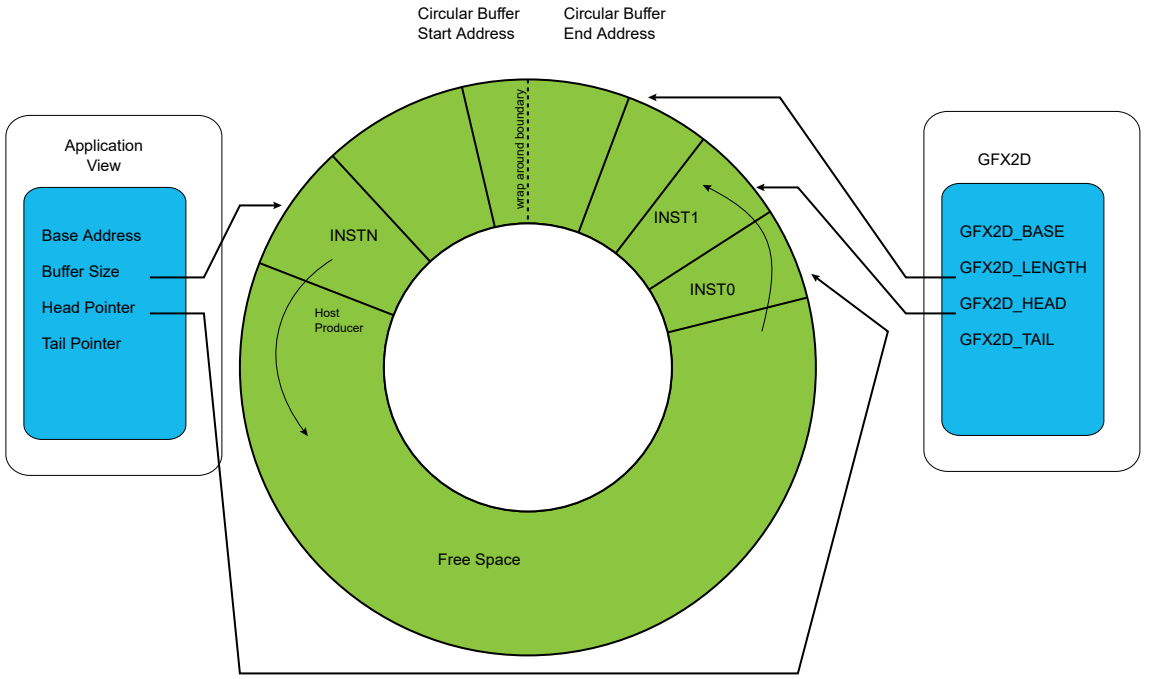
46.4. Functional Description

The GFX2D controller executes a sequence of instructions stored in a ring buffer located in the system memory. Once the instruction decoding is complete, the graphics engine executes the instruction, performs memory accesses and, optionally, raises an interrupt. A single instruction does not modify the whole set of internal registers. Host software must use a load register instruction to change the graphics context.

46.4.1. Ring Buffer Management

46.4.1.1. Ring Buffer Model Diagram

Figure 46.2. Ring Buffer Model



46.4.1.2. Ring Buffer Allocation

The instruction Ring Buffer (RB) is a contiguous memory area shared between the processor and the graphics engine. The size of the ring buffer is 24 x (GFX2D_LEN.LEN+1) bytes.

It is an efficient and effective means of passing instructions to the graphics. It is non-blocking and it facilitates batch processing of surfaces. The total RB length is software-programmable, and is a multiple of the RB memory allocation unit. A large RB is recommended to reduce CPU intervention.

Use the following software procedure to allocate a new ring buffer:

1. Wait for command completion by polling the GFX2D_GS.BUSY bit.
2. Disable the ring buffer by writing to GFX2D_GD.DISABLE.
3. Initialize the tail and head pointers by setting GFX2D_HEAD.HEAD and GFX2D_TAIL.TAIL to ‘0’.
4. Program the ring buffer base address by setting GFX2D_BASE.BASE to the base address. The address must be 24 bytes aligned.
5. Program the ring buffer length by setting GFX2D_LEN.LEN to the length.
6. Enable the ring buffer by writing to GFX2D_GE.ENABLE.

46.4.1.3. Ring Buffer Push-Pull Model

The Ring Buffer uses the producer/consumer model. The processor pushes the required instructions to render frames into the RB. The graphics consume the instructions and inform the processor that these free slots can be recycled. The RB is full when the number of free slots is equal to one. When the tail pointer is equal to the read pointer, the RB is empty. A new instruction can be pushed only if there is enough space available to store the whole instruction. When the memory is updated with one or more instructions, the head pointer is incremented with the total number of words of the batch of instructions. If the end of the RB is reached, the head pointer wraps around.

Use the following software procedure to add a command to the ring buffer queue:

1. Monitor the Ring Buffer status from the Graphics side by reading the GFX2D_TAIL register. This register is updated by the hardware when the instruction has been successfully read from the memory.
2. Verify that there is enough space left in the buffer to insert the command. The queue is full when you have inserted N-1 instructions, where N is the number of entries.
3. Write the instruction at the memory location defined by GFX2D_BASE + GFX2D_HEAD.HEAD*4.
4. Increment the local head pointer and the GFX2D_HEAD.HEAD register to inform the graphics that new instructions have been added to the queue.
5. Wait for command completion by polling the BUSY bit in the GFX2D_GS register or wait for an interrupt if programmed.

46.4.1.4.Ring Buffer Disable

Use the following software procedure to halt the GFX2D:

1. Wait for command completion by polling the GFX2D_GS.BUSY bit.
2. Disable the ring buffer by writing '1' to GFX2D_GD.DISABLE.
3. Reset the tail and head pointers by writing '0' to GFX2D_HEAD.HEAD and GFX2D_TAIL.TAIL.

46.4.2. GFX2D Surface Memory Format

46.4.2.1.Source Surface Memory Format

46.4.2.1.1.4-bit Alpha Channel with 4-bit Indexed Color

Table 46.1. 4-bit Alpha Channel 4-bit Indexed Color Memory Mapping, Little Endian Organization

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
Pixel A4IDX4	A3				IDX3				A2				IDX2				A1				IDX1				A0				IDX0			

The 4-bit Indexed Color mode uses the Color Look-Up Table (CLUT) defined by the GFX2D_CFGx.IDXCX bit.

Table 46.2. 32-bpp Final Color Used at the Input of the Alpha Blending Stage, with Alpha Expansion

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
1st pixel	A0				A0				CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							
2nd pixel	A1				A1				CLUT[IDX1].R								CLUT[IDX1].G								CLUT[IDX1].B							
3rd pixel	A2				A2				CLUT[IDX2].R								CLUT[IDX2].G								CLUT[IDX2].B							
4th pixel	A3				A3				CLUT[IDX3].R								CLUT[IDX3].G								CLUT[IDX3].B							

46.4.2.1.2.8-bit Alpha Channel

Table 46.3. 8-bit Alpha Channel Memory Mapping, Little Endian Organization

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
Pixel 8 bpp	A3								A2								A1								A0							

The alpha channel is loaded from the memory and the color information is retrieved from the general-purpose register defined by the instruction's REG field.

Table 46.4. Constant Color in the General-Purpose Register Defined by Instruction's REG Field

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

Table 46.4. Constant Color in the General-Purpose Register Defined by Instruction's REG Field (continued)

Mem. addr.	0x3	0x2	0x1	0x0
Pixel 8 bpp	A_c	R_c	G_c	B_c

Table 46.5. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

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
1st pixel	A0								R_c								G_c								B_c							
2nd pixel	A1								R_c								G_c								B_c							
3rd pixel	A2								R_c								G_c								B_c							
4th pixel	A3								R_c								G_c								B_c							

46.4.2.1.3.8-bit Indexed Color

Table 46.6. 8-bit Indexed Color Memory Mapping, Little Endian Organization

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
Pixel 8 bpp	IDX3								IDX2								IDX1								IDX0							

The 8-bit Indexed Color mode uses the Color Look-Up Table (CLUT) defined by the GFX2D_CFGx.IDXCX bit.

Table 46.7. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

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
1st pixel	CLUT[IDX0].A								CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							
2nd pixel	CLUT[IDX1].A								CLUT[IDX1].R								CLUT[IDX1].G								CLUT[IDX1].B							
3rd pixel	CLUT[IDX2].A								CLUT[IDX2].R								CLUT[IDX2].G								CLUT[IDX2].B							
4th pixel	CLUT[IDX3].A								CLUT[IDX3].R								CLUT[IDX3].G								CLUT[IDX3].B							

46.4.2.1.4.8-bit Alpha Channel with 8-bit Indexed Color

Table 46.8. 8-bit Alpha Channel 8-bit Indexed Color Memory Mapping, Little Endian Organization

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
Pixel 8 bpp	A1								IDX1								A0								IDX0							

Table 46.9. 32-bpp Final Color Used at the Input of the Alpha Blending Stage

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
1st pixel	A0								CLUT[IDX0].R								CLUT[IDX0].G								CLUT[IDX0].B							
2nd pixel	A1								CLUT[IDX1].R								CLUT[IDX1].G								CLUT[IDX1].B							

46.4.2.1.5.12-bpp Memory Mapping, RGB 4:4:4

Table 46.10. 12-bpp Memory Mapping, Little Endian Organization

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
Pixel 12 bpp	–				R1[3:0]				G1[3:0]				B1[3:0]				–				R0[3:0]				G0[3:0]				B0[3:0]			

Table 46.11. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

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
1st pixel	0xFF								R0[3:0]				R0[3:0]				G0[3:0]				G0[3:0]				B0[3:0]				B0[3:0]			
2nd pixel	0xFF								R1[3:0]				R1[3:0]				G1[3:0]				G1[3:0]				B1[3:0]				B1[3:0]			

46.4.2.1.6.16-bpp Memory Mapping with 4-bit Alpha Channel, ARGB 4:4:4:4

Table 46.12. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

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
1st pixel	A0[3:0]				A0[3:0]				R0[3:0]				R0[3:0]				G0[3:0]				G0[3:0]				B0[3:0]				B0[3:0]			
2nd pixel	A1[3:0]				A1[3:0]				R1[3:0]				R1[3:0]				G1[3:0]				G1[3:0]				B1[3:0]				B1[3:0]			

46.4.2.1.7.16-bpp Memory Mapping with Transparency Bit, TRGB 5:5:5

Table 46.13. 16-bpp Memory Mapping, Little Endian Organization

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
Pixel 4 bpp	T1	R1[4:0]				G1[4:0]				B1[4:0]							T0	R0[4:0]				G0[4:0]				B0[4:0]						

Table 46.14. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

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
1st pixel	T0	T0	T0	T0	T0	T0	T0	T0	R0[4:0]					R0[4:2]			G0[4:0]					G0[4:2]			B0[4:0]					B0[4:2]		
2nd pixel	T1	T1	T1	T1		T1	T1	T1	R1[4:0]					R1[4:2]			G1[4:0					G1[4:2]			B1[4:0]					B1[4:2]		

46.4.2.1.8.16-bpp Memory Mapping with Transparency Bit, RGBT 5:5:5:1

Table 46.15. 16-bpp Memory Mapping, Little Endian Organization

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
Pixel 4 bpp	R1[4:0]					G1[4:0]				B1[4:0]						T1	R0[4:0]					G0[4:0]				B0[4:0]						T0

Table 46.16. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

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
1st pixel	T0	T0	T0	T0	T0	T0	T0	T0	R0[4:0]					R0[4:2]			G0[4:0]					G0[4:2]			B0[4:0]					B0[4:2]		
2nd pixel	T1	T1	T1	T1		T1	T1	T1	R1[4:0]					R1[4:2]			G1[4:0					G1[4:2]			B1[4:0]					B1[4:2]		

46.4.2.1.9.16-bpp Memory Mapping with Alpha Channel, RGB 5:6:5

Table 46.17. 16-bpp Memory Mapping, Little Endian Organization

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
Pixel 16 bpp	R1[4:0]					G1[5:0]				B1[4:0]							R0[4:0]					G0[5:0]				B0[4:0]						

Table 46.18. 32-bpp Final Color Used at the Input of the Alpha Blending Stage with Color Expansion

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
1st pixel	0xFF								R0[4:0]					R0[4:2]			G0[5:0]						G0[5:4]		B0[4:0]					B0[4:2]		
2nd pixel	0xFF								R1[4:0]					R1[4:2]			G1[5:0]						G1[5:4]		B1[4:0]					B1[4:2]		

46.4.2.1.10.32-bpp Memory Mapping, ARGB 8:8:8:8

Table 46.19. 32-bpp Memory Mapping, Little Endian Organization

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
Pixel 32 bpp	A0[7:0]								R0[7:0]								G0[7:0]								B0[7:0]							

46.4.2.1.11.32-bpp Memory Mapping, RGBA 8:8:8:8

Table 46.20. 32-bpp Memory Mapping, Little Endian Organization

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
Pixel 32 bpp	R0[7:0]								G0[7:0]								B0[7:0]								A0[7:0]							

46.4.2.2.Destination Surface Memory Format

46.4.2.2.1.4-bit Alpha Channel with 4-bit Luminance

The 24-bpp RGB final pixel is converted into an 8-bit luminance value and then truncated.

$$Y = \frac{0.299R + 0.587G + 0.114B}{16}$$

Table 46.21. 4-bit Alpha Channel with 4-bit Luminance Memory Mapping, Little Endian Organization

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
Pixel A4IDX4	A3				Y3				A2				Y2				A1				Y1				A0				Y0			

46.4.2.2.2.8-bit Alpha Channel

Table 46.22. 8-bit Alpha Memory Mapping, Little Endian Organization

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
Alpha	A3								A2								A1								A0							

46.4.2.2.3.8-bit Luminance Channel

The 32-bpp ARGB final pixel is converted into an 8-bit luminance value.

$$Y = 0.299R + 0.587G + 0.114B$$

Table 46.23. 8-bit Luminance Memory Mapping, Little Endian Organization

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
Alpha	Y3								Y2								Y1								Y0							

46.4.2.2.4.8-bit Alpha Channel with 8-bit Luminance

$$Y = 0.299R + 0.587G + 0.114B$$

The 32-bpp ARGB final pixel is converted into an 8-bit luminance value.

Table 46.24. 8-bit Alpha with 8-bit Luminance Memory Mapping, Little Endian Organization

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
Alpha	A1								Y1								A0								Y0							

46.4.2.3.Color Look-Up Table (CLUT)

The GFX2D module includes two CLUTs. Each CLUT includes 256 entries containing 32 bits. Each entry contains a 32-bpp ARGB color.

Table 46.25. Color Look-Up Table Entry

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
Color entry	A								R								G								B							

The CLUT can be loaded using an LDR instruction with the LDRC bit set. The LDRCID bit indicates the targeted CLUT. The LDRCS bit provides the amount of data to be loaded. Using a reduced CLUT mapping (i.e., 16 entries) reduces the number of accessible colors but improves loading time. The data structure is loaded from a memory location defined by the address in the REG field.

46.4.3. GFX2D Visible Registers

46.4.3.1.Register Naming

A set of 15 registers is defined and accessible through the use of the load and store architecture.

Table 46.26. Visible Registers

Value	Name	Description
0	GFX2D_PA0	Surface 0 Physical Address Register
1	GFX2D_PITCH0	Surface 0 Pitch Register
2	GFX2D_CFG0	Surface 0 Configuration Register
3	GFX2D_PA1	Surface 1 Physical Address Register
4	GFX2D_PITCH1	Surface 1 Pitch Register
5	GFX2D_CFG1	Surface 1 Configuration Register
6	GFX2D_PA2	Surface 2 Physical Address Register
7	GFX2D_PITCH2	Surface 2 Pitch Register
8	GFX2D_CFG2	Surface 2 Configuration Register
9	GFX2D_PA3	Surface 3 Physical Address Register
10	GPREG0	General-Purpose Register 0
11	GPREG1	General-Purpose Register 1
12	GPREG2	General-Purpose Register 2
13	GPREG3	General-Purpose Register 3
14	GPREG4	General-Purpose Register 4
15	GPREG5	General-Purpose Register 5

General-purpose registers are generic registers. They can hold:

- an address,
- a constant color for a FILL or BLEND operation,
- a constant alpha value for a BLEND, or
- a user-defined 32-bit data for a STORE operation.

Table 46.27. General-Purpose Registers

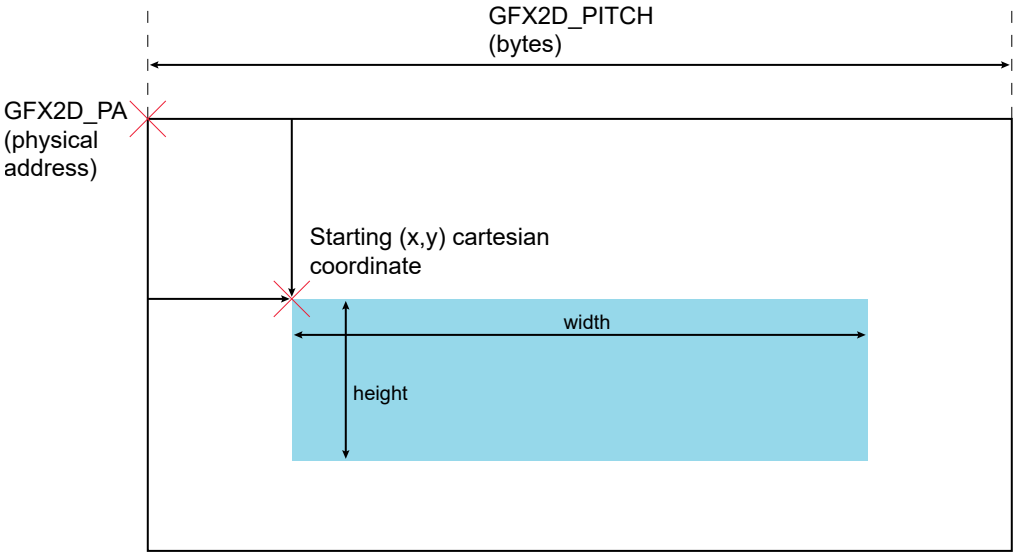
Operation/ Instruction	General-Purpose Register
FILL	The register holds the 32-bit fill color pattern when the FILL arguments are less than 2. In that case, the color is defined by the REG field of the FILL instruction.
BLEND	The register holds the constant 32-bit color (A_c, R_c, G_c, B_c). It can be used for CONSTANT_ALPHA, CONSTANT_ALPHA parameters.

Table 46.27. General-Purpose Registers (continued)

Operation/ Instruction	General-Purpose Register
STR	The store operation requires a valid address defined by the REGAD field register index, and a data defined by the REG field register index. The general-purpose register can hold a 32-bit address or data.
LDR	The LDR instruction can load a general-purpose register.

46.4.3.2.Register Descriptions

Figure 46.3. Surface Registers



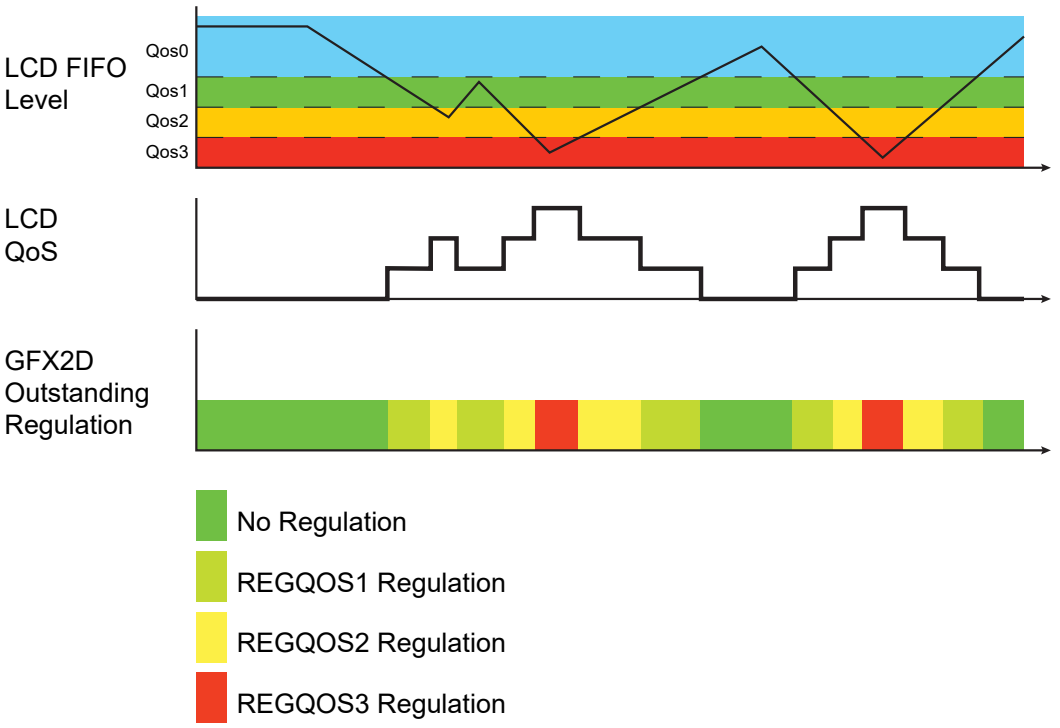
46.4.4. Traffic Balancing Using Outstanding Regulation

Global system performance is significantly impacted when the LCD controller is reading a frame buffer from memory to refresh the LCD screen. To ensure optimal performance, the LCD forwards its quality of service to other system bus hosts to limit their usage of the bus. The GFX2D frame rate is limited to allow latency-critical LCD operation.

Use the following procedure to activate traffic balancing:

1. Program the GFX2D_GC.REGQOS1, GFX2D_GC.REGQOS2 and GFX2D_GC.REGQOS3 registers and enable the outstanding issuance regulation by setting the GFX2D_GC.REGEN.
2. Configure the GFX2D_PCX register, and read the GFX2D_MCX register to adjust the regulation.

Figure 46.4. Traffic Balancing Using Outstanding Regulation



46.4.5. Data Flow Instructions

46.4.5.1. LDR Instruction

The LDR instruction length is variable and depends on the context.

- When the LDR instruction is used to load a register, the instruction length is 2 words.
- When the LDR instruction is used to load a CLUT, the instruction length is a single word.

46.4.5.1.1.LDR_WD0

Name: LDR_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						LDRCS	LDRCID	LDRC
Access								
Reset								

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

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0x8'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
This is the target register for a load operation. When the LDR instruction is used to load a Color Look-Up Table, the REG field points to the general-purpose register that holds the physical base address of the table.

Bit 10 – LDRCS Color Look-Up Table Size

Value	Description
0	256 entries.
1	16 entries.

Bit 9 – LDRCID Color Look-Up Table Selection

Value	Description
0	Table 0 is selected.
1	Table 1 is selected.

Bit 8 – LDRC Load Color Look-Up Table

Value	Description
0	Loads the register with the immediate value.
1	Loads the Color Look-Up Table with the content of the memory.

Bits 2:0 – ARGS[2:0] Arguments Length
This field must be set to '0'.

46.4.5.1.2.LDR_WD1

Name: LDR_WD1

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access								
Reset								

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

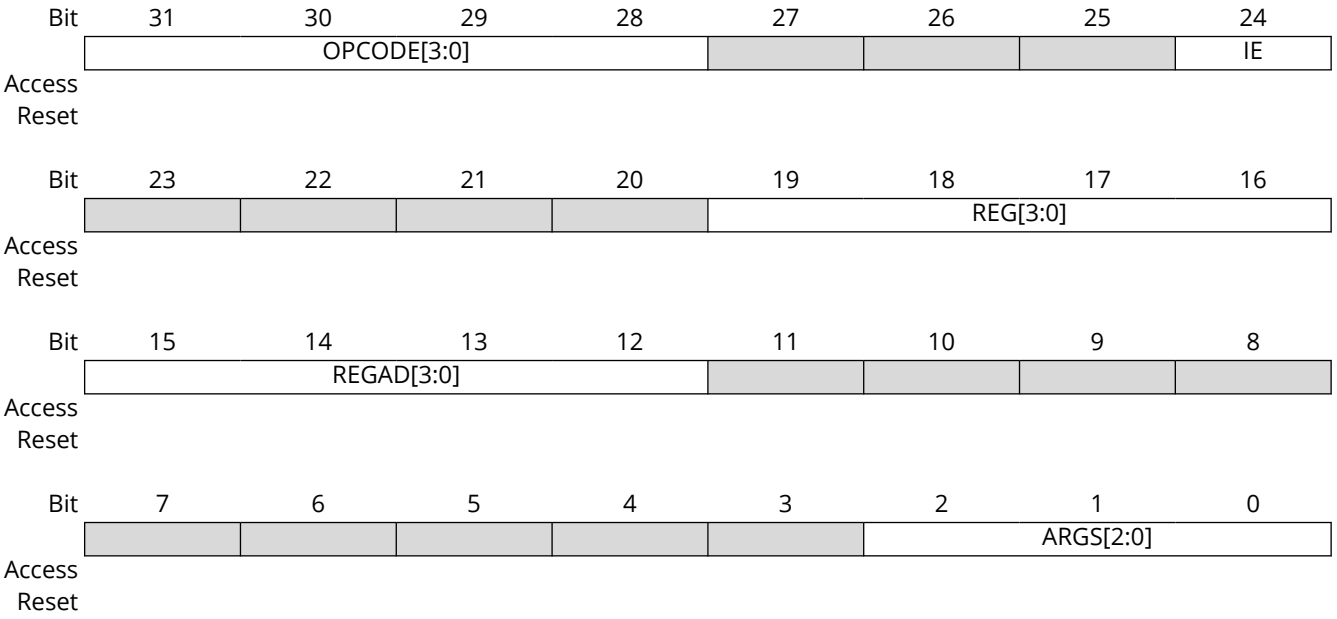
Bits 31:0 – DATA[31:0] 32-bit Data
The 32-bit data word loaded into the register.

46.4.5.2.STR Instruction

The instruction length is 1 word.

46.4.5.2.1.STR_WD0

Name: STR_WD0



Bits 31:28 – OPCODE[3:0] Instruction Code
Must be set to '0x9'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
Points to the general-purpose register that holds the data written to memory.

Bits 15:12 – REGAD[3:0] Register Identifier for Stored Address
Points to the general-purpose register that holds the address of the memory store operation.

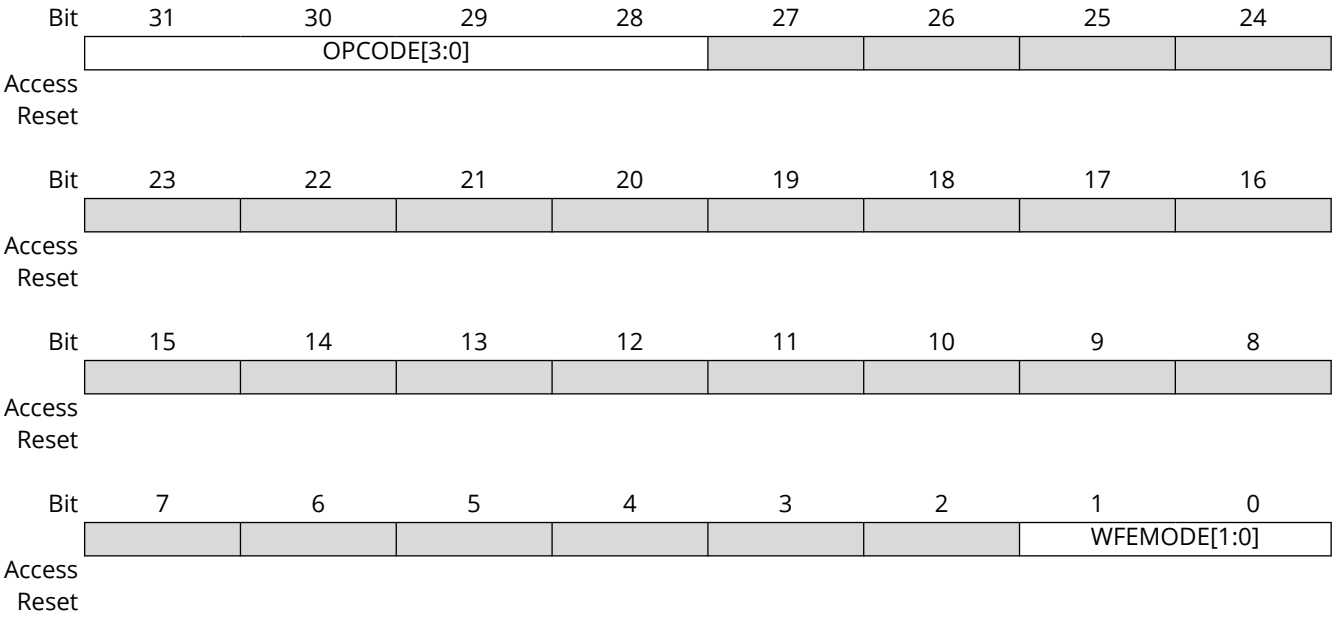
Bits 2:0 – ARGS[2:0] Arguments Length
This field must be set to '0'.

46.4.5.3.WFE Instruction

The instruction length is 1 word.

46.4.5.3.1.WFE_WD0

Name: WFE_WD0



Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xA'.

Bits 1:0 – WFEMODE[1:0] Wait For Event Mode

Value	Description
0	Resume from WFE state when detecting any edge on event signal
1	Resume from WFE state when detecting rising edge on event signal
2	Resume from WFE state when detecting falling edge on event signal

46.4.6. Graphics Instructions

46.4.6.1.FILL Instruction

The FILL instruction length is 4.

Figure 46.5. Fill Operation

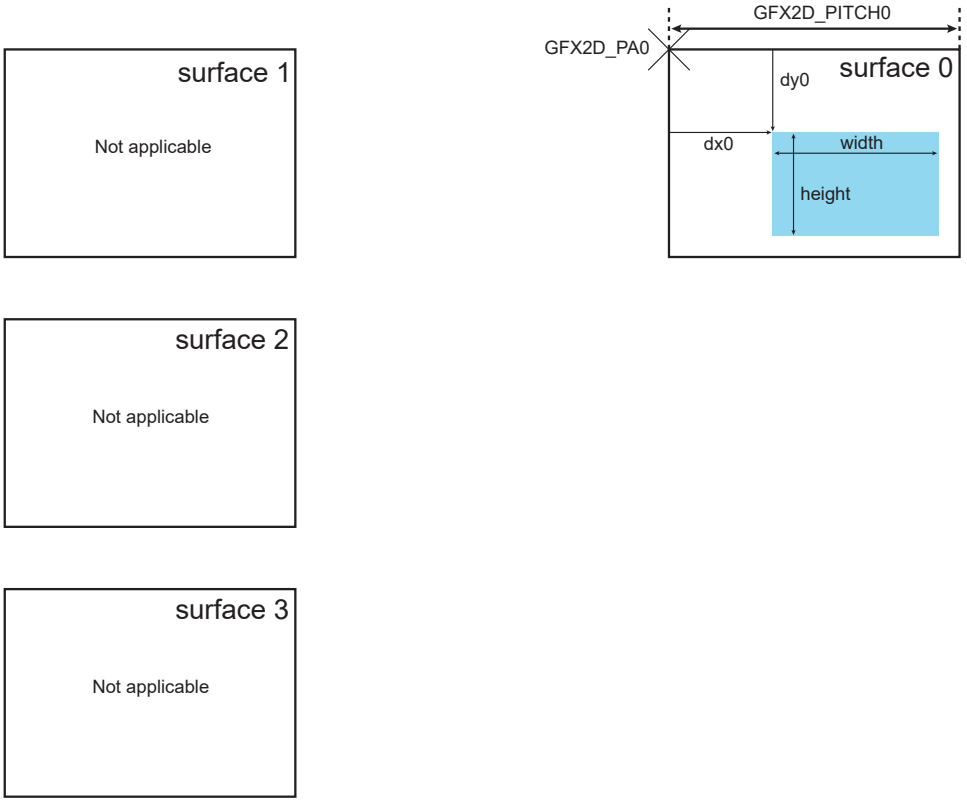


Table 46.28. FILL Instructions

Word Identifier	Description
FILL_WD0	The instruction header
FILL_WD1	The width and height of the destination-filled area
FILL_WD2	The x and y offsets of the filled area
FILL_WD3	The fill color: a 32-bit ARGB color resized to the area pixel format

46.4.6.1.1.FILL_WD0

Name: FILL_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DIR[1:0]	
Access								
Reset								

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

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xB'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
This field is relevant when ARGS is less than 2. In that case, the REG value points to a register with a valid 32-bit ARGB color.

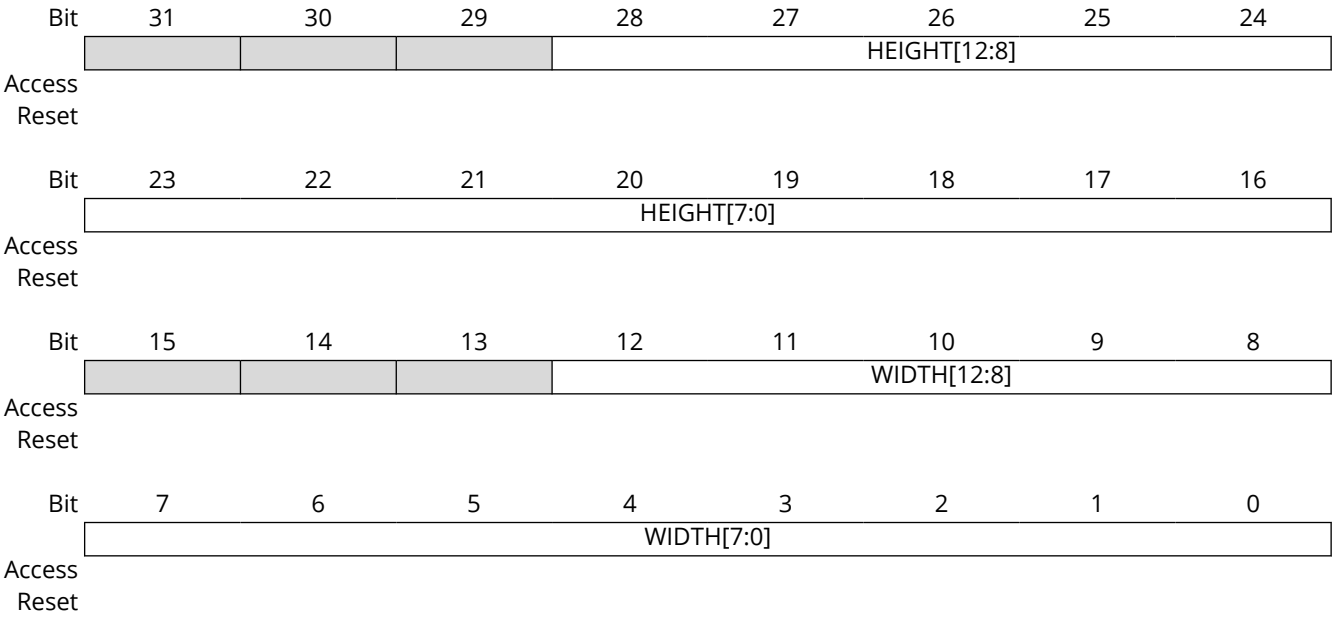
Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be less than or equal to 2.

46.4.6.1.2.FILL_WD1

Name: FILL_WD1

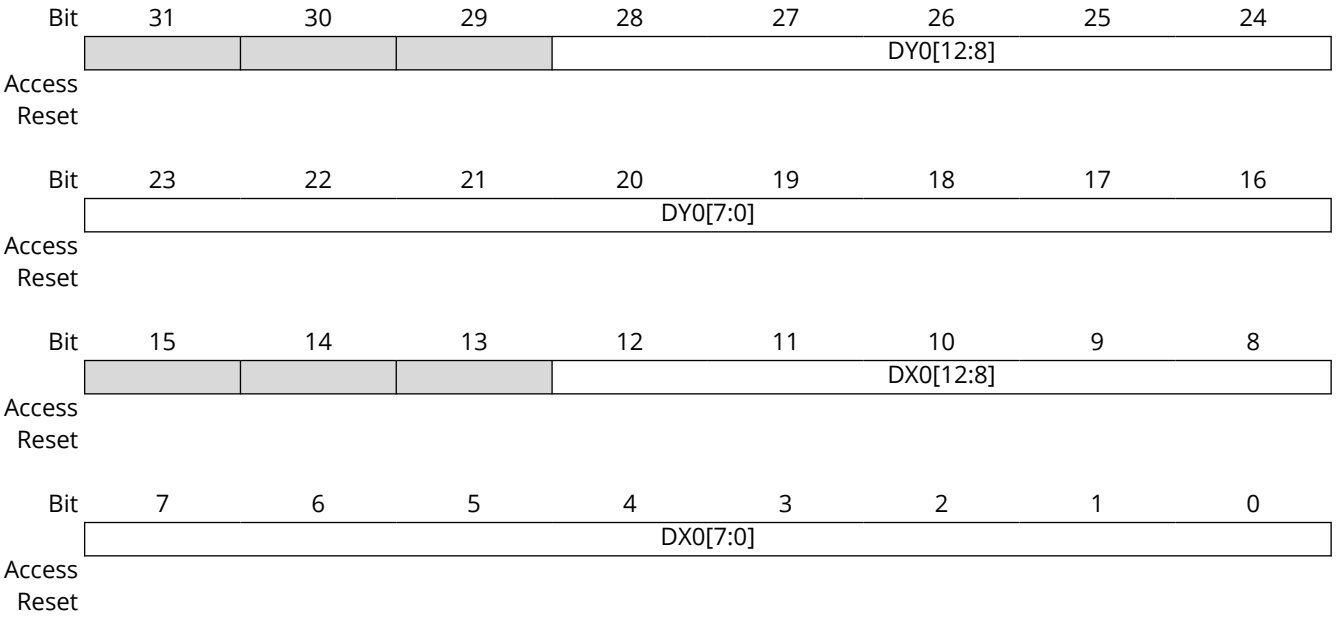


Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

46.4.6.1.3.FILL_WD2

Name: FILL_WD2



Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

46.4.6.1.4.FILL_WD3

Name: FILL_WD3

Bit	31	30	29	28	27	26	25	24
	A[7:0]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	R[7:0]							
Access								
Reset								

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

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

Bits 31:24 – A[7:0] Alpha Color Component Value
Program this field with the desired Alpha value.

Bits 23:16 – R[7:0] Red Color Component Value
Program this field with the desired Red value.

Bits 15:8 – G[7:0] Green Color Component Value
Program this field with the desired Green value.

Bits 7:0 – B[7:0] Blue Color Component Value
Program this field with the desired Blue value.

46.4.6.2.COPY Instruction

The instruction length is 4 words.

Figure 46.6. COPY Operation

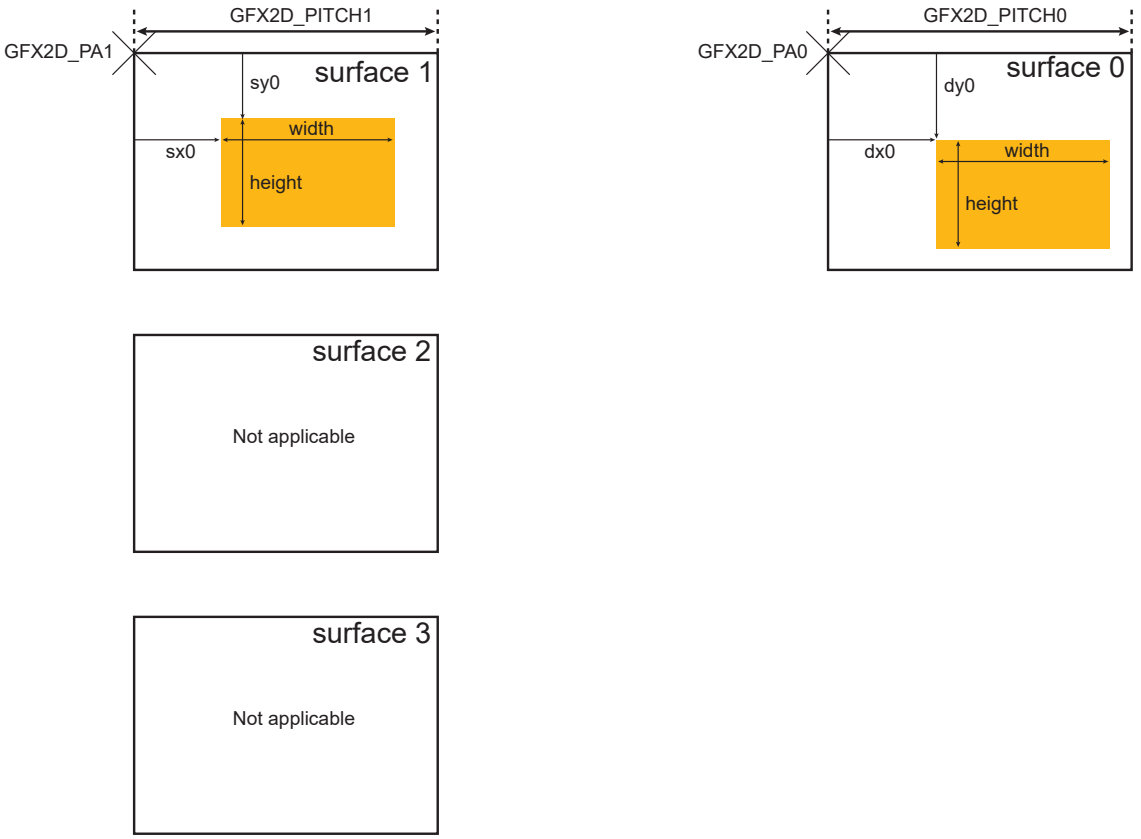
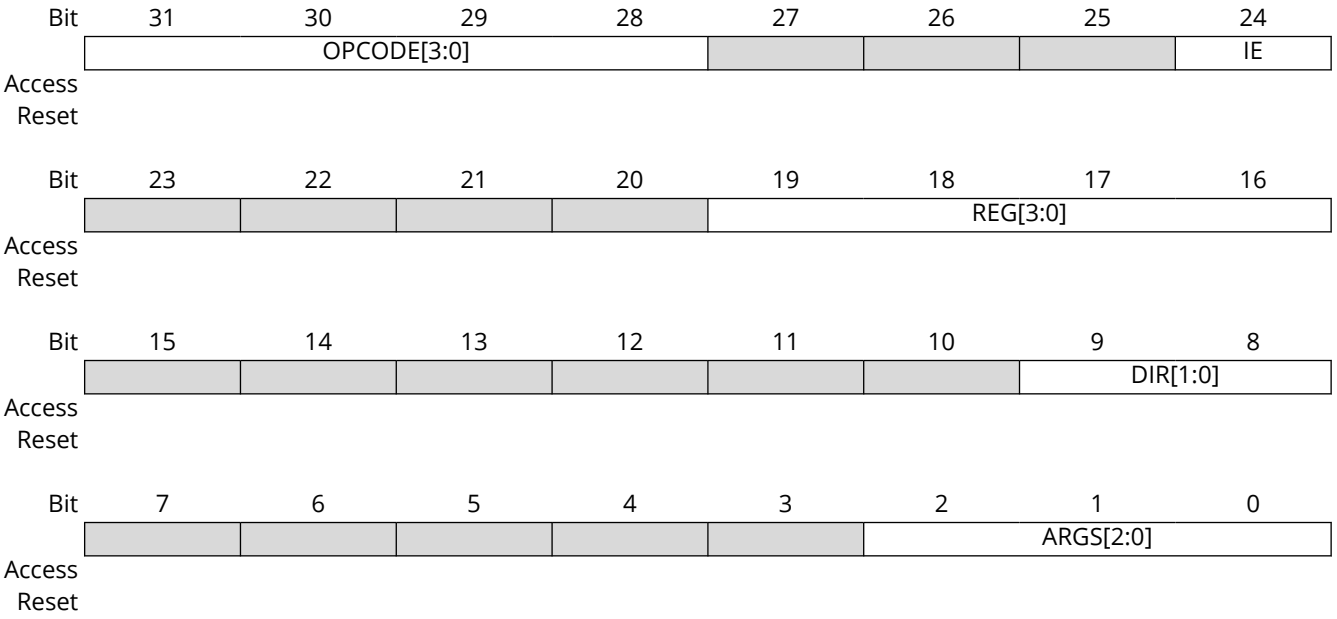


Table 46.29. COPY Instructions

Word Identifier	Description	Surface Register
COPY_WD0	The instruction header	Not applicable
COPY_WD1	The width and height of the destination area	Surface 0 register
COPY_WD2	The x and y offsets of the destination area	Surface 0 register
COPY_WD3	The x and y offsets of the source area	Surface 1 register

46.4.6.2.1.COPY_WD0

Name: COPY_WD0



Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xC'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

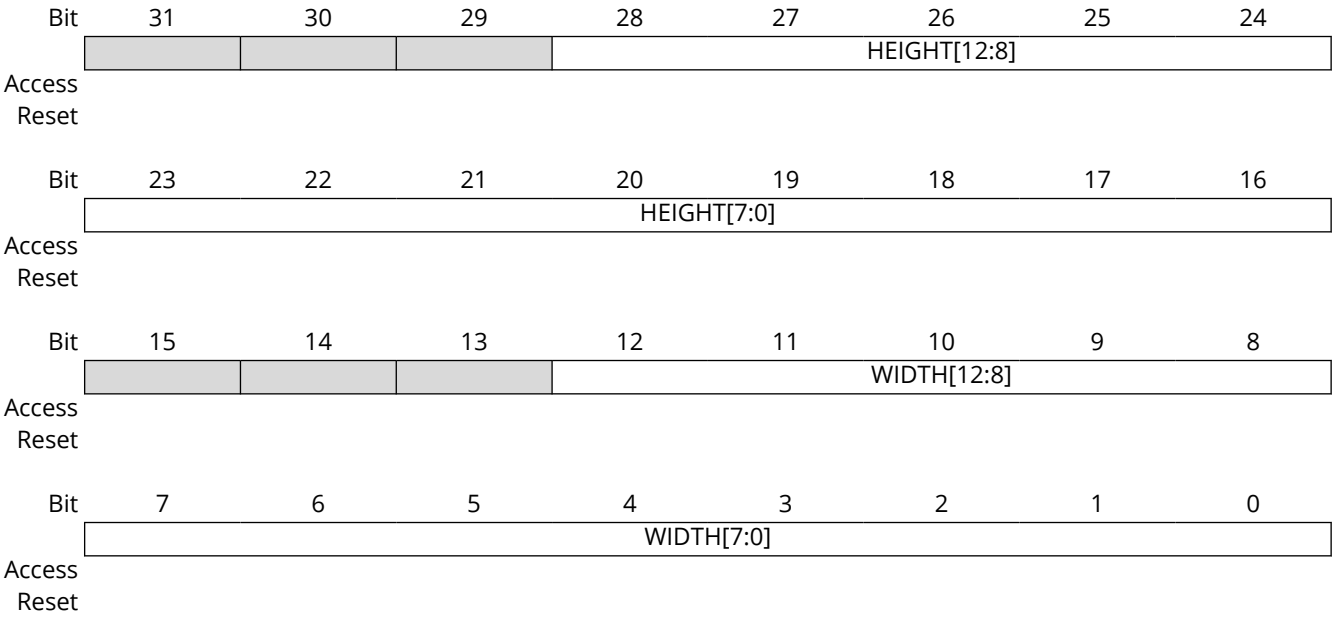
Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '2'.

46.4.6.2.2.COPY_WD1

Name: COPY_WD1

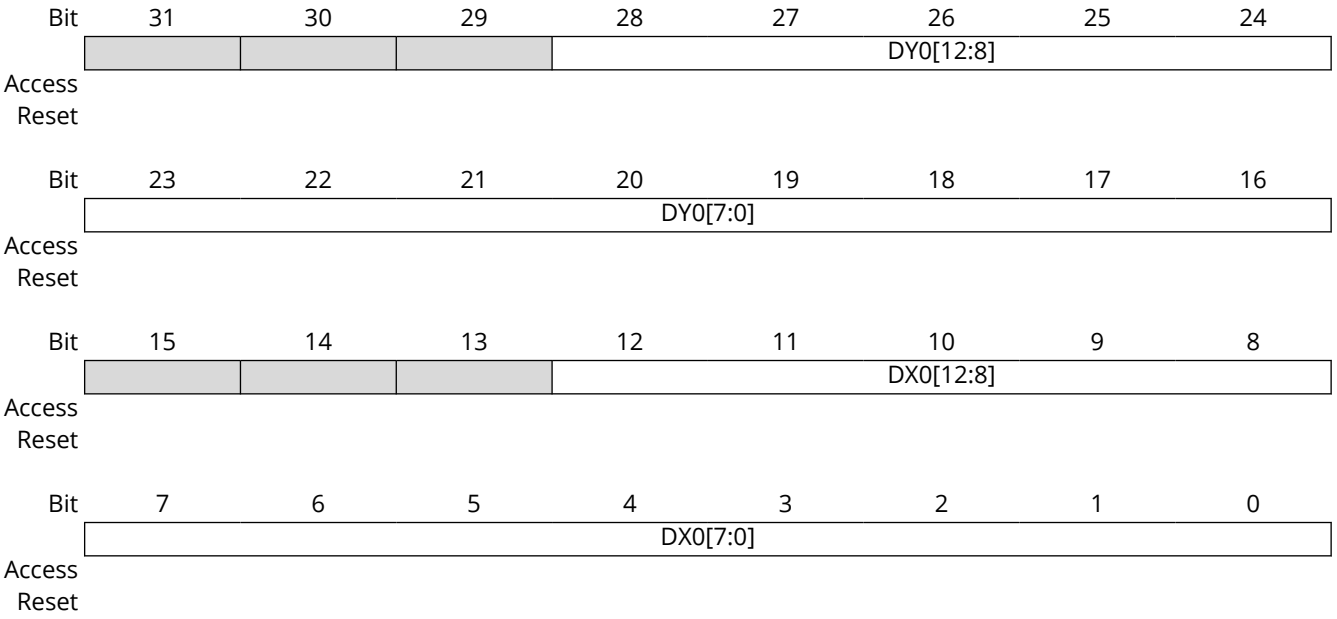


Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

46.4.6.2.3.COPY_WD2

Name: COPY_WD2

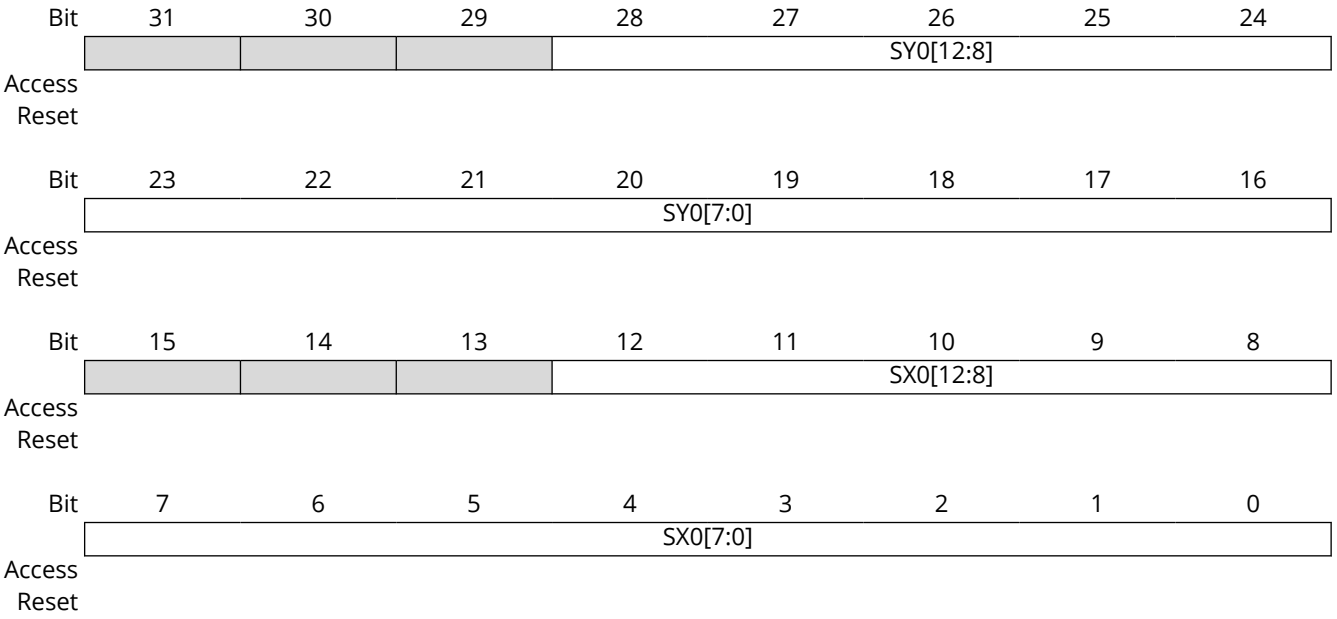


Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

46.4.6.2.4.COPY_WD3

Name: COPY_WD3



Bits 28:16 – SY0[12:0] Source Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source X Position
This field indicates the horizontal position of the window.

46.4.6.3.BLEND Instruction

The instruction length is 6 words.

Figure 46.7. BLEND Operation

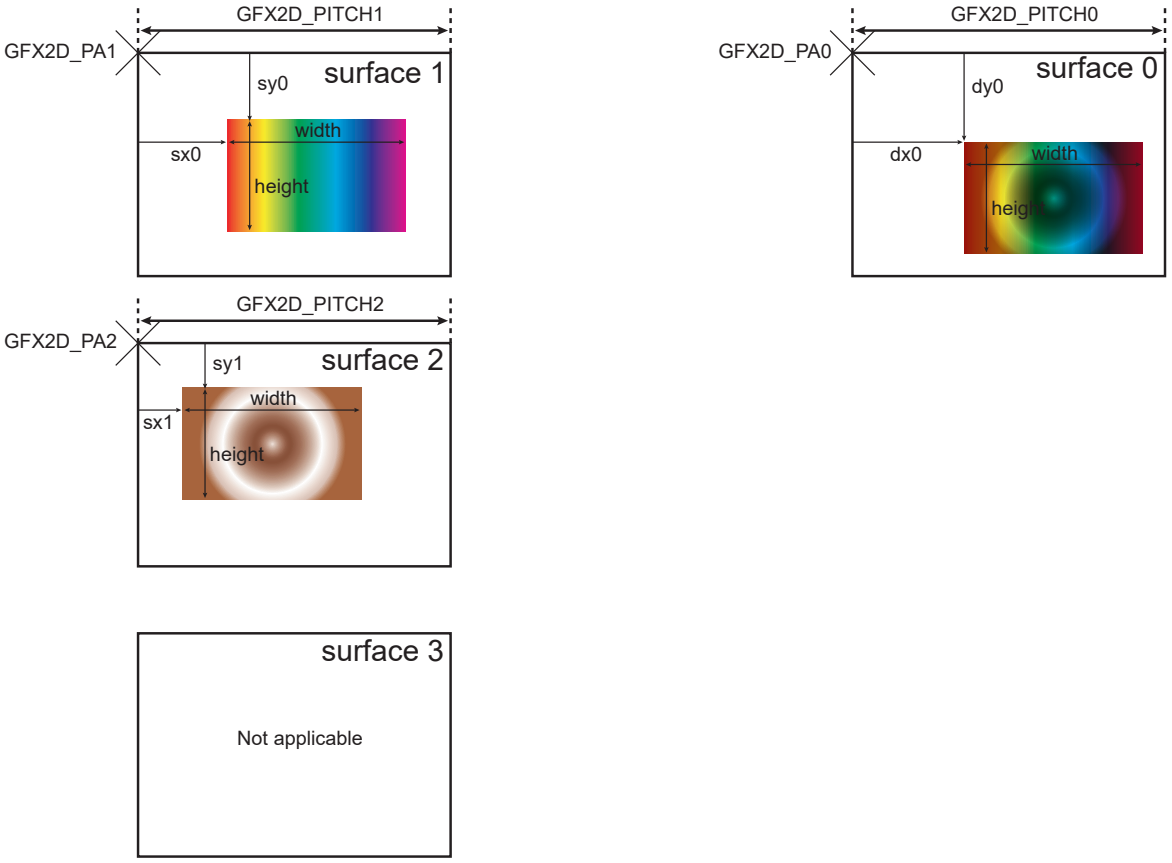


Table 46.30. BLEND Instructions

Word Identifier	Description	Surface Register
BLEND_WD0	The Instruction header	Not applicable
BLEND_WD1	The width and height of the destination area (C_f)	Surface 0 registers
BLEND_WD2	The x and y offsets of the destination area (C_f)	Surface 0 registers
BLEND_WD3	The x and y offsets of source 0 (C_d)	Surface 1 registers
BLEND_WD4	The x and y offsets of source 1 (C_s)	Surface 2 registers
BLEND_WD5	This register configures the blending functions and factors	Not applicable

46.4.6.3.1.BLEND_WD0

Name: BLEND_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DREG[3:0]				REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							DIR[1:0]	
Access								
Reset								

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

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xD'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 23:20 – DREG[3:0] Destination Register Identifier
When a constant is used for the destination surface, the constant value is located in the register defined by the DREG value.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

Bits 9:8 – DIR[1:0] Transfer Direction

Value	Name	Description
0	XY00	Horizontal forward, vertical forward
1	XY01	Horizontal forward, vertical backward
2	XY10	Horizontal backward, vertical forward
3	XY11	Horizontal backward, vertical backward

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '4'.

46.4.6.3.2.BLEND_WD1

Name: BLEND_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

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

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

46.4.6.3.3.BLEND_WD2

Name: BLEND_WD2

Bit	31	30	29	28	27	26	25	24
				DY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				DX0[12:8]				
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	DX0[7:0]							
Access								
Reset								

Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

46.4.6.3.4.BLEND_WD3

Name: BLEND_WD3

Bit	31	30	29	28	27	26	25	24
				SY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX0[12:8]				
Access								
Reset								

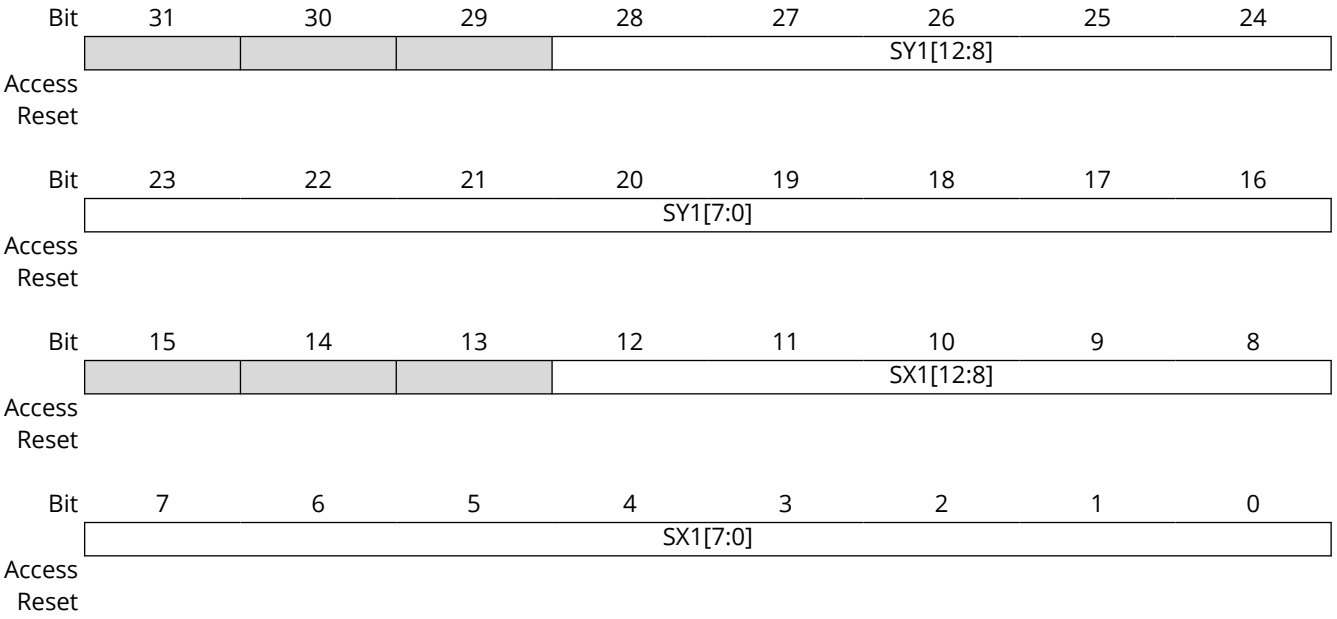
Bit	7	6	5	4	3	2	1	0
	SX0[7:0]							
Access								
Reset								

Bits 28:16 – SY0[12:0] Source 0 Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source 0 X Position
This field indicates the horizontal position of the window.

46.4.6.3.5.BLEND_WD3

Name: BLEND_WD3

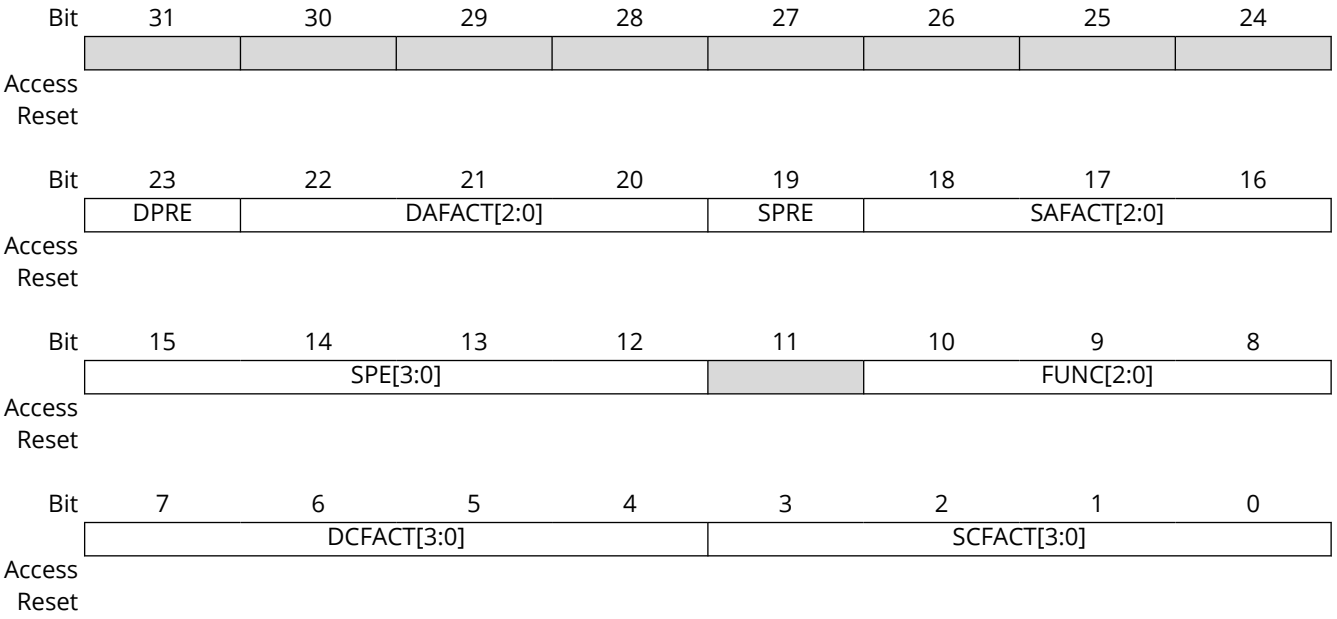


Bits 28:16 – SY1[12:0] Source 1 Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX1[12:0] Source 1 X Position
This field indicates the horizontal position of the window.

46.4.6.3.6.BLEND_WD5

Name: BLEND_WD5



Bit 23 – DPRE Destination Premult

Value	Description
0	Destination pre-multiplication is disabled.
1	Destination pre-multiplication is enabled. Pixels from destination surface are multiplied by constant before blending.

Bits 22:20 – DAFACT[2:0] Destination Alpha Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_ALPHA	(A_s, A_s, A_s, A_s)
3	ONE_MINUS_SRC_ALPHA	(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)
4	DST_ALPHA	(A_d, A_d, A_d, A_d)
5	ONE_MINUS_DST_ALPHA	(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)
6	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
7	ONE_MINUS_CONSTANT_ALPHA	(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)

Bit 19 – SPRE Source Premult

Value	Description
0	Source pre-multiplication is disabled.
1	Source pre-multiplication is enabled. Pixels from source surface are multiplied by constant before blending.

Bits 18:16 – SAFACT[2:0] Source Alpha Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_ALPHA	(A_s, A_s, A_s, A_s)

Value	Name	Description
3	ONE_MINUS_SRC_ALPHA	(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)
4	DST_ALPHA	(A_d, A_d, A_d, A_d)
5	ONE_MINUS_DST_ALPHA	(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)
6	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
7	ONE_MINUS_CONSTANT_ALPHA	(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)

Bits 15:12 – SPE[3:0] Special Blend

Value	Name	Description
0	LIGHTEN	Based on the color information in each channel, selects the base color or the blend color, whichever is lighter, as the resulting color.
1	DARKEN	Based on the color information in each channel, selects the base color or the blend color, whichever is darker, as the resulting color.
2	MULTIPLY	Based on the color information in each channel, multiplies the base color by the blend color. The result is always a darker color.
3	AVERAGE	Based on the color information in each channel, averages the base color and the blend color to generate the resulting color.
4	ADD	This blend mode adds pixel values.
5	SUBTRACT	Based on the color information in each channel, subtracts the blend color from the base color.
6	DIFFERENCE	Based on the color information in each channel, subtracts either the blend color from the base color or the base color from the blend color, depending on which has the greater brightness value.
7	NEGATION	This mode produces the opposite effect to difference.
8	SCREEN	Based on each channel's color information, multiplies the inverse of the blend and base colors. The resulting color is always a lighter color.
9	OVERLAY	Multiplies or screens the colors, depending on the base color.
10	DODGE	Based on the color information in each channel, brightens the base color to reflect the blend color by decreasing the contrast between the two.
11	BURN	Based on the color information in each channel, darkens the base color to reflect the blend color by increasing the contrast between the two.
12	REFLECT	This blend mode can be used to add shiny objects or area if light. Black pixels in the blend are ignored as if they were transparent.
13	GLOW	The reverse of the REFLECT mode. GLOW effectively brightens the composition by the amount of brightness in the blend layer. Black pixels in the blend layer are rendered as if they were transparent.

Bits 10:8 – FUNC[2:0] Blending Function

Value	Name	Description
0	ADD	C_f=S*C_s +D*C_d
1	SUBTRACT	C_f=S*C_s -D*C_d
2	REVERSE	C_f=D*C_d +S*C_s
3	MIN	C_f=Min(C_s, C_d)
4	MAX	C_f=Max(C_s, C_d)
5	SPE	Special blending functions

Bits 7:4 – DCFACT[3:0] Destination Color Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_COLOR	(A_s, R_s, G_s, B_s)
3	ONE_MINUS_SRC_COLOR	(1, 1, 1, 1)-(A_s, R_s, G_s, B_s)
4	DST_COLOR	(A_d, R_d, G_d, B_d)
5	ONE_MINUS_DST_COLOR	(1, 1, 1, 1)-(A_d, R_d, G_d, B_d)
6	SRC_ALPHA	(A_s, A_s, A_s, A_s)
7	ONE_MINUS_SRC_ALPHA	(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)

Value	Name	Description
8	DST_ALPHA	(A_d, A_d, A_d, A_d)
9	ONE_MINUS_DST_ALPHA	(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)
10	CONSTANT_COLOR	(A_c, R_c, G_c, B_c)
11	ONE_MINUS_CONSTANT_COLOR	(1, 1, 1, 1)-(A_c, R_c, G_c, B_c)
12	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
13	ONE_MINUS_CONSTANT_ALPHA	(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)
14	SRC_ALPHA_SATURATE	(1, i, i, i) where i is equal to the minimum between A_s and 1-A_d

Bits 3:0 – SCFACT[3:0] Source Color Factor

Value	Name	Description
0	ZERO	(0, 0, 0, 0)
1	ONE	(1, 1, 1, 1)
2	SRC_COLOR	(A_s, R_s, G_s, B_s)
3	ONE_MINUS_SRC_COLOR	(1, 1, 1, 1)-(A_s, R_s, G_s, B_s)
4	DST_COLOR	(A_d, R_d, G_d, B_d)
5	ONE_MINUS_DST_COLOR	(1, 1, 1, 1)-(A_d, R_d, G_d, B_d)
6	SRC_ALPHA	(A_s, A_s, A_s, A_s)
7	ONE_MINUS_SRC_ALPHA	(1, 1, 1, 1)-(A_s, A_s, A_s, A_s)
8	DST_ALPHA	(A_d, A_d, A_d, A_d)
9	ONE_MINUS_DST_ALPHA	(1, 1, 1, 1)-(A_d, A_d, A_d, A_d)
10	CONSTANT_COLOR	(A_c, R_c, G_c, B_c)
11	ONE_MINUS_CONSTANT_COLOR	(1, 1, 1, 1)-(A_c, R_c, G_c, B_c)
12	CONSTANT_ALPHA	(A_c, A_c, A_c, A_c)
13	ONE_MINUS_CONSTANT_ALPHA	(1, 1, 1, 1)-(A_c, A_c, A_c, A_c)
14	SRC_ALPHA_SATURATE	(1, i, i, i) where i is equal to the minimum between A_s and 1-A_d

46.4.6.4.ROP Instruction

The instruction length is 7 words.

Figure 46.8. ROP Operations

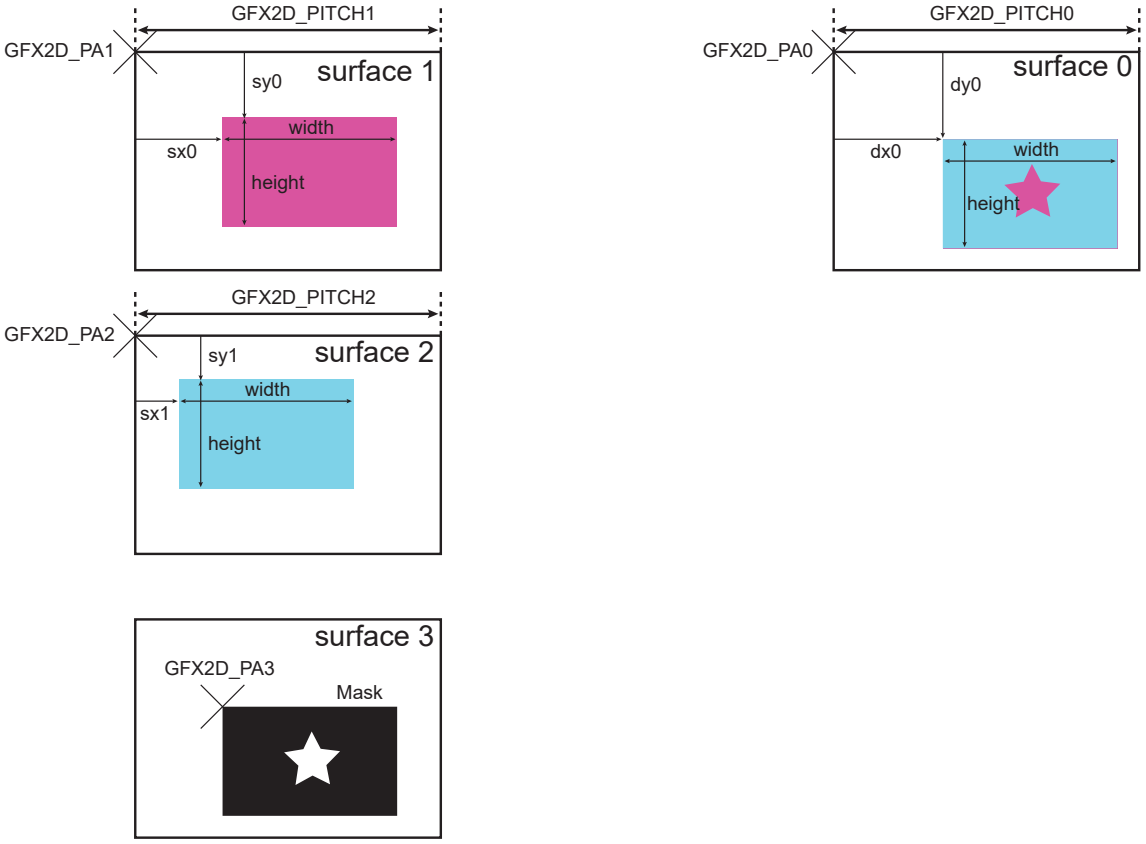


Table 46.31. ROP Instruction, Classic Operations

Word Identifier	Description	Surface Register
ROP_WD0	The instruction header	Not applicable
ROP_WD1	The width and height of the destination area D	Surface 0 registers
ROP_WD2	The x and y offsets of the destination area D	Surface 0 registers
ROP_WD3	The x and y offsets of source S	Surface 1 registers
ROP_WD4	The x and y offsets of pattern P	Surface 2 registers
ROP_WD5	A pointer to raster mask M	Surface 3 registers
ROP_WD6	Indicates the Raster Operation mode	Not applicable

46.4.6.4.1.ROP_WD0

Name: ROP_WD0

Bit	31	30	29	28	27	26	25	24
	OPCODE[3:0]							IE
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REG[3:0]			
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 31:28 – OPCODE[3:0] Instruction Code
This field must be set to '0xE'.

Bit 24 – IE Interrupt Enable

Value	Description
0	The End of Instruction interrupt is disabled.
1	The End of Instruction interrupt is enabled.

Bits 19:16 – REG[3:0] Register Identifier
When a constant is used, the constant value is located in the register defined by the REG value.

Bits 2:0 – ARGS[2:0] Arguments Length
Must be set to '5'.

46.4.6.4.2.ROP_WD1

Name: ROP_WD1

Bit	31	30	29	28	27	26	25	24
				HEIGHT[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	HEIGHT[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				WIDTH[12:8]				
Access								
Reset								

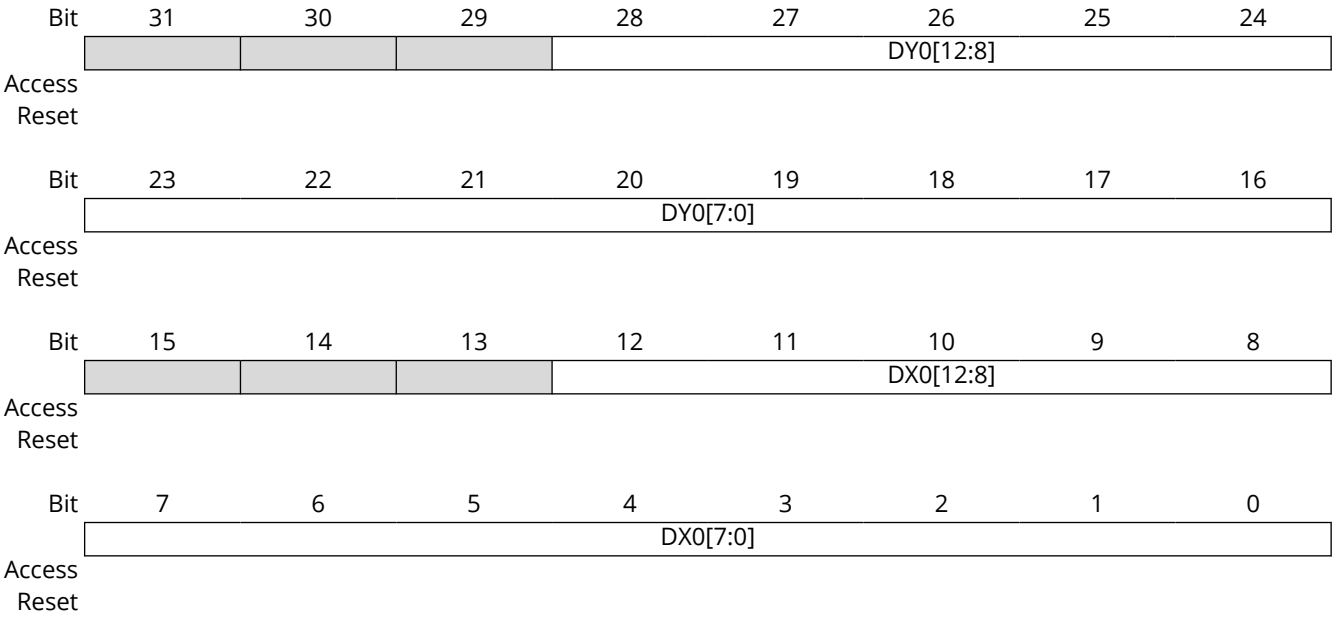
Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access								
Reset								

Bits 28:16 – HEIGHT[12:0] Destination Surface Height
The surface height is set to (HEIGHT+1) pixels.

Bits 12:0 – WIDTH[12:0] Destination Surface Width
The surface width is set to (WIDTH+1) pixels.

46.4.6.4.3.ROP_WD2

Name: ROP_WD2



Bits 28:16 – DY0[12:0] Destination Y Position
This field indicates the vertical position of the window.

Bits 12:0 – DX0[12:0] Destination X Position
This field indicates the horizontal position of the window.

46.4.6.4.4.ROP_WD3

Name: ROP_WD3

Bit	31	30	29	28	27	26	25	24
				SY0[12:8]				
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SY0[7:0]							
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
				SX0[12:8]				
Access								
Reset								

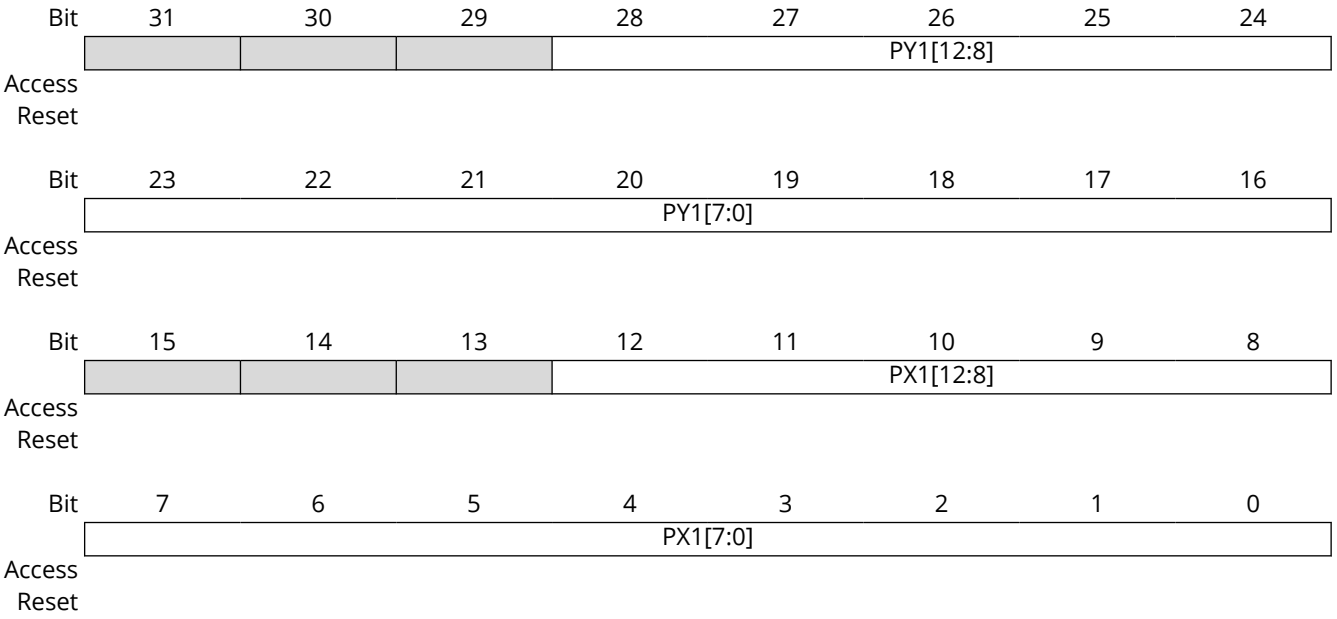
Bit	7	6	5	4	3	2	1	0
	SX0[7:0]							
Access								
Reset								

Bits 28:16 – SY0[12:0] Source Y Position
This field indicates the vertical position of the window.

Bits 12:0 – SX0[12:0] Source X Position
This field indicates the horizontal position of the window.

46.4.6.4.5.ROP_WD4

Name: ROP_WD4



Bits 28:16 – PY1[12:0] Pattern Y Position
This field indicates the pattern Y position of the window.

Bits 12:0 – PX1[12:0] Pattern X Position
This field indicates the pattern X position of the window.

46.4.6.4.6.ROP_WD5

Name: ROP_WD5

Bit	31	30	29	28	27	26	25	24
	PMASK[31:24]							
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	PMASK[23:16]							
Access								
Reset								

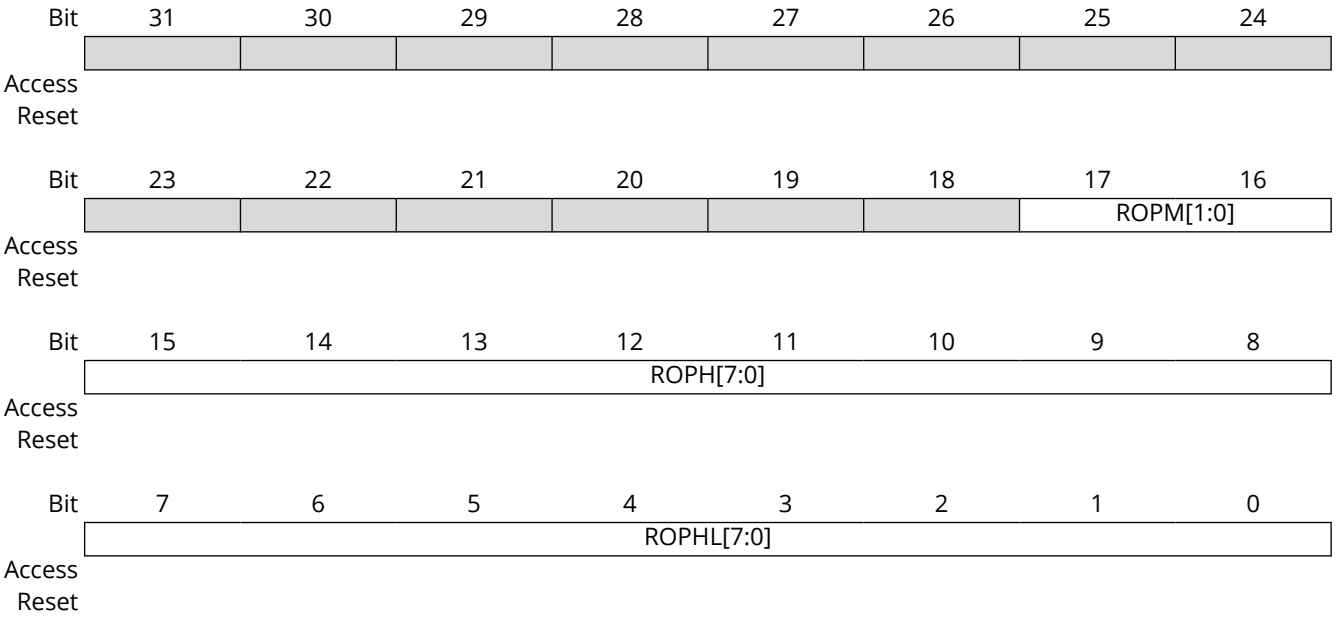
Bit	15	14	13	12	11	10	9	8
	PMASK[15:8]							
Access								
Reset								

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

Bits 31:0 – PMASK[31:0] Mask Pointer
This field must be set to the 32-bit address of the mask memory location.

46.4.6.4.7.ROP_WD6

Name: ROP_WD6



Bits 17:16 – ROPM[1:0] Raster Operation Mode

Value	Name	Description
0	ROP2	Binary raster operation (D, S) Two surfaces are used (required), 16 operations are defined, D and S required. Mask set to 0 internally Pattern set to 0 internally Source read from memory ROPL field value is used to generate the logical expression. ROPL can be any value between 0 and 15 (0x00 to 0x0F). ROPH field is ignored. ROP2_BLACK D = 0 (always 0, black in RGB mode) ROP2_NOTMERGESOURCE D = ~(D|S) Inverse of R2_MERGESOURCE ROP2_MASKNOTSOURCE D = D &~S Conjunction of destination and inverse of source ROP2_NOTCOPYSOURCE D = ~S Inverse of source color ROP2_MASKSOURCENOT D = S & ~D Conjunction of source and inverse of destination ROP2_NOT D = ~D Inverse of destination ROP2_XORSOURCE D = D ^ S Exclusive OR of destination and source ROP2_NOTMASKSOURCE D = ~(D&S) Inverse of ROP2_MASKSOURCE ROP2_MASKSOURCE D = D & P Conjunction of destination and source ROP2_NOTXORSOURCE D = ~(D ^ S) Inverse of ROP2_XORSOURCE ROP2_NOP D = D No change (copy) ROP2_MERGENOTSOURCE D = D | ~S Disjunction of destination and inverse of source ROP2_COPYSOURCE D = S ROP2_MERGESOURCENOT D = S | ~D Disjunction of source and inverse of destination ROP2_MERGESOURCE D = S | D Conjunction of source and destination

Value	Name	Description
		ROP2_WHITE D = 1
1	ROP3	Ternary raster operation (D, P, S) Three surfaces used, 256 operations Mask value set to 0 Source read from memory Pattern read from memory ROPL used to generate the logical expression 256 raster operations from 0x00 to 0xFF
2	ROP4	4-operand raster operation (D, P, S, M) Four surfaces used, 256 operations Mask value is read from memory (pointed by PMASK), mask is bit level. Source read from memory Pattern read from memory ROPL used to generate the logical expression when mask bit is cleared ROPH used to generate the logical expression when mask bit is set

Bits 15:8 – ROPH[7:0] 8-bit Raster Operation High

Bits 7:0 – ROPHL[7:0] 8-bit Raster Operation Low

46.5. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	GFX2D_GC	31:24								
		23:16					REGQOS3[3:0]			
		15:8	REGQOS2[3:0]				REGQOS1[3:0]			
		7:0		MTY		REGEN				
0x04	GFX2D_GE	31:24								
		23:16								
		15:8								
		7:0								ENABLE
0x08	GFX2D_GD	31:24								
		23:16								
		15:8								WFERES
		7:0								DISABLE
0x0C	GFX2D_GS	31:24								
		23:16								
		15:8								WFEIP
		7:0				BUSY				STATUS
0x10	GFX2D_IE	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x14	GFX2D_ID	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x18	GFX2D_IM	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x1C	GFX2D_IS	31:24								
		23:16								
		15:8								
		7:0				IERR	BERR	RERR	EXEND	RBEMPTY
0x20	GFX2D_PC0	31:24								
		23:16								
		15:8								
		7:0		FILT[2:0]					SEL[1:0]	
0x24	GFX2D_MC0	31:24	COUNTER[31:24]							
		23:16	COUNTER[23:16]							
		15:8	COUNTER[15:8]							
		7:0	COUNTER[7:0]							
0x28	GFX2D_PC1	31:24								
		23:16								
		15:8								
		7:0		FILT[2:0]					SEL[1:0]	
0x2C	GFX2D_MC1	31:24	COUNTER[31:24]							
		23:16	COUNTER[23:16]							
		15:8	COUNTER[15:8]							
		7:0	COUNTER[7:0]							
0x30	GFX2D_BASE	31:24	BASE[23:16]							
		23:16	BASE[15:8]							
		15:8	BASE[7:0]							
		7:0								
0x34	GFX2D_LEN	31:24								
		23:16								
		15:8								
		7:0					LEN[3:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	GFX2D_HEAD	31:24								
		23:16								
		15:8							HEAD[9:8]	
		7:0	HEAD[7:0]							
0x3C	GFX2D_TAIL	31:24								
		23:16								
		15:8							TAIL[9:8]	
		7:0	TAIL[7:0]							
0x40	GFX2D_PA0	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x44	GFX2D_PITCH0	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x48	GFX2D_CFG0	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x4C ... 0x4F	Reserved									
0x50	GFX2D_PA1	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x54	GFX2D_PITCH1	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x58	GFX2D_CFG1	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x5C ... 0x5F	Reserved									
0x60	GFX2D_PA2	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							
0x64	GFX2D_PITCH2	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x68	GFX2D_CFG2	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			
0x6C ... 0x6F	Reserved									
0x70	GFX2D_PA3	31:24	PA[31:24]							
		23:16	PA[23:16]							
		15:8	PA[15:8]							
		7:0	PA[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x74	GFX2D_PITCH3	31:24								
		23:16								
		15:8	PITCH[15:8]							
		7:0	PITCH[7:0]							
0x78	GFX2D_CFG3	31:24								
		23:16								
		15:8								
		7:0				IDXCX	PF[3:0]			

46.5.1. GFX2D Global Configuration Register

Name: GFX2D_GC
Offset: 0x00
Reset: 0x00065400
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					REGQOS3[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	1	1	0

Bit	15	14	13	12	11	10	9	8
	REGQOS2[3:0]				REGQOS1[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	0	1	0	1	0	0

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

Bits 19:16 – REGQOS3[3:0] Regulation for QoS Level 3
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS3}} - 1$$

The maximum number of clock cycles is 1023.

Bits 15:12 – REGQOS2[3:0] Regulation for QoS Level 2
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS2}} - 1$$

The maximum number of clock cycles is 1023.

Bits 11:8 – REGQOS1[3:0] Regulation for QoS Level 1
This register indicates the number of clock cycles inserted between outstanding transactions. The number of clock cycles added is calculated as follows.

$$\text{Latency} = 2^{\text{REGQOS1}} - 1$$

The maximum number of clock cycles is 1023.

Bit 6 – MTY Memory Tile Access

Value	Description
0	GFX2D uses tile accesses.
1	GFX2D uses linear accesses.

Bit 4 – REGEN Outstanding Regulation Enable

Value	Description
0	Outstanding Regulation is disabled.
1	Outstanding Regulation is enabled.

46.5.2. GFX2D Global Enable Register

Name: GFX2D_GE
Offset: 0x04
Reset: 0x00000000
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
								ENABLE
Access								W
Reset								0

Bit 0 – ENABLE GFX2D Enable

Value	Description
0	No effect.
1	Enables the GFX2D controller. This operation is permitted if the global status bit was read as ‘0’.

46.5.3. GFX2D Global Disable Register

Name: GFX2D_GD
Offset: 0x08
Reset: 0x00000000
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
								WFERES
Access								W
Reset								0

Bit	7	6	5	4	3	2	1	0
								DISABLE
Access								W
Reset								0

Bit 8 – WFERES WFE Software Resume bit

Value	Description
0	No effect.
1	Exits the WFE state in software.

Bit 0 – DISABLE GFX2D Disable Bit

Value	Description
0	No effect.
1	Disables the graphics engine.

46.5.4. GFX2D Global Status Register

Name: GFX2D_GS
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
								WFEIP
Access								R
Reset								0

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

Bit 8 – WFEIP Wait For Event Status bit

Value	Description
0	The graphics core is running.
1	The graphics core is waiting for an event.

Bit 4 – BUSY GFX2D Busy Bit

Value	Description
0	The graphics core is in idle state.
1	The graphics core is busy.

Bit 0 – STATUS GFX2D Status Bit

Value	Description
0	The graphics engine is disabled.
1	The graphics engine is enabled.

46.5.5. GFX2D Interrupt Enable Register

Name: GFX2D_IE
Offset: 0x10
Reset: 0x00000000
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
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Enable Bit

Bit 3 – BERR Write Data Bus Error Interrupt Enable Bit

Bit 2 – RERR Read Data Bus Error Interrupt Enable Bit

Bit 1 – EXEND End of Execution Interrupt Enable Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Enable Bit

46.5.6. GFX2D Interrupt Disable Register

Name: GFX2D_ID
Offset: 0x14
Reset: 0x00000000
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
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Disable bit

Bit 3 – BERR Write Access Error Interrupt Disable bit

Bit 2 – RERR Read Access Error Interrupt Disable Bit

Bit 1 – EXEND End of Execution Interrupt Disable Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Disable Bit

46.5.7. GFX2D Interrupt Mask Register

Name: GFX2D_IM
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 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
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Mask Bit

Bit 3 – BERR Write Error Interrupt Mask Bit

Bit 2 – RERR Read Error Interrupt Mask Bit

Bit 1 – EXEND Execution Ended Empty Interrupt Mask Bit

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Mask Bit

46.5.8. GFX2D Interrupt Status Register

Name: GFX2D_IS
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
				IERR	BERR	RERR	EXEND	RBEMPTY
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – IERR Illegal Instruction Interrupt Status Bit

Value	Description
0	The interrupt source is masked or no Illegal Instruction interrupt is pending.
1	An Illegal Instruction interrupt is pending.

Bit 3 – BERR Write Error Interrupt Status Bit

Value	Description
0	Either the interrupt source is masked or no Write Error interrupt is pending.
1	A Write Error interrupt is pending.

Bit 2 – RERR Read Error Interrupt Status Bit

Value	Description
0	Either the interrupt source is masked or no read error interrupt is pending.
1	An interrupt is pending.

Bit 1 – EXEND End of Execution Status Bit

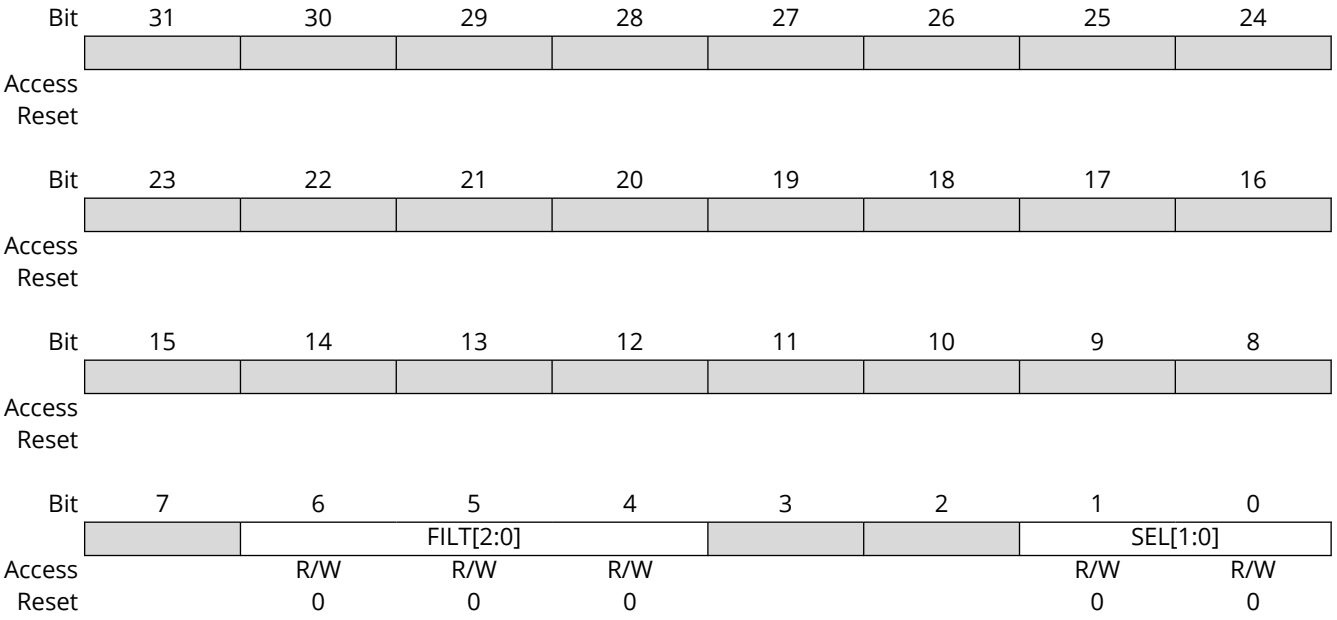
Value	Description
0	Either the interrupt source is masked or no End of Execution interrupt is pending.
1	An End of Execution interrupt is pending (i.e. EXEND is '1' when GFX2D_GS.BUSY is set to '0').

Bit 0 – RBEMPTY Ring Buffer Empty Interrupt Status Bit

Value	Description
0	The Ring Buffer Empty interrupt has not occurred.
1	The Ring Buffer Empty interrupt has occurred since the last read of the status register.

46.5.9. GFX2D Performance Configuration 0 Register

Name: GFX2D_PC0
Offset: 0x20
Reset: 0x00000000
Property: Read/Write



Bits 6:4 – FILT[2:0] Filter Configuration

Value	Name	Description
0	DISABLED	The filter is disabled.
1	QOS0	Events are valid when input QoS is equal to 0.
2	QOS1	Events are valid when input QoS is equal to 1.
3	QOS2	Events are valid when input QoS is equal to 2.
4	QOS3	Events are valid when input QoS is equal to 3.

Bits 1:0 – SEL[1:0] Performance Metrics Selection

Value	Name	Description
0	DISABLED	The performance counter is disabled and reset.
1	READ	The performance counter is incremented when a Read access is performed.
2	WRITE	The performance counter is incremented when a Write access is performed
3	CYCLE	Number of clock cycles

46.5.10. GFX2D Metrics Counter 0 Register

Name: GFX2D_MC0
Offset: 0x24
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COUNTER[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
	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 31:0 – COUNTER[31:0] Metrics Counter

46.5.11. GFX2D Performance Configuration 1 Register

Name: GFX2D_PC1
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
		FILT[2:0]					SEL[1:0]	
Access		R/W	R/W	R/W			R/W	R/W
Reset		0	0	0			0	0

Bits 6:4 – FILT[2:0] Filter Configuration

Value	Name	Description
0	DISABLED	The filter is disabled.
1	QOS0	Events are valid when input QoS is equal to 0.
2	QOS1	Events are valid when input QoS is equal to 1.
3	QOS2	Events are valid when input QoS is equal to 2.
4	QOS3	Events are valid when input QoS is equal to 3.

Bits 1:0 – SEL[1:0] Performance Metrics Selection

Value	Name	Description
0	DISABLED	The performance counter is disabled and reset.
1	READ	The performance counter is incremented when a Read access is performed.
2	WRITE	The performance counter is incremented when a Write access is performed
3	CYCLE	Number of clock cycles

46.5.12. GFX2D Metrics Counter 1

Name: GFX2D_MC1
Offset: 0x2C
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	COUNTER[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
	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 31:0 – COUNTER[31:0] Metrics Counter

46.5.13. GFX2D Ring Buffer Base Register

Name: GFX2D_BASE
Offset: 0x30
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	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	23	22	21	20	19	18	17	16
	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	15	14	13	12	11	10	9	8
	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

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:8 – BASE[23:0] Ring Buffer Base Register
This field is programmed with the Ring Buffer base address and is aligned on the allocation unit size. Ring buffer allocation unit is 2^8=256 bytes. The base address is 256 bytes aligned.

46.5.14. GFX2D Ring Buffer Length Register

Name: GFX2D_LEN
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
Access								
Reset								

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

Bits 3:0 – LEN[3:0] Ring Buffer Length Multiplier
Program this field with the desired buffer length multiplier. Buffer length = 256*(LEN+1) bytes.

46.5.15. GFX2D Ring Buffer Head Register

Name: GFX2D_HEAD
Offset: 0x38
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
							HEAD[9:8]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:0 – HEAD[9:0] Ring Buffer Head Pointer
The head pointer is updated by software in Functional mode to indicate the number of words written to memory.

46.5.16. GFX2D Ring Buffer Tail Register

Name: GFX2D_TAIL
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
Access								
Reset								

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

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

Bits 9:0 – TAIL[9:0] Ring Buffer Tail Pointer
The TAIL pointer is updated by the graphics engine in Functional mode to indicate how many words have been consumed.

46.5.17. GFX2D Surface x Physical Address Register

Name: GFX2D_PAx
Offset: 0x40 + x*0x10 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	PA[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
	PA[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
	PA[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
	PA[7:0]							
Access	R/W	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 – PA[31:0] Surface Physical Start Address
This address must be aligned with the surface pixel format.

46.5.18. GFX2D Surface x Pitch Register

Name: GFX2D_PITCHx
Offset: 0x44 + x*0x10 [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
	PITCH[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
	PITCH[7:0]							
Access	R/W	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 – PITCH[15:0] Surface Pitch
This field indicates the surface pitch size in bytes.

46.5.19. GFX2D Surface x Configuration Register

Name: GFX2D_CFGx
Offset: 0x48 + x*0x10 [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
				IDXCX	PF[3:0]			
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – IDXCX Color Look-Up Table Selection

Value	Description
0	The indexed color is retrieved in Color Look-Up Table 0.
1	The indexed color is retrieved in Color Look-Up Table 1.

Bits 3:0 – PF[3:0] Pixel Format

Value	Name	Description
0	A4IDX4	4-bit indexed color, with 4-bit alpha value
1	A8	8 bits per pixel alpha, with user-defined constant color
2	IDX8	8-bit indexed color, uses the Color Look-Up Table to expand to true color
3	A8IDX8	8-bit indexed color, with 8-bit alpha value
4	RGB12	12 bits per pixel, 4 bits per color channel
5	ARGB16	16 bits per pixel with 4-bit width alpha value, and 4 bits per color channel
6	RGB15	15 bits per pixel, 5 bits per color channel
7	TRGB16	16 bits per pixel, 5 bits for the red and blue channels and 6 bits for the green channel
8	RGBT16	16 bits per pixel, with 1 bit for transparency and 5 bits for color channels
9	RGB16	16 bits per pixel, 5 bits for the red and blue channels and 6 bits for the green channel
10	RGB24	24 bits per pixel, 8 bits for alpha and color channels
11	ARGB32	32 bits per pixel, 8 bits for alpha and color channels
12	RGBA32	32 bits per pixel, 8 bits for alpha and color channels

47. Display Serial Interface (DSI)

47.1. Description

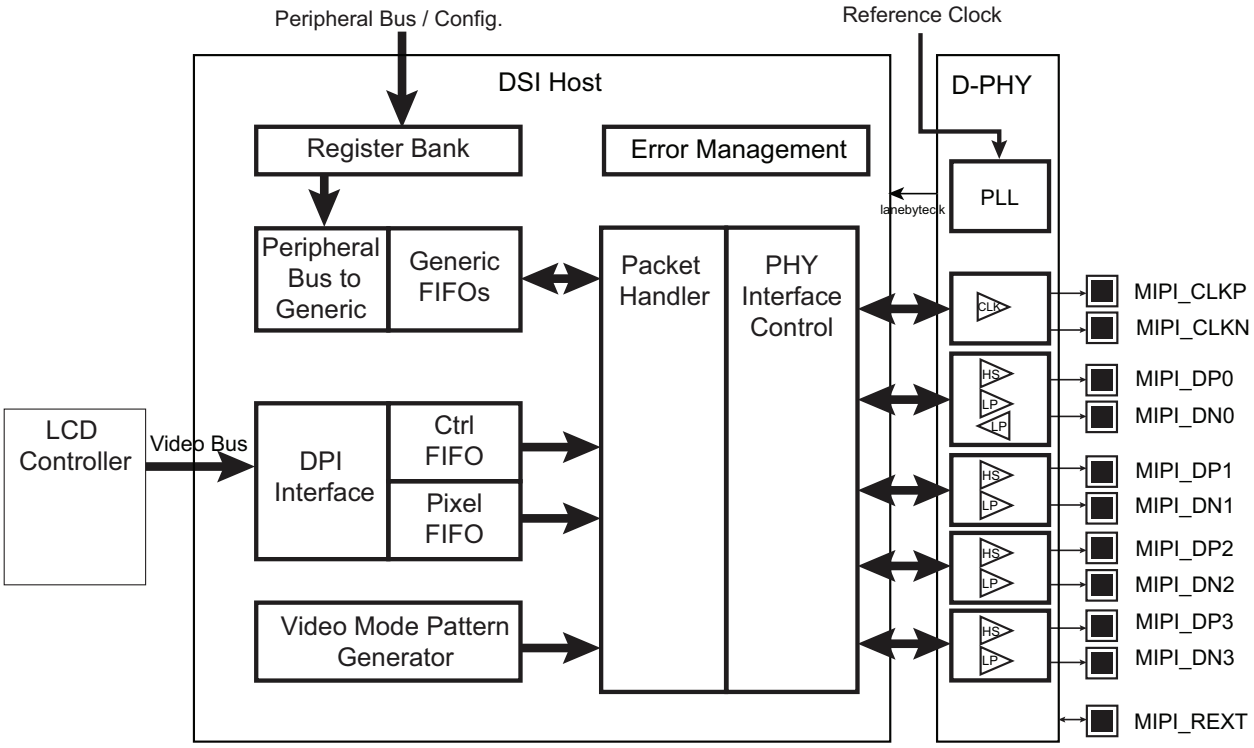
The MIPI Display Serial Interface (DSI) Host Controller is a digital core that implements all protocol functions defined in the MIPI DSI Specification. The DSI host provides an interface between the LCD Controller (LCDC) and the MIPI D-PHY, allowing communication with a DSI-compliant display.

47.2. Embedded Characteristics

- Supports All Commands Defined in the MIPI Alliance Specification for Display Command Set (DCS)
  - Transmission of all Command mode packets through the APB interface
  - Transmission of commands in Low-Power mode and High-Speed mode during Video mode
- Supports Up to 1 Gbps per Lane
- Supports Up to 4 Data Lanes
- Bidirectional Communication and Escape Mode Support Through Data Lane 0
- Supports Continuous and Non-continuous Clock in D-PHY Clock Lane for Additional Power Saving
- Supports Ultra-Low-Power Mode with PLL Disabled
- ECC and Checksum Capabilities
- Fault Recovery Schemes
- 3D Transmission Support
- Color Coding Mappings:
  - DPI-2 16-bit RGB, configurations 1, 2, and 3
  - DPI-2 18-bit RGB, configurations 1 and 2
  - DPI-2 24-bit RGB
- Video Pattern Generator
- Auto Ultra-Low-Power Mode Control Scheme

47.3. Block Diagram

Figure 47.1. DSI Block Diagram



47.4. I/O Lines Description

Table 47.1. I/O Lines Description

Name	Description	Type
MIPI_CLKP	MIPI D-PHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI D-PHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI D-PHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_DP2	MIPI D-PHY differential output data lane 2	Input/Output
MIPI_DN2		
MIPI_DP3	MIPI D-PHY differential output data lane 3	Input/Output
MIPI_DN3		
MIPI_REXT	Calibration reference resistor	Input/Output

47.5. Product Dependencies

47.5.1. I/O Lines

The D-PHY can be used either by DSI or CSI hosts. The programmer must first configure a dedicated a Special Function register to connect D-PHY to the DSI (refer to the section “Special Function Register (SFR)”).

47.5.2. Power Management

The DSI is not continuously clocked. Before using it, the programmer must first enable the DSI and MIPI D-PHY peripheral clocks in the Power Management Controller (PMC).

47.5.3. Interrupt Sources

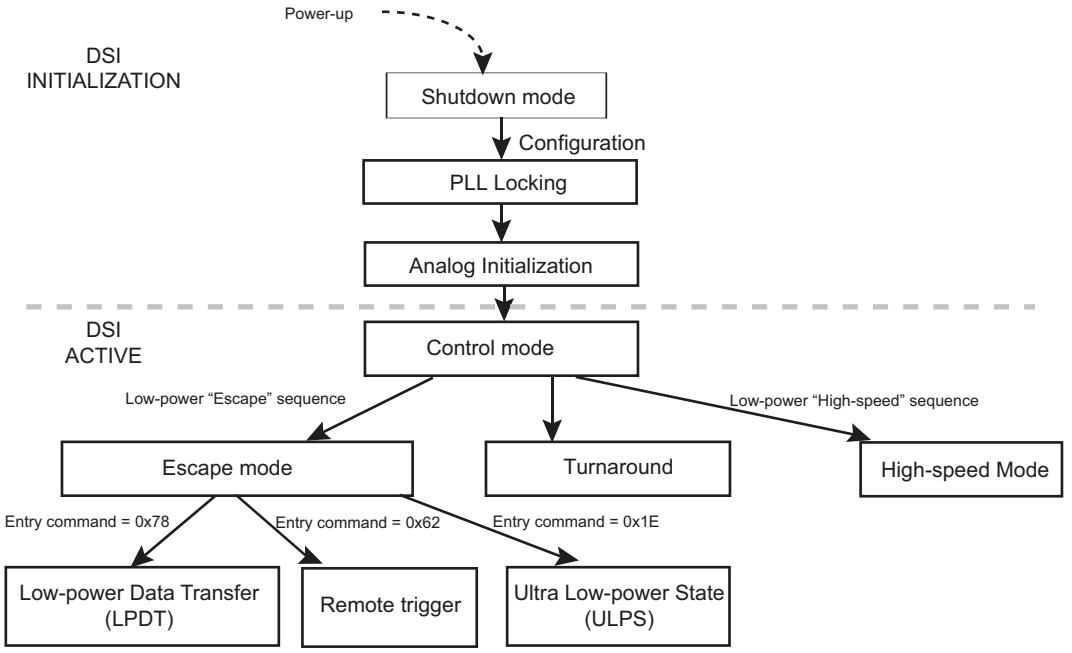
The DSI interrupt line is connected to one of the internal sources of the interrupt controller. Using the DSI interrupt requires prior programming of the interrupt controller.

47.6. Functional Description

47.6.1. D-PHY Operating Modes

The following figure illustrates the D-PHY initialization from Shutdown to Active modes.

Figure 47.2. D-PHY Initialization from Shutdown to Control Modes



47.6.1.1. Initialization

47.6.1.1.1. Shutdown Mode

Shutdown mode is the mode with the lowest power consumption, where all analog blocks are disabled, and digital logic is reset.

To enter this mode, set bits DPHY_RSTZ and SHUTDOWNZ in the Reset and PLL Control register (DSI_DPHY_RSTZ) to '0'.

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 operation range of 80-110 Mbps. If higher bit rate operation is required, the D-PHY hsfreqrange must be programmed as defined in Frequency Ranges. If the D-PHY is always expected to function at the same bit rate, this additional step can be performed while in Shutdown mode. For more information on these options, refer to Active Modes.

When DSI_DPHY_RSTZ.DPHY_RSTZ and DSI_DPHY_RSTZ.SHUTDOWNZ are set to '1', the D-PHY exits Shutdown mode and starts an initialization procedure.

47.6.1.1.2.PLL Locking Mode and Analog Initialization

The PLL is fed by the main crystal oscillator.

The D-PHY consists of 4 data lanes, but applications can use 4 or fewer data lanes. The number of lanes is configured in the field N_LANES in the D-PHY Interface Configuration register (DSI_DPHY_IF_CFG). Configuration of this field must be done prior to exiting Shutdown mode.

Prior to starting normal operation, the following D-PHY parameters must be configured:

- frequency range (refer to High-Speed Frequency Range Control Operation)
- PLL parameters (refer to D-PHY PLL Control Operation).

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.

47.6.1.2.Active Modes

47.6.1.2.1.Control Mode

Control mode is the default operating mode. After the initialization is completed (analog calibrations and PLL locking), the D-PHY remains in this default mode until a request is placed. The request is placed either by the protocol layer for transmission, or directly through the sequence of low-power signals in the lanes in case of reception. While in Control mode, the transmitter side sets the LP-11 state in the lines - this is called the Stop state. The receiver side remains in Control mode while receiving LP-11 in the lines. Any request must start from and end in Stop state.

Following a request, a lane can leave Control mode for either High-Speed Data Transfer mode, Escape mode, Ultra-Low-Power state, or turnaround operation.

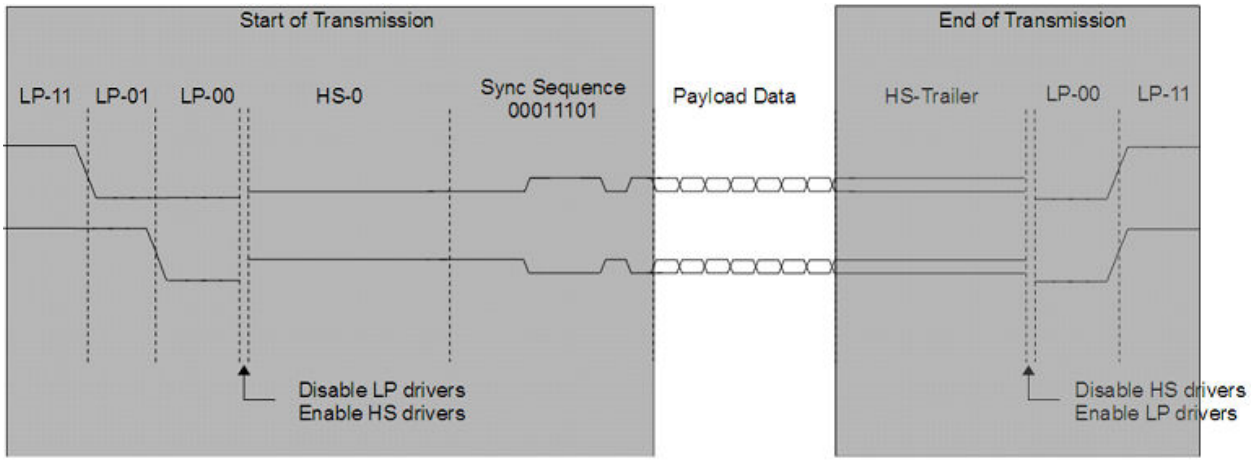
47.6.1.2.2.High-Speed Data Transfer Mode

High-speed data transfer occurs in bursts. The lane is in High-Speed mode only during these bursts. A high-speed burst must start from and return to a Stop state (Control mode).

High-speed data bursts are independent for each lane, which means that each data lane can start and end 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 also the end of transmission sequence.

Figure 47.3. HS Data Transfer Sequence



High-Speed mode request is initiated by the DSI host. This request is processed differently for clock and data lanes.

For a clock lane, the high-speed request is followed by the transmission of a low-power sequence that represents this request for the receiver side (a lane high-speed request). Only after generating this sequence, the low-power driver is disabled, and the high-speed driver enabled. After the time necessary to settle, the transmission of the high-speed DDR clock starts.

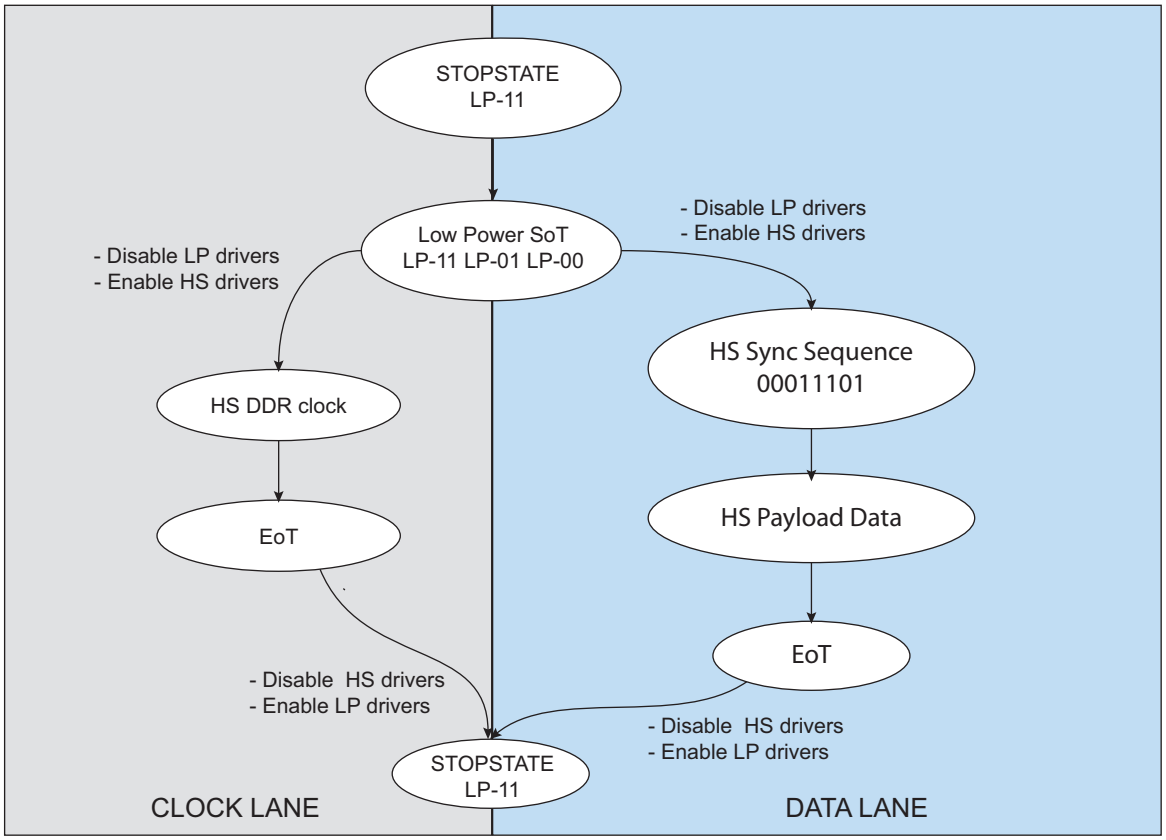
For a data lane, the high-speed request starts with a lane high-speed request and, in addition, extends the payload data with a leader and a trailer sequence that allows for the receiver synchronization. The transmission of such sequence requires the existence of a valid high-speed clock signal in the clock lane.

When the high-speed request is disabled, each lane exits High-Speed Data Transmission mode. It is important that a clock lane is in High-Speed mode during the complete high-speed data transmission state of all the lanes. The clock lane must enter High-Speed mode before a high-speed data transmission begins and it must not leave this state before all the lanes finish their respective high-speed data transmission bursts.

The operation sequence when exiting High-Speed mode is also different for data and clock lanes.

For a clock lane, high-speed transmission always ends with an HS-0 state, followed by the disabling of the high-speed driver, and enabling of the low-power driver. As for a data lane, the transmission ends with the differential state opposite to the last bit transmitted, followed by the disabling of the high-speed driver, and enabling of the low-power driver.

Figure 47.4. HS Data Transfer State Diagram



47.6.1.2.3.Escape Mode

Escape mode uses the data lanes to communicate asynchronously using the low-power states at low speed. The D-PHY supports this mode in both directions. 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.

If the entry command is not valid, it is ignored, the error flag ERRESC goes high (DSI_INT_ST0.DPHY_ERRORS_0), and the receiver waits until the transmitter returns to the Stop state.

The D-PHY applies Spaced-One-Hot encoding (a Mark state is interleaved with a Space state) on commands and data. Each symbol consists of the following two parts:

- One-Hot phase
- Space state

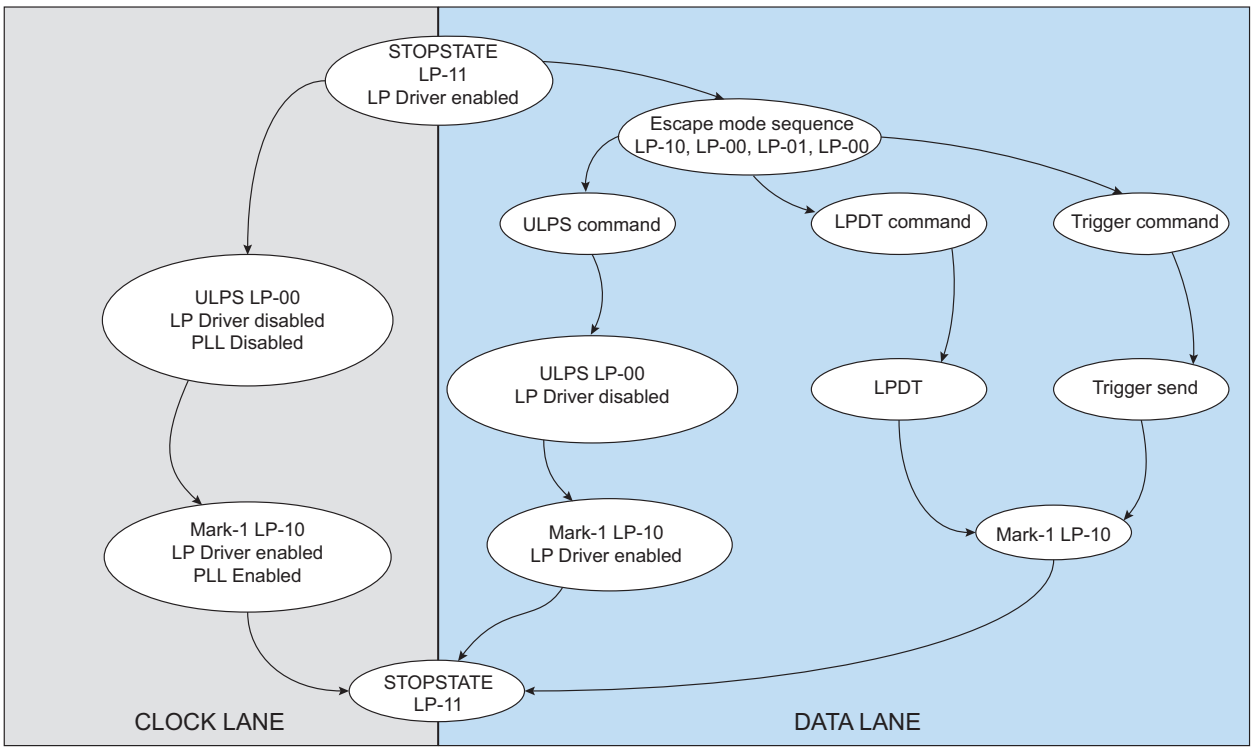
To transmit one bit, a Mark-1 should be sent followed by the Space state. In the case of a zero bit, a Mark-0 should be sent followed by the Space state.

Table 47.2. Possible Escape Mode Sequences for Data Lanes

Escape Mode Action	Entry Command Pattern (First Bit to Last Bit to be Transmitted)	Command Type
Low-Power Data Transmission	11100001	Mode
Ultra-Low-Power State	00011110	Mode
Reset Trigger	01100010	Trigger
Unknown - 3	01011101	Trigger
Unknown - 4	00100001	Trigger
Unknown - 5	10100000	Trigger

The Escape mode transition sequencing from state to state is presented in Figure 7-4.

Figure 47.5. Escape Mode Sequence State Diagram



- Low-Power Data Transmission (LPDT)

In LPDT mode, the data can be transmitted by the protocol at low speed in Low-Power mode. High-speed drivers or receivers are off and low-power drivers or 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.

- Ultra-Low-Power State (ULPS)

Ultra-Low-Power State (ULPS) mode has the lowest power consumption, excluding Shutdown mode.

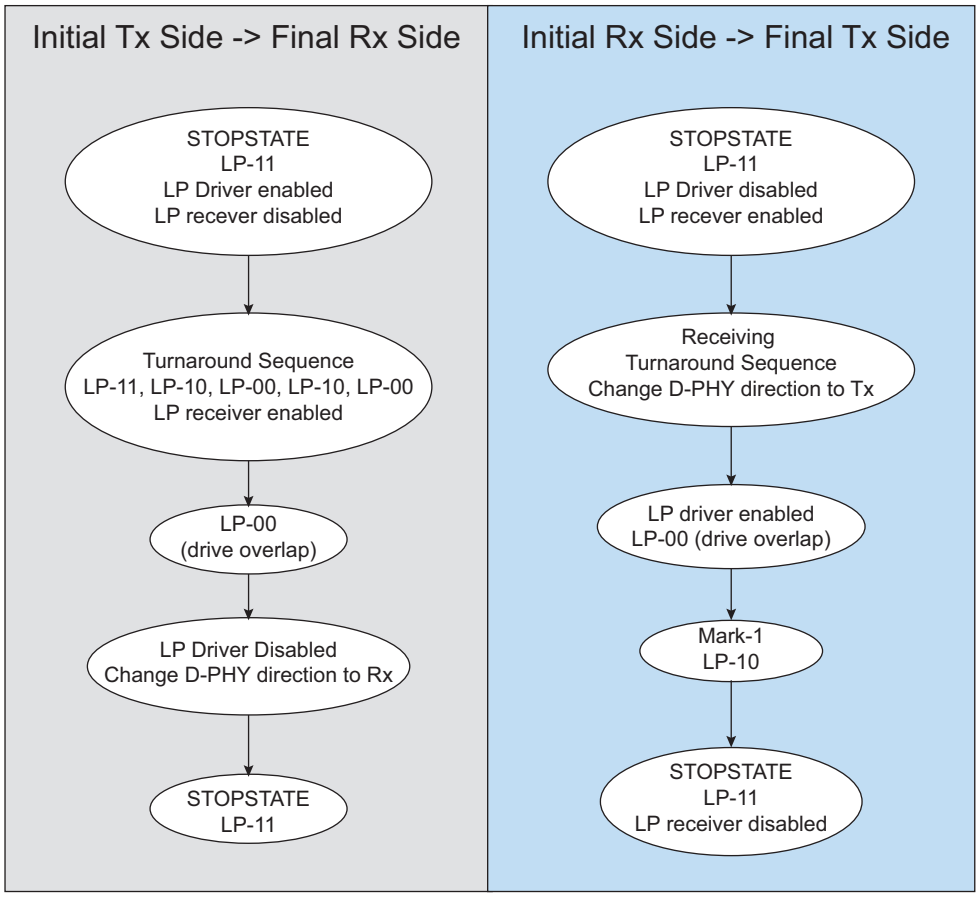
For data lanes, this mode is entered by sending an Ultra-Low-Power state entry command after the Escape mode entry command. During this mode, the lines are in the Space state (LP-00). Although the clock lane does not support regular Escape mode, the clock lane supports ULPS. If all lanes are set to ULPS, the PLL is turned off.

- Turnaround

The D-PHY allows the transmission direction of the data lanes to be swapped by means of a turnaround request. The sequence of a turnaround operation is described below:

1. Transmitter side
   a. Starts when the DSI host initiates a turnaround request, assuming the D-PHY is in Stop state
   b. The transmitter sends specific low-power sequence—LP-11, LP-10, LP-00, L-10, L-00 The turnaround operation can be disabled while in power down mode.
2. Receiver side
   a. After receiving a complete low-power sequence, the receiver enables the low-power driver.
   b. Transmits LP-00, indicating change.
   c. Overlapping LP-00 transmitted by the initial transmitter.
   d. Initial receiver side transmits LP-10, followed by LP-11, returning to Control mode.

Figure 47.6. Turnaround Sequence State Diagram



47.6.2. D-PHY Control Operation

47.6.2.1. General D-PHY Control Operation

The D-PHY is controlled by the DSI host user interface. For any test code and data, the control operation is carried out by following the steps below:

1. Ensure that the D-PHY is in Shutdown mode. See Shutdown Mode.
2. Reset the analog configuration by generating a high pulse on PHY_TESTCLR in the D-PHY Control register 0 (DSI_DPHY_TST_CTRL0).
3. Write a '1' to DSI_DPHY_TST_CTRL0.PHY_TESTCLK.
4. Write test code to PHY_TESTDIN and write a '1' to PHY_TESTEN in the D-PHY Control register 1 (DSI_DPHY_TST_CTRL1).
5. Write a '0' to DSI_DPHY_TST_CTRL0.PHY_TESTCLK.
6. Write a '0' to DSI_DPHY_TST_CTRL1.PHY_TESTEN and write the test data value to DSI_DPHY_TST_CTRL1.PHY_TESTDIN.
7. Write a '1' to DSI_DPHY_TST_CTRL0.PHY_TESTCLK.
8. Write a '0' to DSI_DPHY_TST_CTRL0.PHY_TESTCLK.

47.6.2.2. High-Speed Frequency Range Control Operation

The high-speed frequency range (hsfreqrange) is controlled by following the generic D-PHY Control operation (see General D-PHY Control Operation), and with the test code 0x44 and the data value from the following table:

Table 47.3. Frequency Ranges

Ranges (Mbps)	High-Speed Bit Rate Code
80-89	0000000
90-99	0100000
100-109	1000000
110-129	0000010
130-139	0100010
140-149	1000010
150-169	0000100
170-179	0100100
180-199	1000100
200-219	0000110
220-239	0100110
240-249	1000110
250-269	0001000
270-299	0101000
300-329	0001010
330-359	0101010
360-399	1001010
400-449	0001100
450-499	0101100
500-549	0001110
550-599	0101110
600-649	0010000
650-699	0110000
700-749	0010010
750-799	0110010
800-849	1010010
850-899	1110010
900-949	0010100
950-1000	0110100

47.6.2.3. D-PHY PLL Control Operation

The D-PHY includes a fully programmable PLL, enabling a flexible lane bit rate value. The generated clock frequency (f_{out}) is a function of the input reference frequency and of the multiplication and division ratios. It can be determined as follows:

$$f_{\text{out}} = \frac{M}{N} \cdot f_{\text{ref}}$$

Where:

- M = feedback multiplication ratio
- N = input frequency division ratio
- f_{ref} = frequency of the main crystal oscillator. f_{ref} = 24 MHz.

Note that the following limit applies:

$$40\text{MHz} \geq \frac{f_{\text{ref}}}{N} \geq 5\text{MHz}$$

The PLL output clock (with frequency f_{out}) is the full-rate clock used for bit serialization. A 1000 Mbps bit rate on the data lanes assumes PLL output frequency to be equal to 1000 MHz.

The D-PHY also generates the lanebyteclk clock signal with a frequency of $f_{out}/8$.

The PLL output frequency ranges are selected from the following table:

Table 47.4. PLL Output Frequency Ranges

Ranges (Mbps)	PLL Output Frequency Range (MHz)
000	80-200
001	200-300
010	300-500
011	500-700
100	700-900
101	900-1000

Some combinations of N and M are not allowed, since they violate the limits of operation of the PLL output frequency or the minimum allowed comparison frequency. Due to the use of a “by 2 pre-scaler” the range of the feedback multiplication value M is limited to even division numbers.

To ensure proper operation of the PLL, the loop bandwidth should be configured depending on the selected frequency. The control over the Charge Pump (CP) current (ICPCTRL[3:0]), the Low-Pass Filter (LPF) characteristics (LPFCTRL[5:0]), and VCO control signals (VCORANGE and VCOCAP) is granted. The table below presents the correspondence.

Table 47.5. PLL CP and LPF Control Fields

PLL Output Frequency Range (MHz)	VCORANGE	ICPCTRL	LPFCTRL	VCOCAP
80-110	000	0100	010000	00
110-150		1000	001000	00
150-200		1100	001000	00
200-250	001	1100	001000	00
250-300		0010	001000	00
300-400	010	0101	001000	00
400-500		0101	000001	00
500-600	011	0110	010000	00
600-700		0110	000100	00
700-900	100	0110	000100	00
900-1000	101	0111	010000	00

To configure D-PHY PLL, follow the steps below in conjunction with the procedure in General D-PHY Control Operation:

1. Configure the VCO parameters with the test code 0x10 and data as follows:
   - Bit 7:
     0: VCO range is programmed with the default values for the corresponding hsfreqrange
     1: VCO range is programmed with bits 5:3
   - Bit 6: Reserved
   - Bits 5:3: VCO range control (vcorange)
   - Bits 2:1: VCO internal capacitance control (vcocap)
     00: Default capacitance
     01: Low capacitance (four times lower than the default value)
     10: High capacitance (double the default value)

11: Not allowed

- Bit 0: Reserved

2. Configure PLL Control with test code 0x11 and data icpctrl.
3. Configure PLL Control with test code 0x12 and data as follows:
   - Bit 7: Bypass CP default values
     0: vcocap is programmed with the default values for the corresponding hsfreqrange
     1: vcocap is programmed with bits 3:0 using test code 0x11
   - Bit 6: Bypass LPF default values
     0: LPF is programmed with the default values for the corresponding hsfreqrange
     1: LPF is programmed with bits 5:0
   - Bit 5:0: Loop filter control (lpfctrl)
     000000: Loop Filter resistor is 18kΩ
     000001: Loop Filter resistor is 15.6kΩ
     000010: Loop Filter resistor is 15kΩ
     000100: Loop Filter resistor is 14.4kΩ
     001000: Loop Filter resistor is 12.8kΩ
     010000: Loop Filter resistor is 11.4kΩ
     100000: Loop Filter resistor is 10.5kΩ
4. Enable N and M values with test code 0x19 and data as follows:
   - Bits 7:6: Reserved
   - Bit 5: Bypass the PLL loop divider default values
     0: PLL loop divider is programmed with the default values for the corresponding hsfreqrange
     1: PLL loop divider is programmed using the test code 8'h18 (PLL Loop Divider Ratio)
   - Bit 4: Bypass PLL input divider default values
     0: PLL input divider is programmed with the default values for the corresponding hsfreqrange
     1: PLL input divider is programmed using test code 8'h17 (PLL Input Divider Ratio)
   - Bits 3:0: Reserved
5. Configure PLL Input Divider Ratio (N) with test code 0x17 and data assigned to N-1
6. Configure PLL Loop Divider Ratio (M) with test code 0x18 and data as follows:
   For m = M-1:
   - Data bit 7: Bit Field Selector
     - When data bit 7 is 0, LSBs are accessed:
       - data bits 6:5: Reserved
       - data bits 4:0: m[4:0]
     - When data bit 7 is 1 : MSBs are accessed:
       - data bits 6:4: Reserved
       - data bits 3:0: m[8:5]

47.6.3. Input Video Interface

47.6.3.1. Initialization

The Input Video interface captures the data and control signals from the LCD Controller (LCDC) and conveys them to the FIFO interfaces that transmit them to the DSI link. Two different streams of data are presented at the interface; video control signals and pixel data.

Depending on the interface color coding, the pixel data is disposed differently throughout the bus. Interface pixel color coding is shown in the table below:

Table 47.6. Location and Color Component In Input Video Interface

Order	16-bit Config 1	16-bit Config 2	16-bit Config 3	18-bit Config 1	18-bit Config 2	24-bit	DSC24
D23	–	–	–	–	–	R[7]	Byte1[7]
D22	–	–	–	–	–	R[6]	Byte1[6]
D21	–	–	R[4]	–	R[5]	R[5]	Byte1[5]
D20	–	R[4]	R[3]	–	R[4]	R[4]	Byte1[4]
D19	–	R[3]	R[2]	–	R[3]	R[3]	Byte1[3]
D18	–	R[2]	R[1]	–	R[2]	R[2]	Byte1[2]
D17	–	R[1]	R[0]	R[5]	R[1]	R[1]	Byte1[1]
D16	–	R[0]	–	R[4]	R[0]	R[0]	Byte1[0]
D15	R[4]	–	–	R[3]	–	G[7]	Byte2[7]
D14	R[3]	–	–	R[2]	–	G[6]	Byte2[6]
D13	R[2]	G[5]	G[5]	R[1]	G[5]	G[5]	Byte2[5]
D12	R[1]	G[4]	G[4]	R[0]	G[4]	G[4]	Byte2[4]
D11	R[0]	G[3]	G[3]	G[5]	G[3]	G[3]	Byte2[3]
D10	G[5]	G[2]	G[2]	G[4]	G[2]	G[2]	Byte2[2]
D9	G[4]	G[1]	G[1]	G[3]	G[1]	G[1]	Byte2[1]
D8	G[3]	G[0]	G[0]	G[2]	G[0]	G[0]	Byte2[0]
D7	G[2]	–	–	G[1]	–	B[7]	Byte3[7]
D6	G[1]	–	–	G[0]	–	B[6]	Byte3[6]
D5	G[0]	–	B[4]	B[5]	B[5]	B[5]	Byte3[5]
D4	B[4]	B[4]	B[3]	B[4]	B[4]	B[4]	Byte3[4]
D3	B[3]	B[3]	B[2]	B[3]	B[3]	B[3]	Byte3[3]
D2	B[2]	B[2]	B[1]	B[2]	B[2]	B[2]	Byte3[2]
D1	B[1]	B[1]	B[0]	B[1]	B[1]	B[1]	Byte3[1]
D0	B[0]	B[0]	–	B[0]	B[0]	B[0]	Byte3[0]

These configuration options are as follows:

- Polarity control: All the control signals are programmable to change the polarity depending on the LCDC configuration.
- After the controller reset, the input video interface waits for the first VSYNC active transition to start signal sampling, including pixel data, thus avoiding starting the transmission of the image data in the middle of a frame.
- If the interface pixel color coding is 18 bits and the 18-bit loosely packed stream is disabled, the number of pixels programmed in the field VID_PKT_SIZE in the Video Mode Packet Size Configuration register (DSI_VID_PKT_SIZE) must be a multiple of four. This means that in this mode, the two LSBs in the configuration are always inferred as zero. The specification states that in this mode, the pixel line size should be a multiple of four.
- To avoid FIFO underflows and overflows, the configured number of pixels is assumed to be received at all times. A set of bits in the DSI_VID_PKT_STATUS register report the status of the FIFOs and internal buffers associated with the input video interface support.
- To keep the memory organized with respect to the packet scheduling, the number of pixels per packet parameter is used to separate the memory space of different video packets.

For Shutdown and Color mode signal transmission, the input video signaling must be active. Because of such constraints and for commands to be correctly transmitted, the first VSYNC active pulse should occur for command sampling and transmission. When shutting down the display, the

input video must remain active for one frame after the command has been issued. This ensures that the commands are correctly transmitted before disabling the video generation at the input video interface.

47.6.3.2. Video Transmission Modes

The different video transmission modes are as follows:

- Burst mode
- Non-Burst mode
  - Non-Burst mode with sync pulse
  - Non-Burst mode with sync event

47.6.3.2.1. Burst Mode

In this mode, the entire active pixel line is buffered into a FIFO and transmitted in a single packet with no interruptions. This transmission mode requires that the Input Video Pixel FIFO has the capacity to store a full line of active pixel data. This mode is optimized if the difference between the pixel required bandwidth and DSI link bandwidth is very different. This enables the DSI host to quickly dispatch the entire active video line in a single burst of data and then return to Low-Power mode.

47.6.3.2.2. Non-Burst Mode

In this mode, the processor uses the partitioning properties of the DSI host to divide the video line transmission into several DSI packets. This is done to match the pixel required bandwidth with the DSI link bandwidth. With this mode, the controller configuration does not require a full line of pixel data to be stored inside the input video pixel FIFO. It requires only the content of one video packet.

47.6.3.3. Updating the Input Video Interface Configuration

It is possible to update the Input Video Interface configuration on the fly without impacting the current frame. Using the Video Mode Shadow Feature Control register (DSI_VID_SHADOW_CTRL). The new configuration is only used when the system requests it.

To update the input video interface configuration during the transmission of a video frame, the configuration of that frame must be stored in the auxiliary registers. Thus, the new frame configurations can be set through the user interface without corrupting the current frame.

By default, this feature is disabled. To enable this feature, set the bit DSI_VID_SHADOW_CTRL.VID_SHADOW_EN to ‘1’. When this feature is enabled, the system supplies the configuration stored in the auxiliary registers.

If the active registers must be updated immediately without the reset, ensure that the bits DSI_VID_SHADOW_CTRL.VID_SHADOW_EN and DSI_VID_SHADOW_CTRL.VID_SHADOW_REQ are set to 0.

47.6.4. Generic Interface

47.6.4.1. Description

The user interface allows the transmission of write and read command mode packets. Commands sent through this interface are not limited to the scope of the DCS specification, and can include generic commands described in the DSI specification as manufacturer-specific.

The Generic Interface Payload register (DSI_GEN_PLD_DATA) has two distinct functions based on the operation. Writing to this register sends the data as payload when sending a Command mode packet. Reading this register returns the payload of a read back operation. The Generic Interface Header register (DSI_GEN_HDR) contains the Command mode packet header type and header data. Writing to this register triggers the transmission of the packet implying that for a long Command mode packet, the packet payload must be written in advance in DSI_GEN_PLD_DATA.

The valid packets available to be transmitted through the generic interface are as follows:

- Generic Write Short Packet 0 Parameters
- Generic Write Short Packet 1 Parameters
- Generic Write Short Packet 2 Parameters
- Generic Read Short Packet 0 Parameters
- Generic Read Short Packet 1 Parameters
- Generic Read Short Packet 2 Parameters
- Maximum Read Packet Configuration
- Generic Long Write Packet
- DCS Write Short Packet 0 Parameters
- DCS Write Short Packet 1 Parameters
- DCS Read Short Packet 0 Parameters
- DCS Write Long Packet

A set of bits in the Command Packet Status register (DSI_CMD_PKT_STATUS) reports the status of the FIFOs associated with user interface support.

47.6.4.2. Packet Transmission Using the Generic Interface

While writing the packet information, first write the payload of a given packet into the payload FIFO using DSI_GEN_PLD_DATA. When the payload data is for the command parameters, place the first byte to be transmitted in the least significant byte position of the user interface data bus. For more information, see Generic Interface Payload Register.

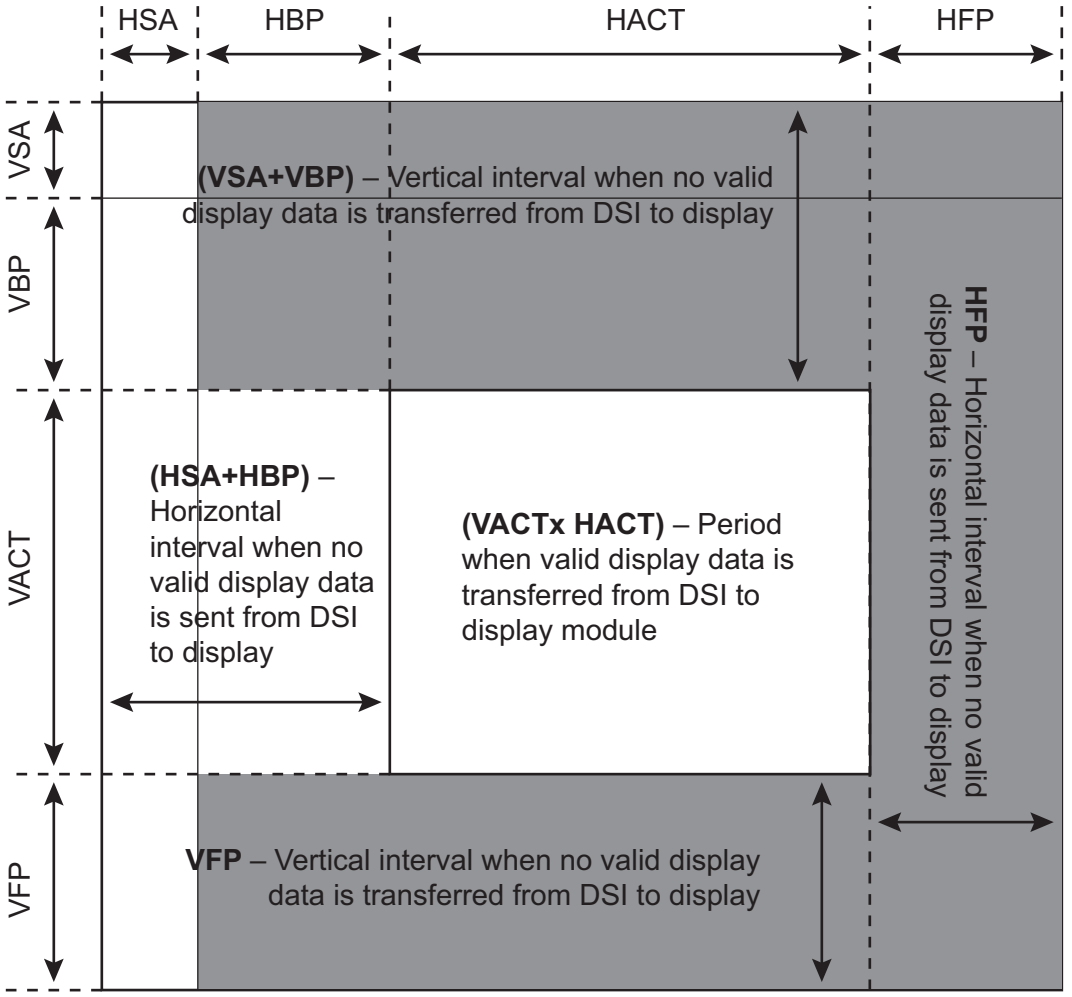
After writing the payload, write the packet header into the command FIFO. For more information about the packet header organization on the user interface, see Generic Interface Header Register.

47.6.5. Transmission of Commands

47.6.5.1. Transmission of Commands in Video Mode

The DSI host supports the transmission of commands, both in high-speed and low-power, while in Video mode. Vertical and horizontal porch periods are used to transmit commands inserted through the user interface. Those periods correspond to the shaded areas of the figure below:

Figure 47.7. Command Transmission Periods within the Image Area



Commands are transmitted in the blanking periods after the following packets/states:

- Vertical Sync Start (VSS) packets, if the video sync pulses are not enabled
- Horizontal Sync End (HSE) packets, in the VSA, VBP, and VFP regions
- Horizontal Sync Start (HSS) packets, if the video sync pulses are not enabled in the VSA, VBP, and VFP regions
- Horizontal Active (HACT) state

The DSI host does not send commands in the last line unless a command does not fit into any blanking period area. In this case, it is postponed to the last line, causing the violation of the line time for Video mode.

There can be only one command sent in Low-Power mode per line. However, one Low-Power mode command is possible for each line.

In high-speed, the DSI host can send more than one command and as many as the host determines can fit in the available time.

The Color Mode and Shutdown signals controlled by the LCDC are also able to trigger the transmission of command packets. For details, refer to the section LCD Controller (LCDC). The commands are DSI data types Color Mode On, Color Mode Off, Shut Down Peripheral, and Turn On Peripheral. These commands are not sent in the VACT region. If the bit LP_CMD_EN in the Video Mode Configuration register (DSI_VID_MODE_CFG) is 1, these commands are sent in LP mode. In

LP mode, the field OUTVACT_LPCMD_TIME in the Input Video Timing for Low-Power Commands Configuration register (DSI_DPI_LP_CMD_TIM) is used to determine if these commands can be transmitted. It is assumed that DSI_DPI_LP_CMD_TIM.OUTVACT_LPCMD_TIME is greater than or equal to four bytes (number of bytes in a short packet).

If DSI_VID_MODE_CFG.FRAME_BTA_ACK_EN is set to 1, a Bus Turn-Around (BTA) is generated by the DSI host after the last line of a frame. This may coincide with a write command or a read command.

47.6.5.2. Transmission of Commands in Low-Power

The DSI host can be configured to send the low-power (LP) commands during the HS video mode transmission. To enable this feature, set the bit DSI_VID_MODE_CFG.LP_CMD_EN to 1. In this case, it is necessary to ensure there is enough time available, in bytes, to transmit a command in LP mode to Horizontal Front Porch (HFP), Vertical Sync Active (VSA), Vertical Back Porch (VBP), and Vertical Front Porch (VFP) regions. Bits 8 to 13 of DSI_VID_MODE_CFG indicate whether the DSI host can go to low-power depending on the current image area.

If DSI_VID_MODE_CFG.LP_CMD_EN is set to 1 and non-video packets are in the queue, the DSI host ignores the Low-Power mode configuration and transmits Low-Power mode commands, even if it is not allowed to enter Low-Power mode in a specific region. After transmitting Low-Power mode commands, the DSI host remains in Low-Power mode until a sync event occurs.

47.6.5.3. Transmission of Commands in High-Speed

If DSI_VID_MODE_CFG.LP_CMD_EN is set to 0, the commands are sent in high-speed in Video mode. In this case, the DSI host automatically determines the area where each command can be sent and no programming or calculation is required.

47.6.5.4. Read Command Transmission

The DSI_PHY_TMR_CFG.MAX_RD_TIME configures the maximum amount of time required to perform a read command in lane byte clock cycles. It is calculated as follows:

MaxRdTime = Time to transmit the read command in LP mode + Time to enter and leave LP mode + Time to return the read data packet from the peripheral device.

The time to return the read data packet from the peripheral depends on the number of bytes read and the escape clock frequency of the peripheral. The field MAX_RD_TIME is used in both High-Speed and Low-Power modes to determine if there is time to complete a read command in a blanking period.

In High-Speed mode (DSI_VID_MODE_CFG.LP_CMD_EN = 0), MaxRdTime is calculated as follows:

$$\text{MaxRdTime} = \frac{t_{\text{HS}\rightarrow\text{LP}} + t_{\text{LP}\rightarrow\text{HS}} + t_{\text{read}} + 2t_{\text{BTA}}}{\text{lanebyteclkperiod}}$$

In Low-Power mode (DSI_VID_MODE_CFG.LP_CMD_EN = 1), MaxRdTime is calculated as follows:

$$\text{MaxRdTime} = \frac{t_{\text{HS}\rightarrow\text{LP}} + t_{\text{LP}\rightarrow\text{HS}} + t_{\text{lpdt}} + t_{\text{lprd}} + t_{\text{read}} + 2t_{\text{BTA}}}{\text{lanebyteclkperiod}}$$

Where:

- $t_{HS->LP}$ = Time to enter LP mode
- $t_{LP->HS}$ = Time to leave LP mode
- t_{LPDT} = D-PHY timing related to Escape mode entry, LPDT command, and Escape mode exit. This value is always 11 bits in Low-Power mode (or 22 TX escape clock cycles).
- t_{lprd} = Read command time in LP mode (64 * TX escape clock)
- t_{read} = Time to return the read data packet from the peripheral
- t_{BTA} = time to perform a bus turnaround (device D-PHY dependent)

It is recommended to keep the maximum number of bytes read from the peripheral to a minimum to have sufficient time available to issue the read commands in a line time.

47.6.5.5. Clock Lane in Low-Power Mode

To reduce the power consumption of the D-PHY, the DSI host, when not transmitting in High-Speed mode, allows the clock lane to enter Low-Power mode. The controller automatically handles the transition of the clock lane from High-Speed mode (Clock lane active sending clock) to Low-Power mode without direct intervention by the software. This feature can be enabled by configuring the bits PHY_TXREQUESTCLKHS and AUTO_CLKLANE_CTRL in the Clock Lane Control register (DSI_LPCLK_CTRL).

In Command mode, the DSI host can place the clock lane in Low-Power mode when it does not have any High-speed packets to transmit. In Video mode, the Low-Power mode controller uses its internal video and D-PHY timing configurations to determine if there is time available for the clock line to enter Low-Power mode and not compromise the video data transmission of pixel data and sync events.

Along with a correct configuration of Video mode, the DSI host needs to know the time required by the clock and data lanes to go from high-speed to low-power and from low-power to high-speed. The values required can be obtained from the following table. Program the Clock Lane Switch Mode Timing Configuration register (DSI_DPHY_TMR_LPCLK_CFG) and the Data Lane Switch Mode Timing Configuration register (DSI_DPHY_TMR_CFG) with the following values expressed in lane byte clock periods:

Table 47.7. High-Speed Transition Times (expressed in lane byte clock periods)

Frequency Range (MHz)	LP->HS Clock Lane	HS->LP Clock Lane	LP->HS Data Lane	HS->LP Data Lane
80-89	32	20	26	13
90-99	35	23	28	14
100-109	32	22	26	13
110-129	31	20	27	13
130-139	33	22	26	14
140-149	33	21	26	14
150-169	32	20	27	13
170-179	36	23	30	15
180-199	40	22	33	15
200-219	40	22	33	15
220-239	44	24	36	16
240-249	48	24	38	17
250-269	48	24	38	17
270-299	50	27	41	18
300-329	56	28	45	18
330-359	59	28	48	19
360-399	61	30	50	20
400-449	67	31	55	21
450-499	73	31	59	22
500-549	79	36	63	24
550-599	83	37	68	25
600-649	90	38	73	27
650-699	95	40	77	28
700-749	102	40	84	28
750-799	106	42	87	30
800-849	113	44	93	31
850-899	118	47	98	32

Table 47.7. High-Speed Transition Times (expressed in lane byte clock periods) (continued)

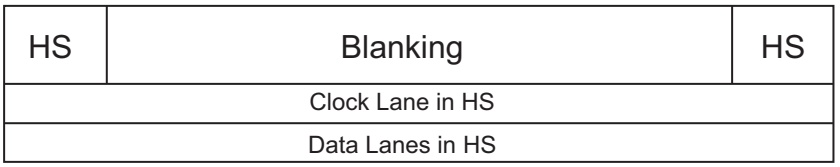
Frequency Range (MHz)	LP->HS Clock Lane	HS->LP Clock Lane	LP->HS Data Lane	HS->LP Data Lane
900-949	124	47	102	34
950-999	130	49	107	35

Based on the programmed values, the DSI host calculates if there is enough time for the clock lane to enter Low-Power mode during inactive regions of the video frame. There is an exception where the clock lane is activated even when there is no high-speed packet required to be transmitted. If a command is not allowed to be transmitted in any of the available blanking periods, it is transmitted during the last line of the frame.

The DSI host determines the best approach regarding power saving from among the following three possible cases:

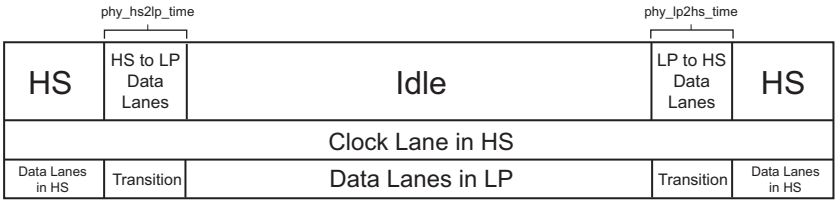
- There is not sufficient time to enter Low-Power mode. Therefore, a blanking period is added.

Figure 47.8. Clock Lane and Data Lanes in High-Speed mode (HS)



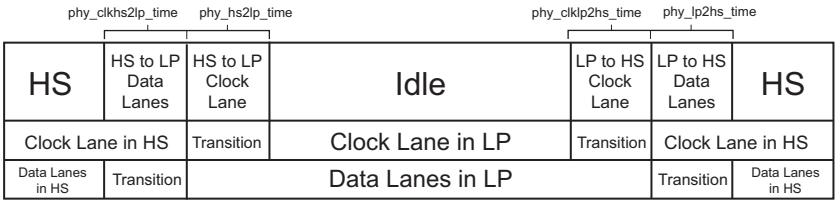
- There is sufficient time for the data lanes to enter Low-Power mode but not enough time for the clock lane to enter Low-Power mode.

Figure 47.9. Clock Lane in High-Speed (HS) and Data Lanes in Low-Power Mode (LP)



- There is sufficient time for both data lanes and the clock lane to enter Low-Power mode.

Figure 47.10. Clock Lane and Data Lanes in Low-Power Mode (LP)



47.6.6. Video Mode Pattern Generator

The Video mode pattern generator allows the transmission of horizontal/vertical color bars and D-PHY BER testing patterns without any stimuli. The frame requirements must be defined in the video registers listed below:

- DSI_VID_MODE_CFG
- DSI_VID_PKT_SIZE
- DSI_VID_NUM_CHUNKS
- DSI_VID_NULL_SIZE

- DSI_VID_HSA_TIME
- DSI_VID_HBP_TIME
- DSI_VID_HLINE_TIME
- DSI_VID_VSA_LINES
- DSI_VID_VBP_LINES
- DSI_VID_VFP_LINES
- DSI_VID_VACTIVE_LINES

Refer to the Register Summary for additional information.

47.6.6.1. Color Bar Pattern

The color bar pattern comprises eight bars of the colors white, yellow, cyan, green, magenta, red, blue, and black.

Each color width is calculated by dividing the line pixel size (vertical pattern) or the number of lines (horizontal pattern) by eight. In Vertical Color Bar mode, each single color bar has a width equal to the number of pixels in a line divided by eight. If the number of pixels in a line is not divisible by eight, the last color (black) contains the remaining pixels. In Horizontal Color Bar mode, each color line has a width equal to the number of lines in a frame divided by eight. If the number of lines in a frame is not divisible by eight, the last color (black) contains the remaining lines.

Figure 47.11. Vertical Color Bar Mode

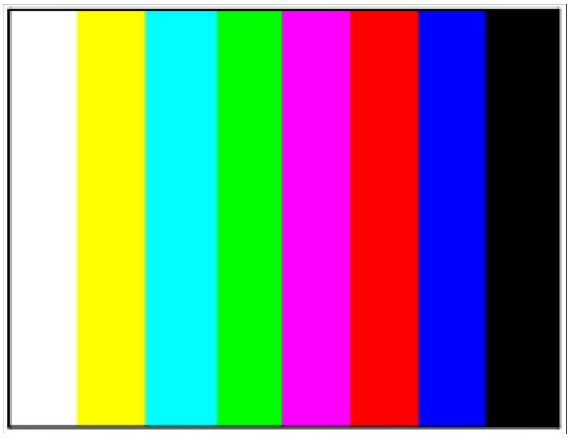
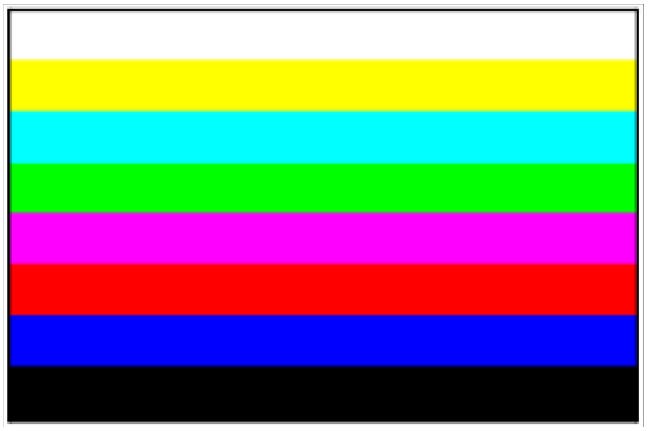


Figure 47.12. Horizontal Color Bar Mode



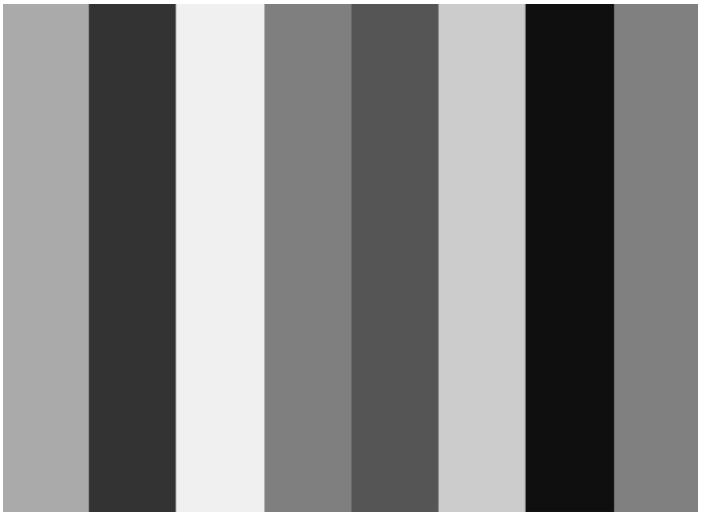
47.6.6.2. BER Testing Pattern

The BER testing pattern simplifies conformance testing. This pattern tests the RX D-PHY capability to receive the data correctly. The following data patterns are required:

- X bytes of 0xAA (high-frequency pattern, inverted)
- X bytes of 0x33 (mid-frequency pattern)
- X bytes of 0xF0 (low-frequency pattern, inverted)
- X bytes of 0x7F (lone 0 pattern)
- X bytes of 0x55 (high-frequency pattern)
- X bytes of 0xCC (mid-frequency pattern, inverted)
- X bytes of 0x0F (low-frequency pattern)
- Y bytes of 0x80 (lone 1 pattern)

In most cases, Y is equal to X. However, depending on line length and the color coding used, Y may be a different value than X. With RGB888 color coding and horizontal resolution in multiples of eight, the following pattern appears on the DSI display:

Figure 47.13. RGB888 BER Testing Pattern



Depending on the orientation, BER mode, and color coding, the smallest resolutions accepted by the video mode pattern generator are:

- BER mode: 8x8
- Horizontal color bar mode: 8x8
- Vertical color bar mode: 8x8

47.6.7. Error Handling

The interrupts triggered by the Interrupt Status registers 0 and 1 (DSI_INT_ST0 and DSI_INT_ST1) are error conditions.

The two following tables provide the causes that trigger these interrupts and also explain how to recover from them.

Table 47.8. Error Cause and Recovery for INT_ST0 Register

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
20	DPHY_ERRORS_4	The D-PHY reports the LP1 contention error. The D-PHY host detects the contention while trying to drive the line high.	Recover the D-PHYs from contention. Reset the DSI host and transmit the packets again. If this error is recurrent, analyze the connectivity between the host and the device carefully.
19	DPHY_ERRORS_3	D-PHY reports the LP0 contention error. The D-PHY host detects the contention while trying to drive the line low.	Recover the D-PHYs from contention. Reset the DSI host and transmit the packets again. If this error is recurrent, analyze the connectivity between the host and the device carefully.
18	DPHY_ERRORS_2	The D-PHY reports the False Control Error. The D-PHY detects an incorrect line state sequence in lane 0 lines.	Device does not behave as expected. Communication with the device is not properly established. This is an unrecoverable error. Reset the DSI host and the D-PHY. If this error is recurrent, analyze the behavior of the device.
17	DPHY_ERRORS_1	The D-PHY reports the LPDT Error. The D-PHY detects that the LDPT did not match a multiple of 8 bits.	The data reception is not reliable. The D-PHY recovers but the received data from the device might not be reliable. It is recommended to reset the DSI host and repeat the RX transmission.
16	DPHY_ERRORS_0	The D-PHY reports the Escape Entry Error. The D-PHY does not recognize the received Escape Entry Code.	The D-PHY does not recognize the Escape Entry Code. The Transmission is ignored. The D-PHY host recovers but the system should repeat the RX reception.
15	ACK_WITH_ERR_15	This error is directly retrieved from Acknowledge with Error packet. The device detected a protocol violation in the reception.	Refer to the display documentation. When this error is active, the device should have another read-back command that reports additional information about this error. Read the additional information and take appropriate actions.
14	ACK_WITH_ERR_14	The Acknowledge with Error packet contains this error. The device chooses to use this bit for error report.	Refer to the device documentation regarding possible reasons for this error and take appropriate actions.
13	ACK_WITH_ERR_13	The Acknowledge with Error packet contains this error. The device reports that the transmission length does not match the packet length.	Possible reason for this is multiple errors present in the packet header (more than 2), so the error detection fails and the device does not discard the packet. In this case, the packet header is corrupt and can cause decoding mismatches. Transmit the packets again. If this error is recurrent, analyze the connectivity between the host and the device carefully.
12	ACK_WITH_ERR_12	The Acknowledge with Error packet contains this error. The device does not recognize the VC ID in at least one of the received packets.	Check the device capabilities and configure the host to properly address the device VC ID. Repeat the transmission.
11	ACK_WITH_ERR_11	The Acknowledge with Error packet contains this error. The device does not recognize the data type of at least one of the received packets.	Check the device capabilities. It is possible that there are some packets that are not supported by the device. Repeat the transmission.
10	ACK_WITH_ERR_10	The Acknowledge with Error packet contains this error. The device detects the CRC errors in at least one of the received packets.	Some of the long packets, transmitted after the last Acknowledge request, might contain the CRC errors in the payload. If the payload content is critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the host and the device carefully.
9	ACK_WITH_ERR_9	The Acknowledge with Error packet contains this error. The device detects multi-bit ECC errors in at least one of the received packets.	The device does not interpret the packets transmitted after the last Acknowledge request. If the packets are critical, transmit the packets again. If this error is recurrent, analyze the connectivity between the host and the device carefully.
8	ACK_WITH_ERR_8	The Acknowledge with Error packet contains this error. The device detects and corrects the 1 bit ECC error in at least one of the received packets	No action is required. The device acknowledges the packet. If this error is recurrent, analyze the signal integrity or the noise conditions of the link.

Table 47.8. Error Cause and Recovery for INT_ST0 Register (continued)

Bit	Name	Cause of the Error	Recommended Method of Handling the Error
7	ACK_WITH_ERR_7	The Acknowledge with Error packet contains this error. The device detects the Line Contention through LP0/LP1 detection.	This error might corrupt the low-power data reception and transmission. Ignore the packets and transmit them again. The device recovers automatically. If this error is recurrent, check the device capabilities and the connectivity between the host and device.
6	ACK_WITH_ERR_6	The Acknowledge with Error packet contains this error. The device detects the False Control Error.	The device detects one of the following: • The LP-10 (LP request) is not followed by the remainder of a valid escape or turnaround sequence. • The LP-01 (HS request) is not followed by a bridge state (LP-00). The D-PHY communications are corrupted. This error is unrecoverable. Reset the DSI host and the D-PHY.
5	ACK_WITH_ERR_5	The Acknowledge with Error packet contains this error. The display timeout counters for a HS reception and LP transmission expire.	It is possible that the host and device timeout counters are not correctly configured. The device HS_TX timeout should be shorter than the host HS_RX timeout. Host LP_RX timeout should be longer than the device LP_TX timeout. Check and confirm that the host configuration is consistent with the device specifications. This error is automatically recovered, although there is no guarantee that all the packets in the transmission or reception are complete.
4	ACK_WITH_ERR_4	The Acknowledge with Error packet contains this error. The device reports that the LPDT is not aligned in an 8-bit boundary	There is no guarantee that the device properly receives the packets. Transmit the packets again.
3	ACK_WITH_ERR_3	The Acknowledge with Error packet contains this error. The device does not recognize the Escape Mode Entry command.	The device does not recognize the Escape Mode Entry code. Check the device capability. Repeat the transmission to the device.
2	ACK_WITH_ERR_2	The Acknowledge with Error packet contains this error. The device detects the HS transmission did not end in an 8-bit boundary when the EoT sequence is detected.	There is no guarantee that the device properly received the packets. Re-transmission should be performed. Transmit the packets again.
1	ACK_WITH_ERR_1	The Acknowledge with Error packet contains this error. The device detects that the SoT leader sequence is corrupted.	The device discards the incoming transmission. Re-transmission should be performed by the host.
0	ACK_WITH_ERR_0	The Acknowledge with Error packet contains this error. The device reports that the SoT sequence is received with errors but synchronization can still be achieved.	The device is tolerant to single bit and some multi-bit errors in the SoT sequence. Yet, the packet correctness is compromised. If the packet content was important, transmit the packets again.

Table 47.9. Error Cause and Recovery for INT_ST1 Register

Bit	NAME	Cause of the Error	Recommended Method of Handling the Error
12	GEN_PLD_RECV_ERR	An overflow occurs in the Generic read FIFO.	The Read FIFO size is not correctly dimensioned for the maximum read-back packet size. Configure the device to return the read data with a suitable size for the host dimensioned FIFO. Data stored in the FIFOs is corrupted. Reset the DSI host and repeat the read procedure.
11	GEN_PLD_RD_ERR	An underflow occurs in the Generic read FIFO.	System does not wait for the read procedure to end and starts retrieving the data from the FIFO. The read data is requested before it is fully received. Data is corrupted. Reset the DSI host and repeat the read procedure. Check that the read procedure is completed before reading the data through the User interface.

Table 47.9. Error Cause and Recovery for INT_ST1 Register (continued)

Bit	NAME	Cause of the Error	Recommended Method of Handling the Error
10	GEN_PLD_SEND_ERR	An underflow occurs in the Generic write payload FIFO.	The system writes the packet header before the respective packet payload is completely loaded into the payload FIFO. This error is unrecoverable, the transmitted packet is corrupted. Reset the DSI host and repeat the write procedure.
9	GEN_PLD_WR_ERR	An overflow occurs in the Generic write payload FIFO.	The payload FIFO size is not correctly dimensioned to store the total payload of a long packet. Data stored in the FIFOs is corrupted. Reset the DSI host and repeat the write procedure.
8	GEN_CMD_WR_ERR	An overflow occurs in the Generic command FIFO.	The command FIFO size is not correctly dimensioned to store the total headers of a burst of packets. Data stored in the FIFOs is corrupted. Reset the DSI host and repeat the write procedure.
7	DPI_PLD_WR_ERR	An overflow occurs in the DPI pixel payload FIFO.	The controller FIFO dimensions are not correctly set up for the operating resolution. Check the Video Mode configuration registers. They should be consistent with the LCDC video resolution. The pixel data sequence is corrupted. Reset the DSI host and re-initiate the Video transmission.
6	EOPT_ERR	Host receives a transmission that does not end with an End of Transmission packet.	This error is not critical for the data integrity of the received packets. Check if the device supports the transmission of EoTp packets.
5	PKT_SIZE_ERR	Host receives a transmission that does not end in the expected by boundaries.	The integrity of the received data cannot be guaranteed. Reset the DSI host and repeat the read procedure.
4	CRC_ERR	Host reports that a received long packet has a CRC error in its payload.	The received payload data is corrupted. Reset the DSI host and repeat the read procedure. If this error is recurrent, check the DSI connectivity link for the noise levels.
3	ECC_MULTI_ERR	Host reports that a received packet contains multiple ECC errors.	The received packet is corrupted. The DSI host ignores all the following packets. The DSI host should repeat the read procedure.
2	ECC_SINGLE_ERR	Host reports that a received packet contains a single bit error.	This error is not critical because the DSI host can correct the error and properly decode the packet. If this error is recurrent, check the DSI connectivity link for signal integrity and noise levels.
1	TO_LP_RX	Host reports that the configured timeout counter for the low-power reception has expired.	Once the configured timeout counter ends, the DSI host automatically resets the controller side and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer should be greater than the maximum low-power transmission generated by the device.
0	TO_HS_TX	Host reports that the configured timeout counter for the high-speed transmission has expired.	Once the configured timeout counter ends, the DSI host automatically resets the controller side and recovers to normal operation. Packet transmissions happening during this event are lost. If this error is recurrent, check the timer configuration for any issue. This timer should be greater than the maximum high-speed transmission bursts generated by the host.

47.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x03	Reserved									
0x04	DSI_PWR_UP	31:24								
		23:16								
		15:8								
		7:0								SHUTDOWNZ
0x08	DSI_CLKMGR_CFG	31:24								
		23:16								
		15:8	TO_CLK_DIVISION[7:0]							
		7:0	TX_ESC_CLK_DIVISION[7:0]							
0x0C	DSI_DPI_VCID	31:24								
		23:16								
		15:8								
		7:0							DPI_VCID[1:0]	
0x10	DSI_DPI_COLOR_CODING	31:24								
		23:16								
		15:8								
		7:0					DPI_COLOR_CODING[3:0]			
0x14	DSI_DPI_CFG_POL	31:24								
		23:16								
		15:8								
		7:0				COLORM_ACTIVE_LOW	SHUTD_ACTIVE_LOW	HSYNC_ACTIVE_LOW	VSYNC_ACTIVE_LOW	DATAEN_ACTIVE_LOW
0x18	DSI_DPI_LP_CMD_TIM	31:24								
		23:16	OUTVACT_LPCMD_TIME[7:0]							
		15:8								
		7:0	INVACT_LPCMD_TIME[7:0]							
0x1C ... 0x2B	Reserved									
0x2C	DSI_PCKHDL_CFG	31:24								
		23:16								
		15:8								
		7:0			EOTP_TX_LP_EN	CRC_RX_EN	ECC_RX_EN	BTA_EN	EOTP_RX_EN	EOTP_TX_EN
0x30	DSI_GEN_VCID	31:24								
		23:16							GEN_VCID_TX_AUTO[1:0]	
		15:8							GEN_VCID_TEAR_AUTO[1:0]	
		7:0							GEN_VCID_RX[1:0]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x34	DSI_MODE_CFG	31:24								
		23:16								
		15:8								
		7:0								CMD_VIDEO_MODE
0x38	DSI_VID_MODE_CFG	31:24								VPG_ORIENTATION
		23:16				VPG_MODE				VPG_EN
		15:8	LP_CMD_EN	FRAME_BTA_ACK_EN	LP_HFP_EN	LP_HBP_EN	LP_VACT_EN	LP_VFP_EN	LP_VBP_EN	LP_VSA_EN
		7:0							VID_MODE_TYPE[1:0]	
0x3C	DSI_VID_PKT_SIZE	31:24								
		23:16								
		15:8			VID_PKT_SIZE[13:8]					
		7:0	VID_PKT_SIZE[7:0]							
0x40	DSI_VID_NUM_CHUNKS	31:24								
		23:16								
		15:8				VID_NUM_CHUNKS[12:8]				
		7:0	VID_NUM_CHUNKS[7:0]							
0x44	DSI_VID_NULL_SIZE	31:24								
		23:16								
		15:8				VID_NULL_SIZE[12:8]				
		7:0	VID_NULL_SIZE[7:0]							
0x48	DSI_VID_HSA_TIME	31:24								
		23:16								
		15:8					VID_HSA_TIME[11:8]			
		7:0	VID_HSA_TIME[7:0]							
0x4C	DSI_VID_HBP_TIME	31:24								
		23:16								
		15:8					VID_HBP_TIME[11:8]			
		7:0	VID_HBP_TIME[7:0]							
0x50	DSI_VID_HLINE_TIME	31:24								
		23:16								
		15:8		VID_HLINE_TIME[14:8]						
		7:0	VID_HLINE_TIME[7:0]							
0x54	DSI_VID_VSA_LINES	31:24								
		23:16								
		15:8							VSA_LINES[9:8]	
		7:0	VSA_LINES[7:0]							
0x58	DSI_VID_VBP_LINES	31:24								
		23:16								
		15:8							VBP_LINES[9:8]	
		7:0	VBP_LINES[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x5C	DSI_VID_VFP_LINES	31:24								
		23:16								
		15:8							VFP_LINES[9:8]	
		7:0	VFP_LINES[7:0]							
0x60	DSI_VID_VACTIVE_LINES	31:24								
		23:16								
		15:8			V_ACTIVE_LINES[13:8]					
		7:0	V_ACTIVE_LINES[7:0]							
0x64 ... 0x67	Reserved									
0x68	DSI_CMD_MODE_CFG	31:24								MAX_RD_PKT_SIZE
		23:16					DCS_LW_TX	DCS_SR_0P_TX	DCS_SW_1P_TX	DCS_SW_0P_TX
		15:8		GEN_LW_TX	GEN_SR_2P_TX	GEN_SR_1P_TX	GEN_SR_0P_TX	GEN_SW_2P_TX	GEN_SW_1P_TX	GEN_SW_0P_TX
		7:0							ACK_RQST_EN	TEAR_FX_EN
0x6C	DSI_GEN_HDR	31:24								
		23:16	GEN_WC_MSBYTE[7:0]							
		15:8	GEN_WC_LSBYTE[7:0]							
		7:0	GEN_VC[1:0]		GEN_DT[5:0]					
0x70	DSI_GEN_PLD_DATA	31:24	GEN_PLD_B4[7:0]							
		23:16	GEN_PLD_B3[7:0]							
		15:8	GEN_PLD_B2[7:0]							
		7:0	GEN_PLD_B1[7:0]							
0x74	DSI_CMD_PKT_STATUS	31:24								
		23:16					GEN_BUFF_PLD_FULL	GEN_BUFF_PLD_EMPTY	GEN_BUFF_CMD_FULL	GEN_BUFF_CMD_EMPTY
		15:8								
		7:0		GEN_RD_CMD_BUSY	GEN_PLD_R_FULL	GEN_PLD_R_EMPTY	GEN_PLD_W_FULL	GEN_PLD_W_EMPTY	GEN_CMD_FULL	GEN_CMD_EMPTY
0x78	DSI_TO_CNT_CFG	31:24	HSTX_TO_CNT[15:8]							
		23:16	HSTX_TO_CNT[7:0]							
		15:8	LPRX_TO_CNT[15:8]							
		7:0	LPRX_TO_CNT[7:0]							
0x7C	DSI_HS_RD_TO_CNT	31:24								
		23:16								
		15:8	HS_RD_TO_CNT[15:8]							
		7:0	HS_RD_TO_CNT[7:0]							
0x80	DSI_LP_RD_TO_CNT	31:24								
		23:16								
		15:8	LP_RD_TO_CNT[15:8]							
		7:0	LP_RD_TO_CNT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x84	DSI_HS_WR_TO_CNT	31:24								
		23:16								
		15:8	HS_WR_TO_CNT[15:8]							
		7:0	HS_WR_TO_CNT[7:0]							
0x88	DSI_LP_WR_TO_CNT	31:24								
		23:16								
		15:8	LP_WR_TO_CNT[15:8]							
		7:0	LP_WR_TO_CNT[7:0]							
0x8C	DSI_BTA_TO_CNT	31:24								
		23:16								
		15:8	BTA_TO_CNT[15:8]							
		7:0	BTA_TO_CNT[7:0]							
0x90	DSI_SDF_3D	31:24								
		23:16								SEND_3D_CFG
		15:8								
		7:0			RIGHT_FIRST	SECOND_VSYNC	FORMAT_3D[1:0]		MODE_3D[1:0]	
0x94	DSI_LPCLK_CTRL	31:24								
		23:16								
		15:8								
		7:0							AUTO_CLKLANE_CTRL	PHY_TXREQUESTCLKHS
0x98	DSI_DPHY_TMR_LPCLK_CFG	31:24							PHY_CLKHS2LP_TIME[9:8]	
		23:16	PHY_CLKHS2LP_TIME[7:0]							
		15:8							PHY_CLKLP2HS_TIME[9:8]	
		7:0	PHY_CLKLP2HS_TIME[7:0]							
0x9C	DSI_DPHY_TMR_CFG	31:24							PHY_HS2LP_TIME[9:8]	
		23:16	PHY_HS2LP_TIME[7:0]							
		15:8							PHY_LP2HS_TIME[9:8]	
		7:0	PHY_LP2HS_TIME[7:0]							
0xA0	DSI_DPHY_RSTZ	31:24								
		23:16								
		15:8								
		7:0					DPHY_FORCEPLL	DPHY_ENABLECLK	DPHY_RSTZ	DPHY_SHUTDOWNZ
0xA4	DSI_DPHY_IF_CFG	31:24								
		23:16								
		15:8	PHY_STOP_WAIT_TIME[7:0]							
		7:0							N_LANES[1:0]	
0xA8	DSI_DPHY_ULPS_CTRL	31:24								
		23:16								
		15:8								
		7:0					PHY_TXEXITULPSLAN	PHY_TXREQULPSLAN	PHY_TXEXITULPSCLK	PHY_TXREQULPSCLK

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xAC	DSI_DPHY_TX_TRIGGERS	31:24								
		23:16								
		15:8								
		7:0					PHY_TX_TRIGGERS[3:0]			
0xB0	DSI_DPHY_STATUS	31:24								
		23:16								
		15:8				PHY_ULPSACTIVENOT3LANE	PHY_STOPSTATE3LANE	PHY_ULPSACTIVENOT2LANE	PHY_STOPSTATE2LANE	PHY_ULPSACTIVENOT1LANE
		7:0	PHY_STOPSTATE1LANE	PHY_RXULPSESC0LANE	PHY_ULPSACTIVENOT0LANE	PHY_STOPSTATE0LANE	PHY_ULPSACTIVENOTCLK	PHY_STOPSTATECLKLANE	PHY_DIRECTION	PHY_LOCK
0xB4	DSI_DPHY_TST_CTRL0	31:24								
		23:16								
		15:8								
		7:0							PHY_TESTCLK	PHY_TESTCLR
0xB8	DSI_DPHY_TST_CTRL1	31:24								
		23:16								PHY_TESTEN
		15:8	PHY_TESTDOUT[7:0]							
		7:0	PHY_TESTDIN[7:0]							
0xBC	DSI_INT_ST0	31:24								
		23:16				DPHY_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_0
		15:8	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_8
		7:0	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
0xC0	DSI_INT_ST1	31:24								
		23:16					DPI_BUFF_PLD_UNDER			
		15:8				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
		7:0	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
0xC4	DSI_INT_MSK0	31:24								
		23:16				DPHY_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_0
		15:8	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_8
		7:0	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
0xC8	DSI_INT_MSK1	31:24								
		23:16					DPI_BUFF_PLD_UNDER			
		15:8				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
		7:0	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
0xCC	DSI_DPHY_CAL	31:24								
		23:16								
		15:8								
		7:0								TXSKEWCALHS

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xD0 ... 0xD7	Reserved									
0xD8	DSI_INT_FORCE0	31:24								
		23:16				DPHY_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_0
		15:8	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_8
		7:0	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
0xDC	DSI_INT_FORCE1	31:24								
		23:16					DPI_BUFF_PLD_UNDER			
		15:8				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
		7:0	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
0xE0 ... 0xF3	Reserved									
0xF4	DSI_DPHY_TMR_RD_CFG	31:24								
		23:16								
		15:8		MAX_RD_TIME[14:8]						
		7:0	MAX_RD_TIME[7:0]							
0xF8 ... 0xFF	Reserved									
0x0100	DSI_VID_SHADOW_CTRL	31:24								
		23:16								
		15:8								VID_SHADOW_REQ
		7:0								VID_SHADOW_EN
0x0104 ... 0x010B	Reserved									
0x010C	DSI_DPI_VCID_ACT	31:24								
		23:16								
		15:8								
		7:0							DPI_VCID[1:0]	
0x0110	DSI_DPI_COLOR_CODING_ACT	31:24								
		23:16								
		15:8								LOOSELY18_EN
		7:0					DPI_COLOR_CODING[3:0]			
0x0114 ... 0x0117	Reserved									

Register Summary (continued)

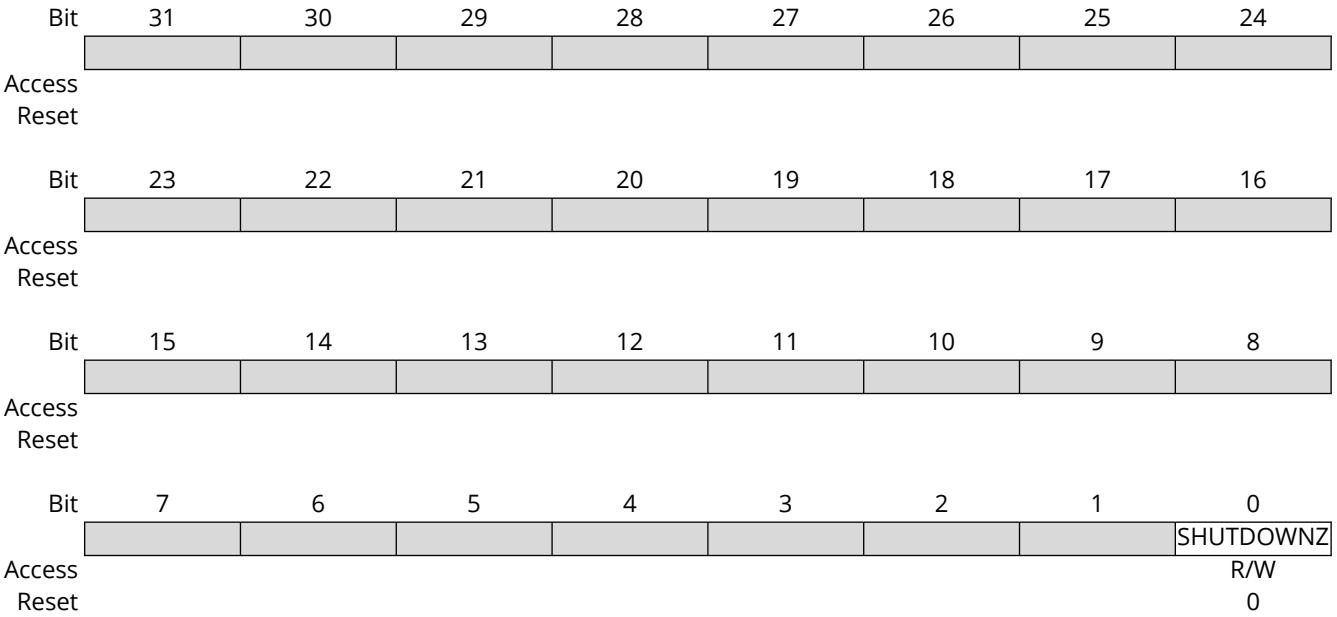
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0118	DSI_DPI_LP_CMD_TIM_ACT	31:24								
		23:16	OUTVACT_LPCMD_TIME[7:0]							
		15:8								
		7:0	INVACT_LPCMD_TIME[7:0]							
0x011C ... 0x0137	Reserved									
0x0138	DSI_VID_MODE_CFG_ACT	31:24								
		23:16								
		15:8							LP_CMD_EN	FRAME_BTA_ACK_EN
		7:0	LP_HFP_EN	LP_HBP_EN	LP_VACT_EN	LP_VFP_EN	LP_VBP_EN	LP_VSA_EN	VID_MODE_TYPE[1:0]	
0x013C	DSI_VID_PKT_SIZE_ACT	31:24								
		23:16								
		15:8			VID_PKT_SIZE[13:8]					
		7:0	VID_PKT_SIZE[7:0]							
0x0140	DSI_VID_NUM_CHUNKS_ACT	31:24								
		23:16								
		15:8				VID_NUM_CHUNKS[12:8]				
		7:0	VID_NUM_CHUNKS[7:0]							
0x0144	DSI_VID_NULL_SIZE_ACT	31:24								
		23:16								
		15:8				VID_NULL_SIZE[12:8]				
		7:0	VID_NULL_SIZE[7:0]							
0x0148	DSI_VID_HSA_TIME_ACT	31:24								
		23:16								
		15:8					VID_HSA_TIME[11:8]			
		7:0	VID_HSA_TIME[7:0]							
0x014C	DSI_VID_HBP_TIME_ACT	31:24								
		23:16								
		15:8					VID_HBP_TIME[11:8]			
		7:0	VID_HBP_TIME[7:0]							
0x0150	DSI_VID_HLINE_TIME_ACT	31:24								
		23:16								
		15:8		VID_HLINE_TIME[14:8]						
		7:0	VID_HLINE_TIME[7:0]							
0x0154	DSI_VID_VSA_LINES_ACT	31:24								
		23:16								
		15:8							VSA_LINES[9:8]	
		7:0	VSA_LINES[7:0]							
0x0158	DSI_VID_VBP_LINES_ACT	31:24								
		23:16								
		15:8							VBP_LINES[9:8]	
		7:0	VBP_LINES[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x015C	DSI_VID_VFP_LINES_ACT	31:24								
		23:16								
		15:8							VFP_LINES[9:8]	
		7:0	VFP_LINES[7:0]							
0x0160	DSI_VID_VACTIVE_LINES_ACT	31:24								
		23:16								
		15:8			V_ACTIVE_LINES[13:8]					
		7:0	V_ACTIVE_LINES[7:0]							
0x0164 ... 0x0167	Reserved									
0x0168	DSI_VID_PKT_STATUS	31:24								
		23:16							DPI_BUFF_PLD_FULL	DPI_BUFF_PLD_EMPTY
		15:8								
		7:0					DPI_PLD_W_FULL	DPI_PLD_W_EMPTY	DPI_CMD_W_FULL	DPI_CMD_W_EMPTY
0x016C ... 0x018F	Reserved									
0x0190	DSI_SDF_3D_ACT	31:24								
		23:16								SEND_3D_CFG
		15:8								
		7:0			RIGHT_FIRST	SECOND_VSYNC	FORMAT_3D[1:0]		MODE_3D[1:0]	

47.7.1. DSI Power-Up Control Register

Name: DSI_PWR_UP
Offset: 0x04
Reset: 0x00000000
Property: Read/Write



Bit 0 – SHUTDOWNZ Shutdown

Value	Description
0	DSI host is reset.
1	DSI host is powered up.

47.7.2. DSI Clock Management Configuration Register

Name: DSI_CLKMGR_CFG
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	TO_CLK_DIVISION[7:0]							
Access	R/W	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
	TX_ESC_CLK_DIVISION[7:0]							
Access	R/W	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 – TO_CLK_DIVISION[7:0] Timeout Clock Division
Indicates the division factor for the timeout clock used as the timing unit in the configuration of HS to LP and LP to HS transition error.

Bits 7:0 – TX_ESC_CLK_DIVISION[7:0] Transmission Escape Clock Division
Indicates the division factor for the TX Escape clock source (lanebyteclk). The values 0 and 1 stop the TX_ESC clock generation.

47.7.3. DSI Input Video Virtual Channel ID Configuration Register

Name: DSI_DPI_VCID
Offset: 0x0C
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
							DPI_VCID[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – DPI_VCID[1:0] Input Video Virtual Channel Identifier
Configures the virtual channel identifier that is indexed to the Video mode packets.

47.7.4. DSI Input Video Color Coding Configuration Register

Name: DSI_DPI_COLOR_CODING
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
					DPI_COLOR_CODING[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – DPI_COLOR_CODING[3:0] Input Video Color Coding
Configures the video color coding as follows:

Value	Name	Description
0	16BIT_CFG1	16-bit configuration 1
1	16BIT_CFG2	16-bit configuration 2
2	16BIT_CFG3	16-bit configuration 3
3	18BIT_CFG1	18-bit configuration 1
4	18BIT_CFG2	18-bit configuration 2
5	24BIT	24-bit
6	–	Reserved
7	–	Reserved
8	–	Reserved
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved
13, 14, 15	–	Reserved

47.7.5. DSI Input Video Polarity Configuration Register

Name: DSI_DPI_CFG_POL
Offset: 0x14
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
				COLORM_ACTIVE_LOW	SHUTD_ACTIVE_LOW	HSYNC_ACTIVE_LOW	VSYNC_ACTIVE_LOW	DATAEN_ACTIVE_LOW
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

Bit 4 – COLORM_ACTIVE_LOW Color Mode Signal Active Low

Value	Description
0	Input interface Color Mode signal is active high.
1	Input interface Color Mode signal is active low.

Bit 3 – SHUTD_ACTIVE_LOW Shutdown Signal Active Low

Value	Description
0	Input interface Shutdown signal is active high.
1	Input interface Shutdown signal is active low.

Bit 2 – HSYNC_ACTIVE_LOW Horizontal Synchronization Signal Active Low

Value	Description
0	Input video interface VSYNC is active high.
1	Input video interface VSYNC is active low.

Bit 1 – VSYNC_ACTIVE_LOW Vertical Synchronisation Signal Active Low

Value	Description
0	Input video interface VSYNC is active high.
1	Input video interface VSYNC is active low.

Bit 0 – DATAEN_ACTIVE_LOW Data Enable Signal Active Low

Value	Description
0	Input video interface DATAEN is active high.

Value	Description
1	Input video interface DATAEN is active low.

47.7.6. DSI Input Video Timing for Low-Power Commands Configuration Register

Name: DSI_DPI_LP_CMD_TIM
Offset: 0x18
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
	OUTVACT_LPCMD_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

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INVACT_LPCMD_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 23:16 – OUTVACT_LPCMD_TIME[7:0] Output Vertical Active Low-Power Command
Used for the transmission of commands in Low-Power mode. It defines the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions.

Bits 7:0 – INVACT_LPCMD_TIME[7:0] Input Vertical Active Low-Power Command Time
Used for the transmission of commands in Low-Power mode. It defines the size, in bytes, of the largest packet that can fit in a line during the VACT region.

47.7.7. DSI Protocol Configuration Register

Name: DSI_PCKHDL_CFG
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
			EOTP_TX_LP_EN	CRC_RX_EN	ECC_RX_EN	BTA_EN	EOTP_RX_EN	EOTP_TX_EN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – EOTP_TX_LP_EN End Of Transmission Packet Low Power Enable

Value	Description
0	Disables the EoTp transmission in low power.
1	Enables the EoTp transmission in low power.

Bit 4 – CRC_RX_EN CRC Reception Enable

Value	Description
0	Disables the CRC reception and error reporting.
1	Enables the CRC reception and error reporting.

Bit 3 – ECC_RX_EN ECC Reception Enable

Value	Description
0	Disables the ECC reception, error correction, and reporting.
1	Enables the ECC reception, error correction, and reporting.

Bit 2 – BTA_EN Bus Turn-Around Enable

Value	Description
0	Disables the Bus Turnaround (BTA) request.
1	Enables the Bus Turnaround (BTA) request.

Bit 1 – EOTP_RX_EN End Of Reception Packet Enable

Value	Description
0	Disables the EoTp reception.

Value	Description
1	Enables the EoTp reception.

Bit 0 – EOTP_TX_EN End Of Transmission Packet Enable

Value	Description
0	Disables the EoTp transmission.
1	Enables the EoTp transmission.

47.7.8. DSI Read Generic Virtual Channel ID Configuration Register

Name: DSI_GEN_VCID
Offset: 0x30
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
							GEN_VCID_TX_AUTO[1:0]	
Access							R/W	R/W
Reset							0	0

Bit	15	14	13	12	11	10	9	8
							GEN_VCID_TEAR_AUTO[1:0]	
Access							R/W	R/W
Reset							0	0

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

Bits 17:16 – GEN_VCID_TX_AUTO[1:0] Generic Transmission Virtual Channel Identifier
Indicates the generic interface virtual channel identification where the generic packet is automatically generated and transmitted.

Bits 9:8 – GEN_VCID_TEAR_AUTO[1:0] Generic Tearing Effect Virtual Channel Identifier
Indicates the virtual channel identification for tear effect by hardware.

Bits 1:0 – GEN_VCID_RX[1:0] Generic Reception Virtual Channel Identifier
Indicates the generic interface read-back virtual channel identification.

47.7.9. DSI Mode of Operation Configuration Register

Name: DSI_MODE_CFG
Offset: 0x34
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
								CMD_VIDEO_MODE
Access								R/W
Reset								1

Bit 0 – CMD_VIDEO_MODE Command/Video Mode

Value	Description
0	Video mode.
1	Command mode.

47.7.10. DSI Video Mode Configuration Register

Name: DSI_VID_MODE_CFG
Offset: 0x38
Reset: 0x00000000
Property: Read/Write

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

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

Bit	15	14	13	12	11	10	9	8
	LP_CMD_EN	FRAME_BTA_ACK_EN	LP_HFP_EN	LP_HBP_EN	LP_VACT_EN	LP_VFP_EN	LP_VBP_EN	LP_VSA_EN
Access	R/W	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
							VID_MODE_TYPE[1:0]	
Access							R/W	R/W
Reset							0	0

Bit 24 – VPG_ORIENTATION Video Pattern Generator Orientation

Value	Description
0	Vertical color bar orientation.
1	Horizontal color bar orientation.

Bit 20 – VPG_MODE Video Pattern Generator Mode

Value	Description
0	Color bar (horizontal or vertical) pattern is selected.
1	BER pattern (vertical only) is selected.

Bit 16 – VPG_EN Video Pattern Generator Enable

Value	Description
0	Disables the video mode pattern generator.
1	Enables the video mode pattern generator.

Bit 15 – LP_CMD_EN Low Power Command Transmission Enable

Value	Description
0	Disables the command transmission only in Low-Power mode.
1	Enables the command transmission only in Low-Power mode.

Bit 14 – FRAME_BTA_ACK_EN Frame Bus Turn Around Acknowledge Enable

Value	Description
0	Disables the request for an acknowledge response at the end of a frame.
1	Enables the request for an acknowledge response at the end of a frame.

Bit 13 – LP_HFP_EN Low-Power Horizontal Front Porch Enable

Value	Description
0	Disables the return to low-power inside the HFP period when timing allows.
1	Enables the return to low-power inside the HFP period when timing allows.

Bit 12 – LP_HBP_EN Low-Power Horizontal Back Porch Enable

Value	Description
0	Disables the return to low-power inside the HBP period when timing allows.
1	Enables the return to low-power inside the HBP period when timing allows.

Bit 11 – LP_VACT_EN Low-Power Vertical Active Enable

Value	Description
0	Disables the return to low-power inside the VACT period when timing allows.
1	Enables the return to low-power inside the VACT period when timing allows.

Bit 10 – LP_VFP_EN Low-Power Vertical Front Porch Enable

Value	Description
0	Disables the return to low-power inside the VFP period when timing allows.
1	Enables the return to low-power inside the VFP period when timing allows.

Bit 9 – LP_VBP_EN Low-Power Vertical Back Porch Enable

Value	Description
0	Disables the return to low-power inside the VBP period when timing allows.
1	Enables the return to low-power inside the VBP period when timing allows.

Bit 8 – LP_VSA_EN Low-Power Vertical Sync Active Enable

Value	Description
0	Disables the return to low-power inside the VSA period when timing allows.
1	Enables the return to low-power inside the VSA period when timing allows.

Bits 1:0 – VID_MODE_TYPE[1:0] Video Mode Transmission Type
This field indicates the video mode transmission type:

Value	Description
0	Non-burst with sync pulses
1	Non-burst with sync events
2	Burst mode
3	Burst mode

47.7.11. DSI Video Mode Packet Size Configuration Register

Name: DSI_VID_PKT_SIZE
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			VID_PKT_SIZE[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
	VID_PKT_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

Bits 13:0 – VID_PKT_SIZE[13:0] Video Mode Packet Size
Configures the number of pixels in a single video packet.
For 18-bit data types, this number must be a multiple of 4.

47.7.12. DSI Video Mode Number of Chunks Configuration Register

Name: DSI_VID_NUM_CHUNKS
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
				VID_NUM_CHUNKS[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
	VID_NUM_CHUNKS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – VID_NUM_CHUNKS[12:0] Video Mode Number of Chunks
Configures the number of chunks to be transmitted during a Line period (a chunk is pair made up of a video packet and a null packet). If set to 0 or 1, the video line is still transmitted in a single packet. If set to 1, the packet is part of a chunk, meaning that a null packet follows it (if vid_null_size>0). Otherwise, multiple chunks are used to transmit each video line.

47.7.13. DSI Video Mode Null Packets Size Configuration Register

Name: DSI_VID_NULL_SIZE
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
				VID_NULL_SIZE[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
	VID_NULL_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

Bits 12:0 – VID_NULL_SIZE[12:0] Video Mode Null Packet Size
Configures the number of bytes inside a null packet. Setting to 0 disables null packets.

47.7.14. DSI Video Mode HSA Time Configuration Register

Name: DSI_VID_HSA_TIME
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
					VID_HSA_TIME[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
	VID_HSA_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 11:0 – VID_HSA_TIME[11:0] Video Mode Horizontal Sync Active Time
Configures the Horizontal Synchronism Active period in lane byte clock cycles.

47.7.15. DIS Video Mode HBP Time Configuration Register

Name: DSI_VID_HBP_TIME
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
					VID_HBP_TIME[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
	VID_HBP_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 11:0 – VID_HBP_TIME[11:0] Video Mode Horizontal Back PorchTime
Configures the Horizontal Back Porch period in lane byte clock cycles.

47.7.16. DIS Video Mode HLINE Time Configuration Register

Name: DSI_VID_HLINE_TIME
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
		VID_HLINE_TIME[14:8]						
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
	VID_HLINE_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 14:0 – VID_HLINE_TIME[14:0] Video Mode Horizontal Line Time
Configures the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles.

47.7.17. DSI Video Mode VSA Lines Configuration Register

Name: DSI_VID_VSA_LINES
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
Access								
Reset								

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

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

Bits 9:0 – VSA_LINES[9:0] Vertical Sync Active Lines
Configures the Vertical Synchronism Active period measured in number of horizontal lines.

47.7.18. DSI Video Mode VBP Lines Configuration Register

Name: DSI_VID_VBP_LINES
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
							VBP_LINES[9:8]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:0 – VBP_LINES[9:0] Vertical Back Porch Lines
Configures the Vertical Back Porch period measured in number of horizontal lines.

47.7.19. DSI Video Mode VFP Lines Configuration Register

Name: DSI_VID_VFP_LINES
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
							VFP_LINES[9:8]	
Access							R/W	R/W
Reset							0	0

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

Bits 9:0 – VFP_LINES[9:0] Video Mode Vertical Front Porch Lines
Configures the Vertical Front Porch period measured in number of horizontal lines.

47.7.20. DSI Video Mode VACTIVE Lines Configuration Register

Name: DSI_VID_VACTIVE_LINES
Offset: 0x60
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
			V_ACTIVE_LINES[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
	V_ACTIVE_LINES[7:0]							
Access	R/W	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 – V_ACTIVE_LINES[13:0] Video Mode Active Lines
Configures the Vertical Active period measured in number of horizontal lines.

47.7.21. DSI Command Mode Configuration Register

Name: DSI_CMD_MODE_CFG
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

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

Bit	23	22	21	20	19	18	17	16
					DCS_LW_TX	DCS_SR_0P_TX	DCS_SW_1P_TX	DCS_SW_0P_TX
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
		GEN_LW_TX	GEN_SR_2P_TX	GEN_SR_1P_TX	GEN_SR_0P_TX	GEN_SW_2P_TX	GEN_SW_1P_TX	GEN_SW_0P_TX
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
							ACK_RQST_EN	TEAR_FX_EN
Access							R/W	R/W
Reset							0	0

Bit 24 – MAX_RD_PKT_SIZE Maximum Read Packet Size
Configures the maximum read packet size command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 19 – DCS_LW_TX DCS Long Write Packet Transmission Type
Configures the DCS long write packet command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 18 – DCS_SR_0P_TX DCS Short Read Packet 0 Transmission Type
Configures the DCS short read packet with zero parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 17 – DCS_SW_1P_TX DCS Short Write Packet 1 Transmission Type
Configures the DCS short write packet with one parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 16 – DCS_SW_0P_TX DCS Short Write Packet 0 Transmission Type
Configures the DCS short write packet with zero parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 14 – GEN_LW_TX Generic Long Write Packet Transmission Type
Configures the Generic long write packet command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 13 – GEN_SR_2P_TX Generic Short Read Packet 2 Transmission Type
Configures the generic short read packet with two parameters command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 12 – GEN_SR_1P_TX Generic Short Read Packet 1 Transmission Type
Configures the generic short read packet with one parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 11 – GEN_SR_0P_TX Generic Short Read Packet 0 Transmission Type
Configures the generic short read packet with zero parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 10 – GEN_SW_2P_TX Generic Short Write Packet 2 Transmission Type
Configures the generic short write packet with two parameters command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 9 – GEN_SW_1P_TX Generic Short Write Packet 1 Transmission Type
Configures the generic short write packet with one parameter command transmission type:

Value	Description
0	High-speed.
1	Low-power.

Bit 8 – GEN_SW_0P_TX Generic Short Write Packet 0 Transmission Type
Configures the generic short write packet with zero parameter command transmission type:

Value	Description
0	High-speed.

Value	Description
1	Low-power.

Bit 1 – ACK_RQST_EN Acknowledge Request Enable

Value	Description
0	Disables the acknowledge request after each packet transmission.
1	Enables the acknowledge request after each packet transmission.

Bit 0 – TEAR_FX_EN Tearing Effect Enable

Value	Description
0	Disables the tearing effect acknowledge request.
1	Enables the tearing effect acknowledge request.

47.7.22. DSI Generic Interface Header Register

Name: DSI_GEN_HDR
Offset: 0x6C
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
	GEN_WC_MSBYTE[7:0]							
Access	R/W	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
	GEN_WC_LSBYTE[7:0]							
Access	R/W	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
	GEN_VC[1:0]		GEN_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 23:16 – GEN_WC_MSBYTE[7:0] Generic Interface Word Count Most Significant Byte
Configures the most significant byte of the header packet word count for long packets or data 1 for short packets.

Bits 15:8 – GEN_WC_LSBYTE[7:0] Generic Interface Word Count Least Significant Byte
Configures the least significant byte of the header packet word count for long packets or data 0 for short packets.

Bits 7:6 – GEN_VC[1:0] Generic Interface Virtual Channel
Configures the virtual channel ID of the header packet.

Bits 5:0 – GEN_DT[5:0] Generic Interface Data Type
Configures the packet data type of the header packet.

47.7.23. DSI Generic Interface Payload Register

Name: DSI_GEN_PLD_DATA
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	GEN_PLD_B4[7:0]							
Access	R/W	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
	GEN_PLD_B3[7:0]							
Access	R/W	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
	GEN_PLD_B2[7:0]							
Access	R/W	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
	GEN_PLD_B1[7:0]							
Access	R/W	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 – GEN_PLD_B4[7:0] Generic Interface Payload Byte 4
Indicates byte 4 of the packet payload.

Bits 23:16 – GEN_PLD_B3[7:0] Generic Interface Payload Byte 3
Indicates byte 3 of the packet payload.

Bits 15:8 – GEN_PLD_B2[7:0] Generic Interface Payload Byte 2
Indicates byte 2 of the packet payload.

Bits 7:0 – GEN_PLD_B1[7:0] Generic Interface Payload Byte 1
Indicates byte 1 of the packet payload.

47.7.24. DSI Command Packet Status Register

Name: DSI_CMD_PKT_STATUS
Offset: 0x74
Reset: 0x00050015
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					GEN_BUFF_PLD_FULL	GEN_BUFF_PLD_EMPTY	GEN_BUFF_CMD_FULL	GEN_BUFF_CMD_EMPTY
Access					R	R	R	R
Reset					0	1	0	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		GEN_RD_CMD_BUSY	GEN_PLD_R_FULL	GEN_PLD_R_EMPTY	GEN_PLD_W_FULL	GEN_PLD_W_EMPTY	GEN_CMD_FULL	GEN_CMD_EMPTY
Access		R	R	R	R	R	R	R
Reset		0	0	1	0	1	0	1

Bit 19 – GEN_BUFF_PLD_FULL Generic Payload Internal Buffer Full
Indicates the full status of the generic payload internal buffer.

Bit 18 – GEN_BUFF_PLD_EMPTY Generic Payload Internal Buffer Empty
Indicates the empty status of the generic payload internal buffer.

Bit 17 – GEN_BUFF_CMD_FULL Generic Command Internal Buffer Full
Indicates the full status of the generic command internal buffer.

Bit 16 – GEN_BUFF_CMD_EMPTY Generic Command Internal Buffer Empty
Indicates the empty status of the generic command internal buffer.

Bit 6 – GEN_RD_CMD_BUSY Generic Interface Read Command FIFO Busy
Set when a read command is issued and cleared when the entire response is stored in the FIFO for the generic interface.

Bit 5 – GEN_PLD_R_FULL Generic Interface Read Payload FIFO Full
This bit indicates the full status of the generic read payload FIFO.

Bit 4 – GEN_PLD_R_EMPTY Generic Interface Read Payload FIFO Empty
Indicates the empty status of the generic read payload FIFO.

Bit 3 – GEN_PLD_W_FULL Generic Interface Write Payload FIFO Full
Indicates the full status of the generic write payload FIFO.

Bit 2 – GEN_PLD_W_EMPTY Generic Interface Write Payload FIFO Empty
Indicates the empty status of the generic write payload FIFO.

Bit 1 – GEN_CMD_FULL Generic Interface Command FIFO Full
Indicates the full status of the generic command FIFO.

Bit 0 – GEN_CMD_EMPTY Generic Interface Command FIFO Empty
Indicates the empty status of the generic command FIFO.

47.7.25. DSI Timeout Counters Configuration Register

Name: DSI_TO_CNT_CFG
Offset: 0x78
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	HSTX_TO_CNT[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
	HSTX_TO_CNT[7:0]							
Access	R/W	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
	LPRX_TO_CNT[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
	LPRX_TO_CNT[7:0]							
Access	R/W	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 – HSTX_TO_CNT[15:0] High-Speed Transmission Timeout Counter
Configures the timeout counter that triggers a high-speed transmission timeout contention detection (measured in TO_CLK_DIVISION cycles).

Bits 15:0 – LPRX_TO_CNT[15:0] Low-Power Reception Timeout Counter
Configures the timeout counter that triggers a low-power reception timeout contention detection (measured in TO_CLK_DIVISION cycles).

47.7.26. DSI HS Read Response Timeout Counter Configuration Register

Name: DSI_HS_RD_TO_CNT
Offset: 0x7C
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
	HS_RD_TO_CNT[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
	HS_RD_TO_CNT[7:0]							
Access	R/W	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 – HS_RD_TO_CNT[15:0] High-Speed Read Operation Timeout Counter
Sets a period for which the DSI host keeps the link still, after sending a High-Speed Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

47.7.27. DSI LP Read Response Timeout Counter Configuration Register

Name: DSI_LP_RD_TO_CNT
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
	LP_RD_TO_CNT[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
	LP_RD_TO_CNT[7:0]							
Access	R/W	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 – LP_RD_TO_CNT[15:0] Low-Power Read Operation Timeout Counter
Sets a period for which DSI host keeps the link still, after sending a Low-Power Read operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

47.7.28. DSI HS Write Response Timeout Counter Configuration Register

Name: DSI_HS_WR_TO_CNT
Offset: 0x84
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
	HS_WR_TO_CNT[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
	HS_WR_TO_CNT[7:0]							
Access	R/W	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 – HS_WR_TO_CNT[15:0] High-Speed Write Operation Timeout Counter
Sets a period for which the DSI host controller keeps the link still, after sending a High-Speed Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

47.7.29. DSI LP Write Response Timeout Counter Configuration Register

Name: DSI_LP_WR_TO_CNT
Offset: 0x88
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
	LP_WR_TO_CNT[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
	LP_WR_TO_CNT[7:0]							
Access	R/W	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 – LP_WR_TO_CNT[15:0] Low-Power Write Operation Timeout Counter
Sets a period for which DSI host keeps the link still, after sending a High-Speed Write operation. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

47.7.30. DSI BTA Response Timeout Counter Configuration Register

Name: DSI_BTA_TO_CNT
Offset: 0x8C
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
	BTA_TO_CNT[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
	BTA_TO_CNT[7:0]							
Access	R/W	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 – BTA_TO_CNT[15:0] Bus Turn Around Timeout Counter
Sets a period for which DSI host keeps the link still, after completing a bus turnaround. This period is measured in cycles of lanebyteclk, starts to count when the D-PHY enters stop state and causes no interrupts.

47.7.31. DSI VSS Packet 3D Control Register

Name: DSI_SDF_3D
Offset: 0x90
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
								SEND_3D_CFG
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
			RIGHT_FIRST	SECOND_VSYNC	FORMAT_3D[1:0]		MODE_3D[1:0]	
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 16 – SEND_3D_CFG Send 3D Configuration Enable
When set, causes the next VSS packet to include 3D control payload in every VSS packet.

Bit 5 – RIGHT_FIRST Right First
Defines the left/right order:

Value	Description
0	Left eye is sent first, then right eye.
1	Right eye data is sent first, then left eye.

Bit 4 – SECOND_VSYNC Second Vertical Sync Enable
Defines whether there is a second VSYNC pulse between left and right images, when 3D image format is frame-based:

Value	Description
0	No sync pulses between left and right data.
1	Sync pulse (HSYNC, VSYNC, blanking) between left and right data.

Bits 3:2 – FORMAT_3D[1:0] 3D Format
Defines 3D image format:

Value	Description
0	Line (alternating lines of left and right data).
1	Frame (alternating frames of left and right data).
2	Pixel (alternating pixels of left and right data).
3	Reserved

Bits 1:0 – MODE_3D[1:0] 3D Mode
Defines 3D mode on/off and display orientation:

Value	Description
0	3D mode off (2D mode on).
1	3D mode on, portrait orientation.
2	3D mode on, landscape orientation.
3	Reserved.

47.7.32. DSI Clock Lane Control Register

Name: DSI_LPCLK_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
							AUTO_CLKLANE_CTRL	PHY_TXREQUESTCLKHS
Access							R/W	R/W
Reset							0	0

Bit 1 – AUTO_CLKLANE_CTRL Automatic Clock Lane Control

Value	Description
0	Disables the automatic mechanism to stop providing the clock in the clock lane when time allows.
1	Enables the automatic mechanism to stop providing clock in the clock lane when time allows.

Bit 0 – PHY_TXREQUESTCLKHS D-PHY High-Speed Transmission Request

Value	Description
0	Disables High-Speed mode transmission request on clock lane.
1	Enables High-Speed mode transmission request on clock lane.

47.7.33. DSI Clock Lane Switch Mode Timing Configuration Register

Name: DSI_DPHY_TMR_LPCLK_CFG
Offset: 0x98
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							PHY_CLKHS2LP_TIME[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	PHY_CLKHS2LP_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

Bit	15	14	13	12	11	10	9	8
							PHY_CLKLP2HS_TIME[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	PHY_CLKLP2HS_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 25:16 – PHY_CLKHS2LP_TIME[9:0] D-PHY Clock High-Speed to Low-Power Time
Configures the maximum time that the D-PHY clock lane takes to go from high-speed to low-power transmission measured in lane byte clock cycles.

Bits 9:0 – PHY_CLKLP2HS_TIME[9:0] D-PHY Clock Low-Power to High-Speed Time
Configures the maximum time that the D-PHY clock lane takes to go from low-power to high-speed transmission measured in lane byte clock cycles.

47.7.34. DSI Data Lane Switch Mode Timing Configuration Register

Name: DSI_DPHY_TMR_CFG
Offset: 0x9C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
							PHY_HS2LP_TIME[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	PHY_HS2LP_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

Bit	15	14	13	12	11	10	9	8
							PHY_LP2HS_TIME[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	PHY_LP2HS_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 25:16 – PHY_HS2LP_TIME[9:0] D-PHY Data High-Speed to Low-Power Time
Configures the maximum time that the D-PHY data lanes take to go from high-speed to low-power transmission measured in lane byte clock cycles.

Bits 9:0 – PHY_LP2HS_TIME[9:0] D-PHY Data Low-Power to High-Speed Time
Configures the maximum time that the D-PHY data lanes take to go from low-power to high-speed transmission measured in lane byte clock cycles.

47.7.35. DSI Reset and PLL Control Register

Name: DSI_DPHY_RSTZ
Offset: 0xA0
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_FORCEPLL	DPHY_ENABLECLK	DPHY_RSTZ	DPHY_SHUTDOWNZ
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – DPHY_FORCEPLL D-PHY Force PLL Enable

Value	Description
0	Disables the D-PHY PLL when the D-PHY is in ULPS.
1	Enables the D-PHY PLL when the D-PHY is in ULPS.

Bit 2 – DPHY_ENABLECLK D-PHY Enable Clock Lane

Value	Description
0	Disables the D-PHY clock lane module.
1	Enables the D-PHY clock lane module.

Bit 1 – DPHY_RSTZ D-PHY Reset Disable

Value	Description
0	Places the digital section of the D-PHY in the reset state.
1	Allows the digital section of the D-PHY to leave the reset state.

Bit 0 – DPHY_SHUTDOWNZ D-PHY Shutdown Disable

Value	Description
0	Places the complete D-PHY macro in power-down state.
1	Allow the complete D-PHY macro to leave power-down state.

47.7.36. DSI D-PHY Interface Configuration Register

Name: DSI_DPHY_IF_CFG
Offset: 0xA4
Reset: 0x00000003
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
	PHY_STOP_WAIT_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

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

Bits 15:8 – PHY_STOP_WAIT_TIME[7:0] D-PHY Stop State Wait Time
Configures the minimum wait period to request a high-speed transmission after the Stop state.

Bits 1:0 – N_LANES[1:0] Number of Active Data Lanes

Value	Description
0	One data lane (lane 0)
1	Two data lanes (lanes 0 and 1)
2	Three data lanes (lanes 0, 1, and 2)
3	Four data lanes (lanes 0, 1, 2, and 3)

47.7.37. DSI D-PHY ULPS Control Register

Name: DSI_DPHY_ULPS_CTRL
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
					PHY_TXEXITULPSLAN	PHY_TXREQULPSLAN	PHY_TXEXITULPSCLK	PHY_TXREQULPSCLK
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – PHY_TXEXITULPSLAN D-PHY Ultra-Low-Power Exit Transmission on Data Lane

Value	Description
0	No effect.
1	ULPS mode exit on clock lane.

Bit 2 – PHY_TXREQULPSLAN D-PHY Ultra-Low-Power Request Transmission on Data Lane

Value	Description
0	No effect.
1	ULPS mode request on data lane.

Bit 1 – PHY_TXEXITULPSCLK D-PHY Ultra-Low-Power Exit Transmission on Clock Lane

Value	Description
0	No effect.
1	ULPS mode exit on clock lane.

Bit 0 – PHY_TXREQULPSCLK D-PHY Ultra-Low-Power Request Transmission on Clock Lane

Value	Description
0	No effect.
1	ULPS mode request on clock lane.

47.7.38. DSI D-PHY Tx Trigger Control Register

Name: DSI_DPHY_TX_TRIGGERS
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
					PHY_TX_TRIGGERS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – PHY_TX_TRIGGERS[3:0] D-PHY Trigger Transmission
Controls the trigger transmissions

47.7.39. DSI D-PHY Status Register

Name: DSI_DPHY_STATUS
Offset: 0xB0
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
				PHY_ULPSACTIVENOT3LANE	PHY_STOPSTATE3LANE	PHY_ULPSACTIVENOT2LANE	PHY_STOPSTATE2LANE	PHY_ULPSACTIVENOT1LANE
Access				R	R	R	R	R
Reset				–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	PHY_STOPSTATE1LANE	PHY_RXULPSESC0LANE	PHY_ULPSACTIVENOT0LANE	PHY_STOPSTATE0LANE	PHY_ULPSACTIVENOTCLK	PHY_STOPSTATECLKLANE	PHY_DIRECTION	PHY_LOCK
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit 12 – PHY_ULPSACTIVENOT3LANE D-PHY Ultra-Low-Power State Active on Data Lane 3 Status

Value	Description
0	Data lane 3 is in Ultra-Low-Power (ULP) state.
1	Data lane 3 is not in Ultra-Low-Power (ULP) state.

Bit 11 – PHY_STOPSTATE3LANE D-PHY Stop State on Data Lane 3 Status

Value	Description
0	Data lane 3 module is not in Stop state.
1	Data lane 3 module is in Stop state.

Bit 10 – PHY_ULPSACTIVENOT2LANE D-PHY Ultra-Low-Power State Active on Data Lane 2 Status

Value	Description
0	Data lane 2 is in Ultra-Low-Power (ULP) state.
1	Data lane 2 is not in Ultra-Low-Power (ULP) state.

Bit 9 – PHY_STOPSTATE2LANE D-PHY Stop State on Data Lane 2 Status

Value	Description
0	Data lane 2 module is not in Stop state.
1	Data lane 2 module is in Stop state.

Bit 8 – PHY_ULPSACTIVENOT1LANE D-PHY Ultra-Low-Power State Active on Data Lane 1 Status

Value	Description
0	Data lane 1 is in Ultra-Low-Power (ULP) state.
1	Data lane 1 is not in Ultra-Low-Power (ULP) state.

Bit 7 – PHY_STOPSTATE1LANE D-PHY Stop State on Data Lane 1 Statu

Value	Description
0	Data lane 1 module is not in Stop state.
1	Data lane 1 module is in Stop state.

Bit 6 – PHY_RXULPSESC0LANE Ultra-Low-Power State Escape Code Reception

Value	Description
0	Data lane 0 is not in Ultra Low-Power state after reception of the Ultra Low-Power state Escape code.
1	Data lane 0 is in Ultra Low-Power state after reception of the Ultra Low-Power state Escape code.

Bit 5 – PHY_ULPSACTIVENOT0LANE D-PHY Ultra-Low-Power State Active on Data Lane 0 Status

Value	Description
0	Data lane 0 is in Ultra-Low-Power (ULP) state.
1	Data lane 0 is not in Ultra-Low-Power (ULP) state.

Bit 4 – PHY_STOPSTATE0LANE D-PHY Stop State on Data Lane 0 Status

Value	Description
0	Data lane 0 module is not in Stop state.
1	Data lane 0 module is in Stop state.

Bit 3 – PHY_ULPSACTIVENOTCLK D-PHY Ultra-Low-Power State Active on Clock Lane Status

Value	Description
0	The clock lane is in Ultra-Low-Power (ULP) state.
1	The clock lane is not in the Ultra-Low-Power state.

Bit 2 – PHY_STOPSTATECLKLANE D-PHY Stop State Clock Lane Status

Value	Description
0	The clock lane module is not in Stop state.
1	The clock lane module is in Stop state.

Bit 1 – PHY_DIRECTION D-PHY Direction Status
Used to indicate the current direction of the lane interconnect.

Value	Description
0	The lane interconnect is in Transmit mode.
1	The lane interconnect is in Receive mode.

Bit 0 – PHY_LOCK D-PHY Lock Status
When set, indicates that the PLL acquired the lock with the reference clock.

47.7.40. DSI D-PHY Control Register 0

Name: DSI_DPHY_TST_CTRL0
Offset: 0xB4
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
Access							R/W	R/W
Reset							0	1

Bit 1 – PHY_TESTCLK D-PHY Test Interface Clock
Used to clock the TESTDIN bus into the D-PHY.

Bit 0 – PHY_TESTCLR D-PHY Test Interface Clear

Value	Description
0	No effect.
1	Clears the PHY test interface.

47.7.41. DSI D-PHY Control Register 1

Name: DSI_DPHY_TST_CTRL1
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
								PHY_TESTEN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	PHY_TESTDOUT[7:0]							
Access	R/W	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
	PHY_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 – PHY_TESTEN D-PHY Test Interface Operation Type

Value	Name	Description
0	WRITEADDR	The data write operation is set on the rising edge of DSI_DPHY_TST_CTRL0.PHY_TESTCLK.
1	WRITEDATA	The address write operation is set on the falling edge of DSI_DPHY_TST_CTRL0.PHY_TESTCLK.

Bits 15:8 – PHY_TESTDOUT[7:0] D-PHY Test Interface Data Out
PHY output 8-bit data bus for read-back and internal probing functionalities.

Bits 7:0 – PHY_TESTDIN[7:0] D-PHY Test Interface Data In
PHY test interface input 8-bit data bus for internal register programming and test functionalities access.

47.7.42. DSI Interrupt Status Register 0

Name: DSI_INT_ST0
Offset: 0xBC
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_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_0
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_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
	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 20 – DPHY_ERRORS_4 D-PHY Errors 4
Indicates LP1 contention error ErrContentionLP1 from Lane 0.

Bit 19 – DPHY_ERRORS_3 D-PHY Errors 3
Indicates LP0 contention error ErrContentionLP0 from Lane 0.

Bit 18 – DPHY_ERRORS_2 D-PHY Errors 2
Indicates control error ErrControl from Lane 0.

Bit 17 – DPHY_ERRORS_1 D-PHY Errors 1
Indicates ErrSyncEsc low-power data transmission synchronization error from Lane 0.

Bit 16 – DPHY_ERRORS_0 D-PHY Errors 0
This bit indicates ErrEsc escape entry error from Lane 0.

Bit 15 – ACK_WITH_ERR_15 Acknowledge With Error 15
Retrieves the DSI protocol violation from the Acknowledge error report.

Bit 14 – ACK_WITH_ERR_14 Acknowledge With Error 14
Retrieves the reserved (specific to device) from the Acknowledge error report.

Bit 13 – ACK_WITH_ERR_13 Acknowledge With Error 13
Retrieves the invalid transmission length from the Acknowledge error report.

Bit 12 – ACK_WITH_ERR_12 Acknowledge With Error 12
Retrieves the DSI VC ID Invalid from the Acknowledge error report.

Bit 11 – ACK_WITH_ERR_11 Acknowledge With Error 11
Retrieves the not recognized DSI data type from the Acknowledge error report.

Bit 10 – ACK_WITH_ERR_10 Acknowledge With Error 10
Retrieves the checksum error (long packet only) from the Acknowledge error report.

Bit 9 – ACK_WITH_ERR_9 Acknowledge With Error 9
Retrieves the ECC error, multi-bit (detected, not corrected) from the Acknowledge error report.

Bit 8 – ACK_WITH_ERR_8 Acknowledge With Error 8
Retrieves the ECC error, single-bit (detected and corrected) from the Acknowledge error report.

Bit 7 – ACK_WITH_ERR_7 Acknowledge With Error 7
Retrieves the reserved (specific to device) from the acknowledge error report.

Bit 6 – ACK_WITH_ERR_6 Acknowledge With Error 6
Retrieves the False Control error from the Acknowledge error report.

Bit 5 – ACK_WITH_ERR_5 Acknowledge With Error 5
Retrieves the Peripheral Timeout error from the Acknowledge error report.

Bit 4 – ACK_WITH_ERR_4 Acknowledge With Error 4
Retrieves the LP Transmit Sync error from the Acknowledge error report.

Bit 3 – ACK_WITH_ERR_3 Acknowledge With Error 3
Retrieves the Escape Mode Entry Command error from the Acknowledge error report.

Bit 2 – ACK_WITH_ERR_2 Acknowledge With Error 2
Retrieves the EoT Sync error from the Acknowledge error report.

Bit 1 – ACK_WITH_ERR_1 Acknowledge With Error 1
Retrieves the SoT Sync error from the Acknowledge error report.

Bit 0 – ACK_WITH_ERR_0 Acknowledge With Error 0
Retrieves the SoT error from the Acknowledge error report.

47.7.43. DSI Interrupt Status Register 1

Name: DSI_INT_ST1
Offset: 0xC0
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
					DPI_BUFF_PLD_UNDER			
Access					R			
Reset					0			

Bit	15	14	13	12	11	10	9	8
				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 19 – DPI_BUFF_PLD_UNDER Input Video Interface Payload Buffer Underflow Status
Indicates that an underflow has occurred when reading the payload to build a DSI packet for Video mode.

Bit 12 – GEN_PLD_RECEV_ERR Generic Interface Payload Receive Error Status
Indicates that during a generic interface packet read back, the payload FIFO becomes full and the received data is corrupted.

Bit 11 – GEN_PLD_RD_ERR Generic Interface DCS Payload Read Error Status
Indicates that during a DCS read data, the payload FIFO becomes empty and the data sent to the interface is corrupted.

Bit 10 – GEN_PLD_SEND_ERR Generic Interface Payload Send Error Status
Indicates that during a generic interface packet build, the payload FIFO becomes empty and corrupt data is sent.

Bit 9 – GEN_PLD_WR_ERR Generic Interface Payload Write Error Status
Indicates that the system tried to write a payload data through the Generic interface and the FIFO is full. Therefore, the payload is not written.

Bit 8 – GEN_CMD_WR_ERR Generic Interface Command Write Error Status
Indicates that the system tried to write a command through the Generic interface and the FIFO is full. Therefore, the command is not written.

Bit 7 – DPI_PLD_WR_ERR Input Video Payload Write Error Status
Indicates that during a DPI pixel line storage, the payload FIFO becomes full and the data stored is corrupted.

Bit 6 – EOTP_ERR End Of Transmission Packet Error Status
Indicates that the EoTp packet has not been received at the end of the incoming peripheral transmission.

Bit 5 – PKT_SIZE_ERR Packet Size Error Status
Indicates that the packet size error has been detected during the packet reception.

Bit 4 – CRC_ERR CRC Error Status
Indicates that the CRC error has been detected in the received packet payload.

Bit 3 – ECC_MULTI_ERR ECC Multiple Error Status
Indicates that the ECC multiple error has been detected in a received packet.

Bit 2 – ECC_SINGLE_ERR ECC Single Error Status
Indicates that the ECC single error has been detected and corrected in a received packet.

Bit 1 – TO_LP_RX Low-Power Reception Timeout Status
Indicates that the low-power reception timeout counter reached the end and contention has been detected.

Bit 0 – TO_HS_TX High-Speed Transmission Timeout Status
Indicates that the high-speed transmission timeout counter reached the end and contention has been detected.

47.7.44. DSI Interrupt Mask Configuration Register 0

Name: DSI_INT_MSK0
Offset: 0xC4
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt source is masked.

1: The corresponding interrupt source is activated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				DPHY_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_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
	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_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
	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
Access	R/W	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 – DPHY_ERRORS_4 D-PHY Errors Lane 4 Mask

Bit 19 – DPHY_ERRORS_3 D-PHY Errors Lane 3 Mask

Bit 18 – DPHY_ERRORS_2 D-PHY Errors Lane 2 Mask

Bit 17 – DPHY_ERRORS_1 D-PHY Errors Lane 1 Mask

Bit 16 – DPHY_ERRORS_0 D-PHY Errors Lane 0 Mask

Bit 15 – ACK_WITH_ERR_15 Acknowledge With Error 15 Mask

Bit 14 – ACK_WITH_ERR_14 Acknowledge With Error 14 Mask

Bit 13 – ACK_WITH_ERR_13 Acknowledge With Error 13 Mask

Bit 12 – ACK_WITH_ERR_12 Acknowledge With Error 12 Mask

Bit 11 – ACK_WITH_ERR_11 Acknowledge With Error 11 Mask

Bit 10 – ACK_WITH_ERR_10 Acknowledge With Error 10 Mask

Bit 9 – ACK_WITH_ERR_9 Acknowledge With Error 9 Mask

Bit 8 – ACK_WITH_ERR_8 Acknowledge With Error 8 Mask

Bit 7 – ACK_WITH_ERR_7 Acknowledge With Error 7 Mask

Bit 6 – ACK_WITH_ERR_6 Acknowledge With Error 6 Mask

Bit 5 – ACK_WITH_ERR_5 Acknowledge With Error 5 Mask

Bit 4 – ACK_WITH_ERR_4 Acknowledge With Error 4 Mask

Bit 3 – ACK_WITH_ERR_3 Acknowledge With Error 3 Mask

Bit 2 – ACK_WITH_ERR_2 Acknowledge With Error 2 Mask

Bit 1 – ACK_WITH_ERR_1 Acknowledge With Error 1 Mask

Bit 0 – ACK_WITH_ERR_0 Acknowledge With Error 0 Mask

47.7.45. DSI Interrupt Mask Configuration Register 1

Name: DSI_INT_MSK1
Offset: 0xC8
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt source is masked.

1: The corresponding interrupt source is activated.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
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
	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
Access	R/W	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 – DPI_BUFF_PLD_UNDER Input Video Interface Payload Buffer Underflow Mask

Bit 12 – GEN_PLD_RECEV_ERR Generic Interface Payload Receive Error Mask

Bit 11 – GEN_PLD_RD_ERR Generic Interface Payload Read Error Mask

Bit 10 – GEN_PLD_SEND_ERR Generic Interface Payload Send Error Mask

Bit 9 – GEN_PLD_WR_ERR Generic Interface Payload Write Error Mask

Bit 8 – GEN_CMD_WR_ERR Generic Interface Command Write Error Mask

Bit 7 – DPI_PLD_WR_ERR Input Video Payload Write Error Mask

Bit 6 – EOTP_ERR End Of Transmission Packet Error Mask

Bit 5 – PKT_SIZE_ERR Packet Size Error Mask

Bit 4 – CRC_ERR CRC Error Mask

Bit 3 – ECC_MULTI_ERR ECC Multiple Error Mask

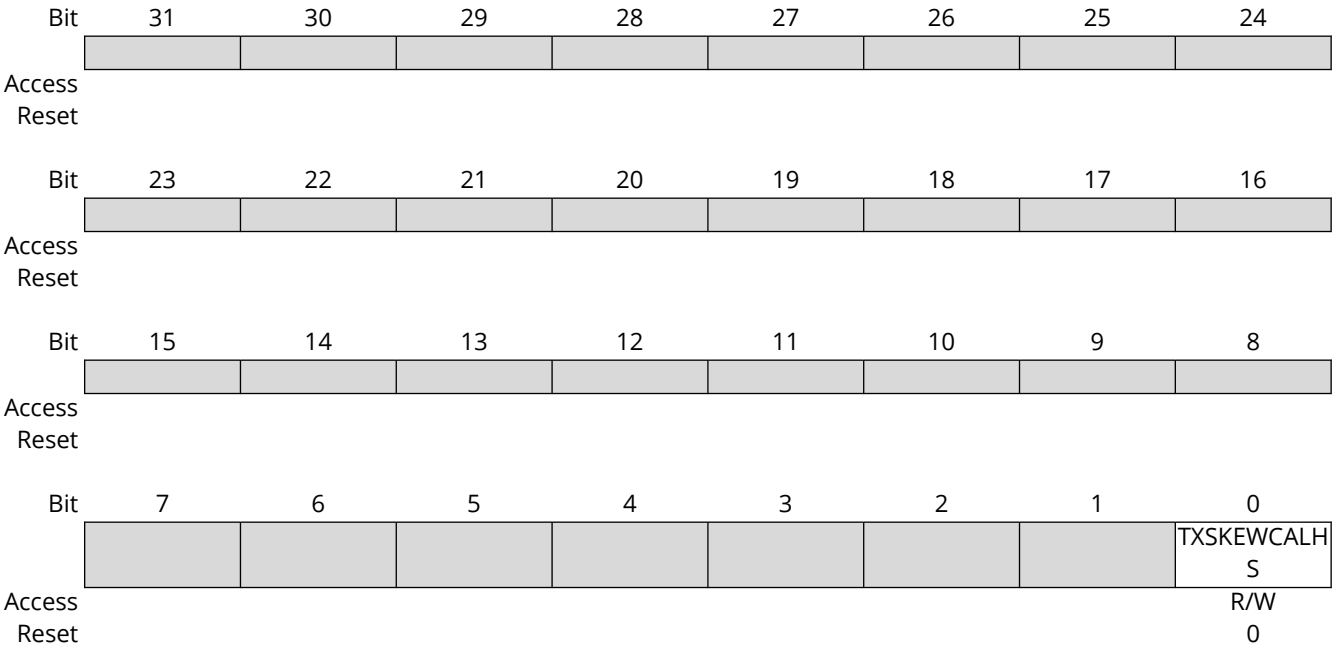
Bit 2 – ECC_SINGLE_ERR ECC Single Error Mask

Bit 1 – TO_LP_RX Timeout Low-Power Reception Mask

Bit 0 – TO_HS_TX Timeout High-Speed Transmission Mask

47.7.46. DSI D-PHY Calibration Control Register

Name: DSI_DPHY_CAL
Offset: 0xCC
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
								TXSKEWCALHS
Access								R/W
Reset								0

Bit 0 – TXSKEWCALHS Transmission Skew Calibration High-Speed
High-speed skew calibration is started when txskewcalhs is set to 1 (assuming that the D-PHY is in Stop state).

47.7.47. DSI Interrupt Force Control Register 0

Name: DSI_INT_FORCE0
Offset: 0xD8
Reset: 0x00000000
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Forces the triggering of the corresponding interrupt source.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
				DPHY_ERRORS_4	DPHY_ERRORS_3	DPHY_ERRORS_2	DPHY_ERRORS_1	DPHY_ERRORS_0
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ACK_WITH_ERR_15	ACK_WITH_ERR_14	ACK_WITH_ERR_13	ACK_WITH_ERR_12	ACK_WITH_ERR_11	ACK_WITH_ERR_10	ACK_WITH_ERR_9	ACK_WITH_ERR_8
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
	ACK_WITH_ERR_7	ACK_WITH_ERR_6	ACK_WITH_ERR_5	ACK_WITH_ERR_4	ACK_WITH_ERR_3	ACK_WITH_ERR_2	ACK_WITH_ERR_1	ACK_WITH_ERR_0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 20 – DPHY_ERRORS_4 D-PHY Errors 4 Force

Bit 19 – DPHY_ERRORS_3 D-PHY Errors 3 Force

Bit 18 – DPHY_ERRORS_2 D-PHY Errors 2 Force

Bit 17 – DPHY_ERRORS_1 D-PHY Errors 1 Force

Bit 16 – DPHY_ERRORS_0 D-PHY Errors 0 Force

Bit 15 – ACK_WITH_ERR_15 Acknowledge With Error 15 Force

Bit 14 – ACK_WITH_ERR_14 Acknowledge With Error 14 Force

Bit 13 – ACK_WITH_ERR_13 Acknowledge With Error 13 Force

Bit 12 – ACK_WITH_ERR_12 Acknowledge With Error 12 Force

Bit 11 – ACK_WITH_ERR_11 Acknowledge With Error 11 Force

Bit 10 – ACK_WITH_ERR_10 Acknowledge With Error 10 Force

Bit 9 – ACK_WITH_ERR_9 Acknowledge With Error 9 Force

Bit 8 – ACK_WITH_ERR_8 Acknowledge With Error 8 Force

Bit 7 – ACK_WITH_ERR_7 Acknowledge With Error 7 Force

Bit 6 – ACK_WITH_ERR_6 Acknowledge With Error 6 Force

Bit 5 – ACK_WITH_ERR_5 Acknowledge With Error 5 Force

Bit 4 – ACK_WITH_ERR_4 Acknowledge With Error 4 Force

Bit 3 – ACK_WITH_ERR_3 Acknowledge With Error 3 Force

Bit 2 – ACK_WITH_ERR_2 Acknowledge With Error 2 Force

Bit 1 – ACK_WITH_ERR_1 Acknowledge With Error 1 Force

Bit 0 – ACK_WITH_ERR_0 Acknowledge With Error 0 Force

47.7.48. DSI Interrupt Force Control Register 1

Name: DSI_INT_FORCE1
Offset: 0xDC
Reset: 0x00000000
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Forces the triggering of the corresponding interrupt source.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					DPI_BUFF_PLD_UNDER			
Access					W			
Reset					0			

Bit	15	14	13	12	11	10	9	8
				GEN_PLD_RECEV_ERR	GEN_PLD_RD_ERR	GEN_PLD_SEND_ERR	GEN_PLD_WR_ERR	GEN_CMD_WR_ERR
Access				W	W	W	W	W
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	DPI_PLD_WR_ERR	EOTP_ERR	PKT_SIZE_ERR	CRC_ERR	ECC_MULTI_ERR	ECC_SINGLE_ERR	TO_LP_RX	TO_HS_TX
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 19 – DPI_BUFF_PLD_UNDER Input Video Interface Payload Buffer Underflow Force

Bit 12 – GEN_PLD_RECEV_ERR Generic Interface Payload Receive Error Force

Bit 11 – GEN_PLD_RD_ERR Generic Interface Payload Read Error Force

Bit 10 – GEN_PLD_SEND_ERR Generic Interface Payload Send Error Force

Bit 9 – GEN_PLD_WR_ERR Generic Interface Payload Write Error Force

Bit 8 – GEN_CMD_WR_ERR Generic Interface Command Write Error Force

Bit 7 – DPI_PLD_WR_ERR Input Video Interface Payload Write Error Force

Bit 6 – EOTP_ERR End Of Transmission Packet Error Force

Bit 5 – PKT_SIZE_ERR Packet Size error Force

Bit 4 – CRC_ERR CRC Error Force

Bit 3 – ECC_MULTI_ERR ECC Multiple Error Force

Bit 2 – ECC_SINGLE_ERR ECC Single Error Force

Bit 1 – TO_LP_RX Timeout Low-Power Reception Force

Bit 0 – TO_HS_TX Timeout High-Speed Transmission Force

47.7.49. DSI D-PHY Read Timing Configuration Register

Name: DSI_DPHY_TMR_RD_CFG
Offset: 0xF4
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
		MAX_RD_TIME[14:8]						
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
	MAX_RD_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 14:0 – MAX_RD_TIME[14:0] Maximum Read Time
Configures the maximum time required to perform a read command in lane byte clock cycles. This field can only be modified when no read command is in progress.

47.7.50. DSI Video Mode Shadow Feature Control Register

Name: DSI_VID_SHADOW_CTRL
Offset: 0x100
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
								VID_SHADOW_REQ
Access								R/W
Reset								0

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

Bit 8 – VID_SHADOW_REQ Video Interface Shadow Request

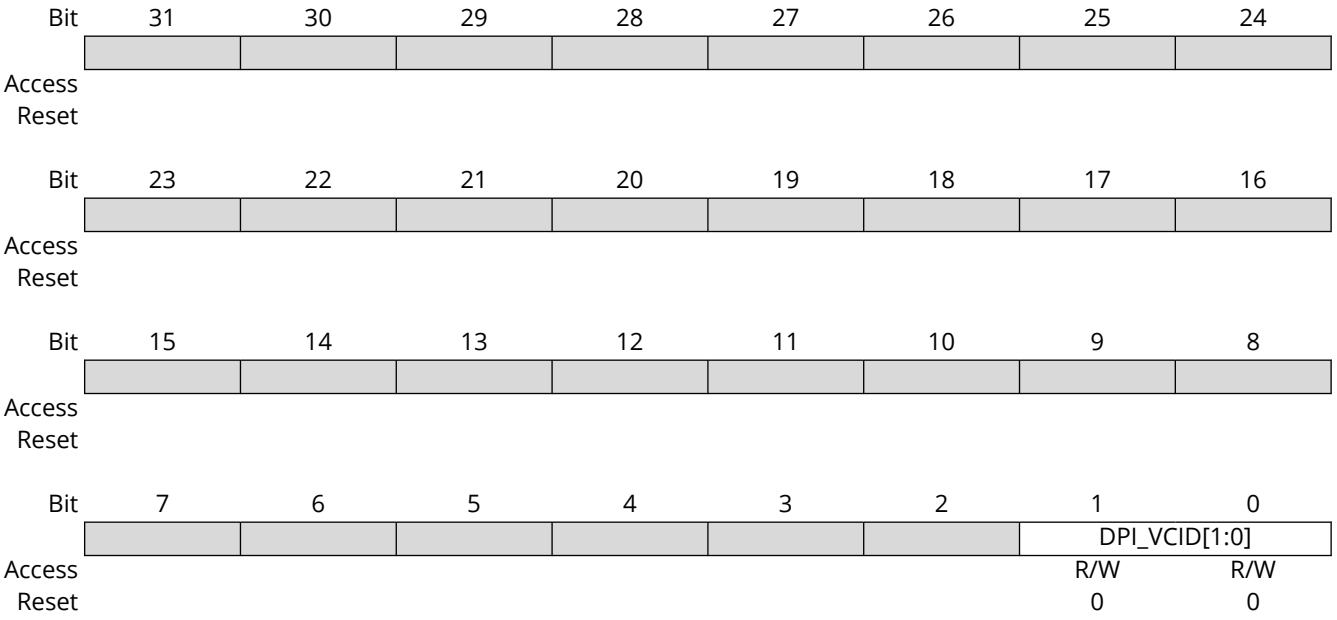
Value	Description
0	No effect.
1	Requests that the input video interface registers from regbank are copied to the auxiliary registers. When the request is completed, this bit is automatically cleared.

Bit 0 – VID_SHADOW_EN Video Interface Shadow Registers Enable

Value	Description
0	No effect.
1	Input video bus receives the active configuration from the auxiliary registers. When the feature is set at the same time as VID_SHADOW_REQ, the auxiliary registers are automatically updated.

47.7.51. DSI Input Video Virtual Channel ID Active Value Register

Name: DSI_DPI_VCID_ACT
Offset: 0x10C
Reset: 0x00000000
Property: Read-only



Bits 1:0 – DPI_VCID[1:0] Input Video Interface Virtual Channel Identifier
Specifies the input video virtual channel ID that is indexed to the Video mode packets.

47.7.52. DSI Input Video Color Coding Active Value Register

Name: DSI_DPI_COLOR_CODING_ACT
Offset: 0x110
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
								LOOSELY18_EN
Access								R
Reset								0

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

Bit 8 – LOOSELY18_EN Loosely Packed Variant to 18-bit Enable

Value	Description
0	Disables loosely packed variant to 18-bit configurations.
1	Enables loosely packed variant to 18-bit configurations.

Bits 3:0 – DPI_COLOR_CODING[3:0] Input Video Interface Color Coding
Specifies the DPI color coding.

Value	Name	Description
0	16BIT_CFG1	16-bit configuration 1
1	16BIT_CFG2	16-bit configuration 2
2	16BIT_CFG3	16-bit configuration 3
3	18BIT_CFG1	18-bit configuration 1
4	18BIT_CFG2	18-bit configuration 2
5	24BIT	24-bit
6	–	Reserved
7	–	Reserved
8	–	Reserved
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved

47.7.53. DSI Input Video Timing for Low-Power Commands Active Value Register

Name: DSI_DPI_LP_CMD_TIM_ACT
Offset: 0x118
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
	OUTVACT_LPCMD_TIME[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
	INVACT_LPCMD_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – OUTVACT_LPCMD_TIME[7:0] Output Vertical Active Low-Power Command Time
Used for the transmission of commands in Low-Power mode. It specifies the size, in bytes, of the largest packet that can fit in a line during the VSA, VBP, and VFP regions.

Bits 7:0 – INVACT_LPCMD_TIME[7:0] Input Vertical Active Low-Power Command Time
Used for the transmission of commands in Low-Power mode. It specifies the size, in bytes, of the largest packet that can fit in a line during the VACT region.

47.7.54. DSI Video Mode Active Value Register

Name: DSI_VID_MODE_CFG_ACT
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							LP_CMD_EN	FRAME_BTA_ACK_EN
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LP_HFP_EN	LP_HBP_EN	LP_VACT_EN	LP_VFP_EN	LP_VBP_EN	LP_VSA_EN	VID_MODE_TYPE[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 9 – LP_CMD_EN Low-Power Command Enable

Value	Description
0	Disables the command transmission only in Low-Power mode.
1	Enables the command transmission only in Low-Power mode.

Bit 8 – FRAME_BTA_ACK_EN Frame Bus Turn Around Acknowledge Enable

Value	Description
0	Disables the request for an acknowledge response at the end of a frame.
1	Enables the request for an acknowledge response at the end of a frame.

Bit 7 – LP_HFP_EN Low-Power Horizontal Front Proch Enable

Value	Description
0	Disables return to low-power inside the HFP period when timing allows.
1	Enables the return to low-power inside the HFP period when timing allows.

Bit 6 – LP_HBP_EN Low-Power Horizontal Back Proch Enable

Value	Description
0	Disables the return to low-power inside the HBP period when timing allows.
1	Enables the return to low-power inside the HBP period when timing allows.

Bit 5 – LP_VACT_EN Low-Power Vertical Active Enable

Value	Description
0	Disables the return to low-power inside the VACT period when timing allows.
1	Enables the return to low-power inside the VACT period when timing allows.

Bit 4 – LP_VFP_EN Low-Power Vertical Front Porch Enable

Value	Description
0	Disables the return to low-power inside the VFP period when timing allows.
1	Enables the return to low-power inside the VFP period when timing allows.

Bit 3 – LP_VBP_EN Low-Power Vertical Back Porch Enable

Value	Description
0	Disables the return to low-power inside the VBP period when timing allows.
1	Enables the return to low-power inside the VBP period when timing allows.

Bit 2 – LP_VSA_EN Low-Power Vertical Sync Active Enable

Value	Description
0	Disables the return to low-power inside the VSA period when timing allows.
1	Enables the return to low-power inside the VSA period when timing allows.

Bits 1:0 – VID_MODE_TYPE[1:0] Video Mode Type
Specifies the video mode transmission type.

Value	Description
0	Non-burst with sync pulses
1	Non-burst with sync events
2	Burst mode
3	Burst mode

47.7.55. DSI Video Mode Packet Size Active Value Register

Name: DSI_VID_PKT_SIZE_ACT
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
Access								
Reset								

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

Bits 13:0 – VID_PKT_SIZE[13:0] Video Mode Packet Size
Specifies the number of pixels in a single video packet. For 18-bit , this number must be a multiple of 4 , as described in the DSI specification.

47.7.56. DSI Video Mode Number of Chunks Active Value Register

Name: DSI_VID_NUM_CHUNKS_ACT
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								

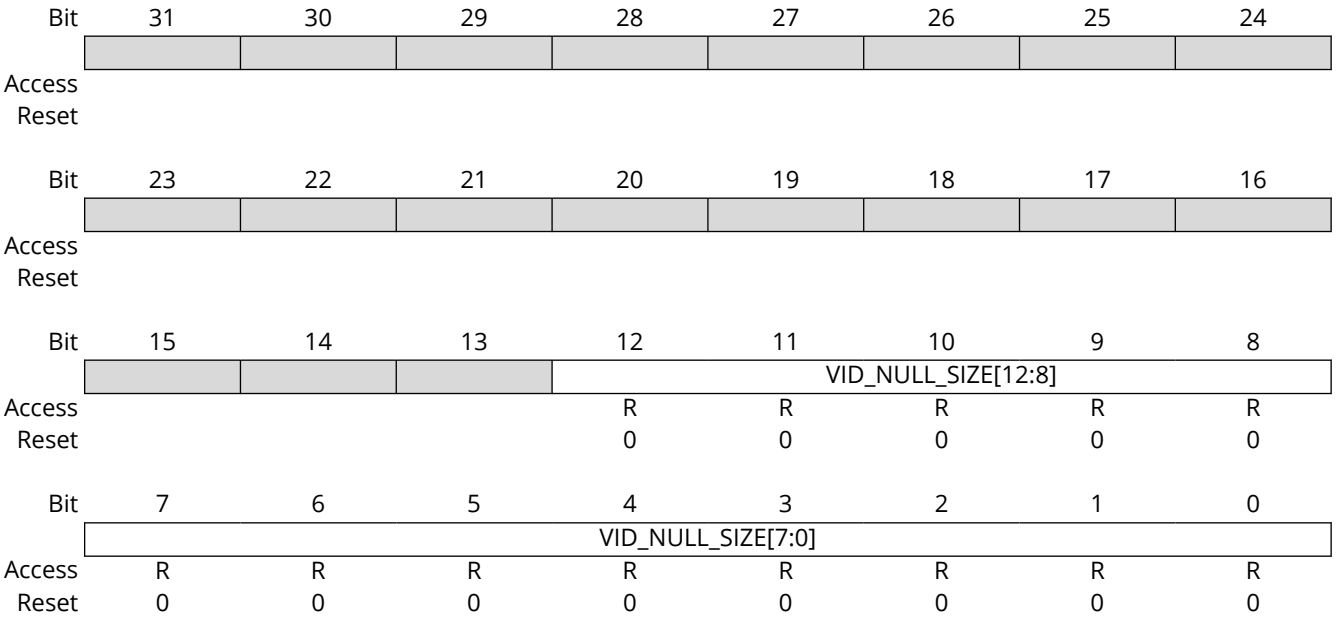
Bit	15	14	13	12	11	10	9	8
				VID_NUM_CHUNKS[12:8]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

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

Bits 12:0 – VID_NUM_CHUNKS[12:0] Video Mode Number of Chunks
Specifies the number of chunks to be transmitted during a line period (a chunk is pair made of a video packet and a null packet). If set to 0 or 1, the video line is still transmitted in a single packet. If set to 1, that packet is part of a chunk, meaning that a null packet follows it (if vid_null_size>0). Otherwise, multiple chunks are used to transmit each video line.

47.7.57. DSI Video Mode Null Packets Size Active Value Register

Name: DSI_VID_NULL_SIZE_ACT
Offset: 0x144
Reset: 0x00000000
Property: Read-only



Bits 12:0 – VID_NULL_SIZE[12:0] Video Mode Null Packet Size
Specifies the number of bytes inside a null packet. Setting to 0 disables null packets.

47.7.58. DSI Video Mode HSA Time Active Value Register

Name: DSI_VID_HSA_TIME_ACT
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
Access								
Reset								

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

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

Bits 11:0 – VID_HSA_TIME[11:0] Video Mode Horizontal Sync Active Time
Specifies the horizontal synchronism active period in lane byte clock cycles.

47.7.59. DIS Video Mode HBP Time Active Value Register

Name: DSI_VID_HBP_TIME_ACT
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
					VID_HBP_TIME[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

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

Bits 11:0 – VID_HBP_TIME[11:0] Video Mode Horizontal Back Porch Time
Specifies the horizontal back porch period in lane byte clock cycles.

47.7.60. DSI Video Mode HLINE Time Active Value Register

Name: DSI_VID_HLINE_TIME_ACT
Offset: 0x150
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
		VID_HLINE_TIME[14:8]						
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
	VID_HLINE_TIME[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 14:0 – VID_HLINE_TIME[14:0] Video Mode Horizontal Line Time
Specifies the size of the total line time (HSA+HBP+HACT+HFP) counted in lane byte clock cycles.

47.7.61. DSI Video Mode VSA Lines Active Value Register

Name: DSI_VID_VSA_LINES_ACT
Offset: 0x154
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
							VSA_LINES[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – VSA_LINES[9:0] Video Mode Vertical Sync Active Lines
Specifies the vertical synchronism active period measured in number of horizontal lines.

47.7.62. DSI Video Mode VBP Lines Active Value Register

Name: DSI_VID_VBP_LINES_ACT
Offset: 0x158
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
							VBP_LINES[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – VBP_LINES[9:0] Video Mode Vertical Back Porch Lines
Specifies the vertical back porch period measured in number of horizontal lines.

47.7.63. DSI Video Mode VFP Lines Active Value Register

Name: DSI_VID_VFP_LINES_ACT
Offset: 0x15C
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
							VFP_LINES[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – VFP_LINES[9:0] Video Mode Vertical Front Porch Lines
Specifies the vertical front porch period measured in number of horizontal lines.

47.7.64. DSI Video Mode VACTIVE Lines Active Value Register

Name: DSI_VID_VACTIVE_LINES_ACT
Offset: 0x160
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
			V_ACTIVE_LINES[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
	V_ACTIVE_LINES[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 13:0 – V_ACTIVE_LINES[13:0] Video Mode Vertical Active Lines
Specifies the vertical active period measured in number of horizontal lines.

47.7.65. DSI Video Mode Packet Status Register

Name: DSI_VID_PKT_STATUS
Offset: 0x168
Reset: 0x00010005
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
							DPI_BUFF_PLD_FULL	DPI_BUFF_PLD_EMPTY
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
					DPI_PLD_W_FULL	DPI_PLD_W_EMPTY	DPI_CMD_W_FULL	DPI_CMD_W_EMPTY
Access					R	R	R	R
Reset					0	1	0	1

Bit 17 – DPI_BUFF_PLD_FULL Input Video Interface Write Payload FIFO Full
Indicates the full status of the payload internal buffer for Video mode.

Bit 16 – DPI_BUFF_PLD_EMPTY Input Video Interface Write Payload FIFO Empty
Indicates the empty status of the payload internal buffer for Video mode.

Bit 3 – DPI_PLD_W_FULL Input Video Interface Write Payload FIFO Full
Indicates the full status of the write payload FIFO for Video mode.

Bit 2 – DPI_PLD_W_EMPTY Input Video Interface Write Payload FIFO Empty
Indicates the empty status of the write payload FIFO for Video mode.

Bit 1 – DPI_CMD_W_FULL Input Video Interface Command FIFO Full
Indicates the full status of the write command FIFO for Video mode.

Bit 0 – DPI_CMD_W_EMPTY Input Video Interface Command FIFO Empty
Indicates the empty status of the write command FIFO for Video mode.

47.7.66. DSI VSS Packet 3D Active Value Register

Name: DSI_SDF_3D_ACT
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
								SEND_3D_CFG
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			RIGHT_FIRST	SECOND_VSYNC	FORMAT_3D[1:0]		MODE_3D[1:0]	
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 16 – SEND_3D_CFG Send 3D Configuration Enable
When set, causes the next VSS packet to include 3D control payload in every VSS packet.

Bit 5 – RIGHT_FIRST Right First
Specifies the left/right order:

Value	Description
0	Left eye is sent first, then right eye.
1	Right eye data is sent first, then left eye.

Bit 4 – SECOND_VSYNC Second Vertical Sync Enable
Specifies whether there is a second VSYNC pulse between Left and Right Images, when 3D Image Format is Frame-based:

Value	Description
0	No sync pulses between left and right data.
1	Sync pulse (HSYNC, VSYNC, blanking) between left and right data.

Bits 3:2 – FORMAT_3D[1:0] 3D Format
Specifies 3D image format:

Value	Description
0	Line (alternating lines of left and right data).
1	Frame (alternating frames of left and right data).
2	Pixel (alternating pixels of left and right data).
3	Reserved

Bits 1:0 – MODE_3D[1:0] 3D Mode
Specifies 3D Mode on/off and display orientation:

Value	Description
0	3D Mode off (2D Mode on).
1	3D Mode on, portrait orientation.
2	3D Mode on, landscape orientation.
3	Reserved.

48. Camera Serial Interface (CSI)

48.1. 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.

48.2. 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:

- Four-lane Receive MIPI D-PHY Compliant with Revision 1.1 Specification
- Lane Operation Ranging from 80 Mbps to 1 Gbps in Forward Direction
- Aggregate Throughput up to 4 Gbps with Four 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

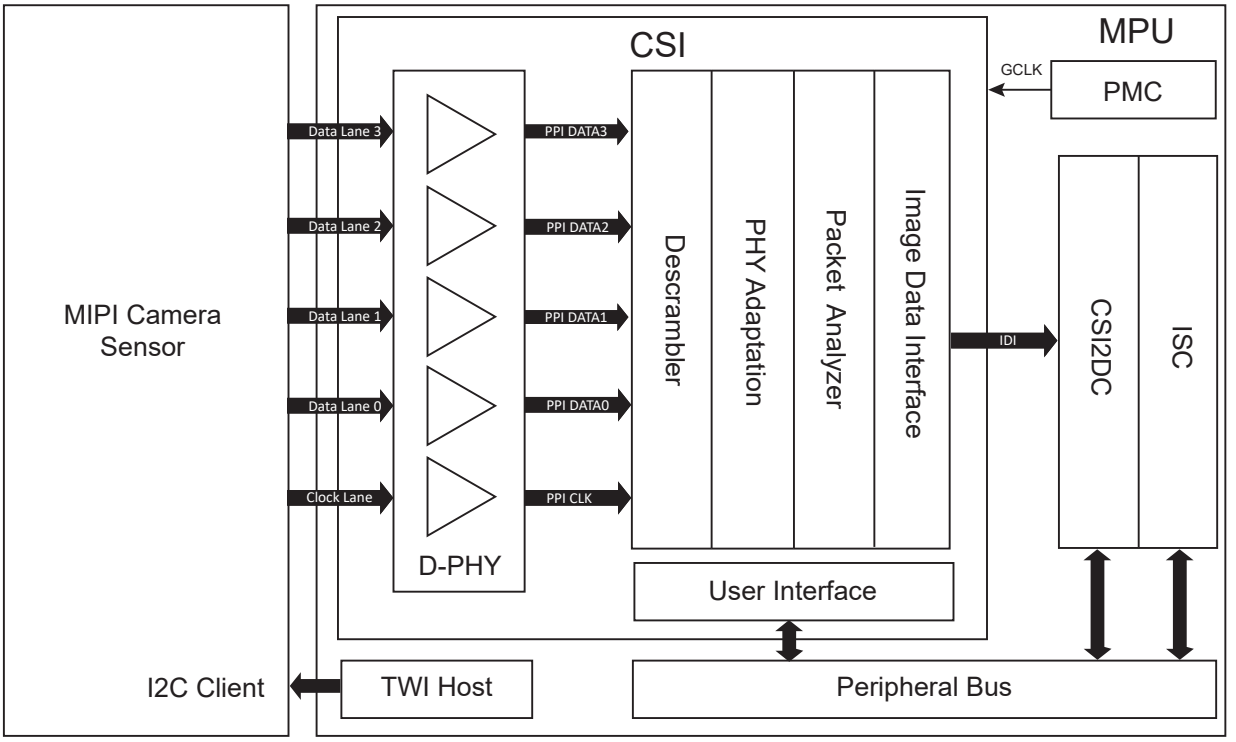
48.3. I/O Lines Description

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

I/O Lines Description (continued)		
Signal Name	**Type**	**Function**
MIPI_DN1	Input	Negative D-PHY differential data line receiver, Lane 1
MIPI_DP2	Input	Positive D-PHY differential data line receiver, Lane 2
MIPI_DN2	Input	Negative D-PHY differential data line receiver, Lane 2
MIPI_DP3	Input	Positive D-PHY differential data line receiver, Lane 3
MIPI_DN3	Input	Negative D-PHY differential data line receiver, Lane 3
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

48.4. Block Diagram

Figure 48.1. CSI Block Diagram



48.5. Product Dependencies

48.5.1. I/O Lines

The D-PHY can be used either by DSI or CSI hosts. The programmer must first configure a dedicated Special Function Register, SFR_ISS_CFG, to connect D-PHY to CSI (refer to the section "Special Function Register (SFR)").

48.5.2. Power Management

The CSI is not continuously clocked. Before using it, the programmer must first enable the CSI and the MIPI D-PHY peripheral clocks in the Power Management Controller (PMC).

48.5.3. 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.

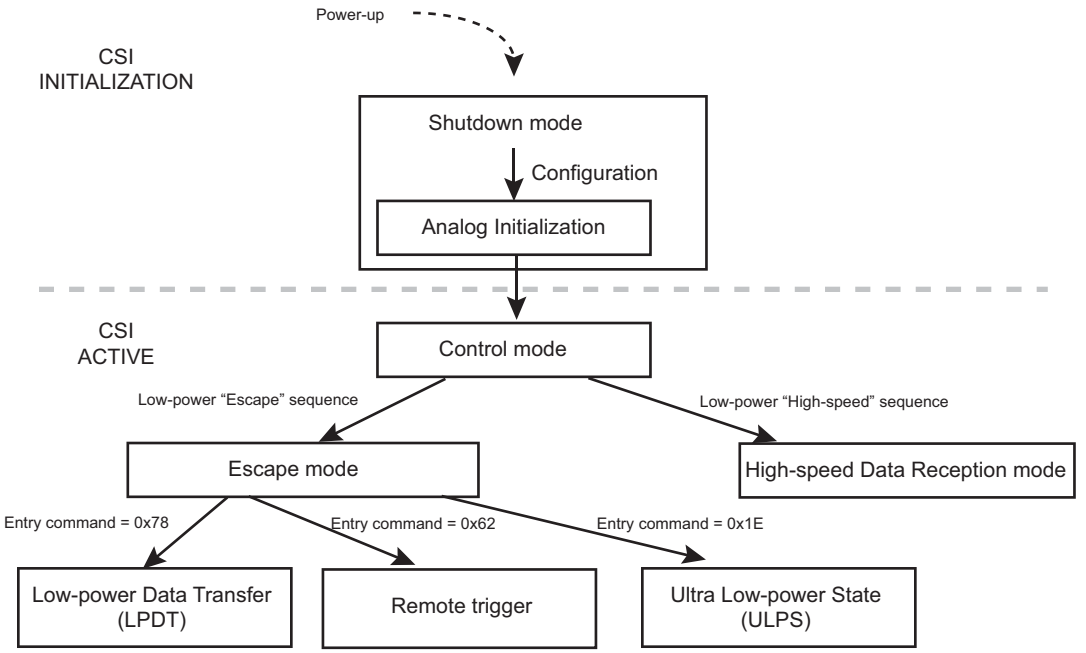
48.6. Functional Description

The CSI interfaces the Image Sensor Controller (ISC) and a MIPI CSI-2 compliant Camera Sensor.

48.6.1. D-PHY Operating Modes

The following figure illustrates the D-PHY initialization from Shutdown to Active modes.

Figure 48.2. D-PHY Initialization from Shutdown to Active Modes



48.6.1.1. Power-Up Mode

The power-up sequence is as follows:

1. Core voltage (VDDCORE) power-up
2. I/O voltage (VDDDPHY) power-up

48.6.1.2. 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 Reset register (CSI_DPHY_RSTZ) and PHY_SHUTDOWNZ in the Shutdown register (CSI_PHY_SHUTDOWNZ). The bit PHY_TESTCLR in the Analog Configuration Control register (CSI_PHY_TEST_CTRL0) is asserted by default.

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 must be updated while in Control mode.

When CSI_DPHY_RSTZ.DPHY_RSTZ and CSI_PHY_SHUTDOWNZ.PHY_SHUTDOWNZ are set to '1', the D-PHY exits Shutdown mode and starts an initialization procedure.

48.6.1.3. Analog Initialization

The D-PHY comprises 4 data lanes, but some applications may require fewer. 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_PHY_TEST_CTRL0. PHY_TESTCLR.
3. Write a '1' to CSI_PHY_TEST_CTRL0.PHY_TESTCLK.
4. Write 0x44 to CSI_PHY_TEST_CTRL1.PHY_TESTDIN and write a '1' to CSI_PHY_TEST_CTRL1.PHY_TESTEN.
5. Write a '0' to CSI_PHY_TEST_CTRL0.PHY_TESTCLK to create a falling edge on PHY_TESTCLK.
6. Write a '0' to CSI_PHY_TEST_CTRL1.PHY_TESTEN and write the configuration value from the following table to CSI_PHY_TEST_CTRL1.PHY_TESTDIN.
7. Write a high pulse to CSI_PHY_TEST_CTRL0.PHY_TESTCLK by writing '1' immediately followed by '0'.

Table 48.1. 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
950-1000	011010

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.

48.6.1.4.Active Modes

48.6.1.4.1.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.

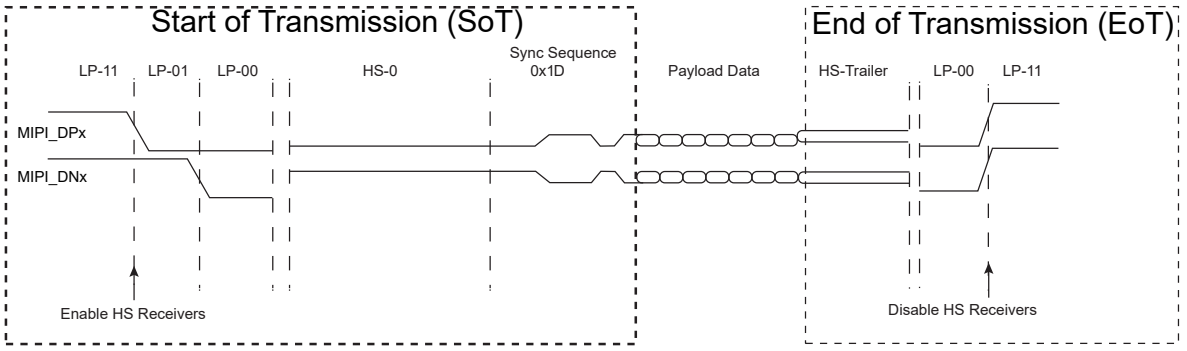
48.6.1.4.2.High-Speed Data Reception Mode

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 48.3. 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.

48.6.1.4.3.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 PHY_ERRESC_x error flag goes high, and the receiver waits until the transmitter returns to the Stop state.

Table 48.2. 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.

48.6.2. 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 500 MHz, which results in a bandwidth of 500 Mbps to 1 Gbps for each data lane.
- No significant control/reserved traffic is present on the link when pixel data is being transmitted.

Table 48.3. 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
Camera Formats						
2 Mpixels	2560000	15	24	922	2	1
3 Mpixels	3840000	15	24	1382	3	2
5 Mpixels	5500000	15	24	1980	4	2
Video Formats						
1280x720 pixels (720p)	921600	30	24	664	2	1
1280x720 pixels (720p)	921600	60	24	1327	3	2

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.

48.6.3. 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 0/1/2/3 Scrambling Seed registers (CSI_SCRAMBLING_SEED1/2/3/4). The default seed values after reset of the controller are given in the table below.

Lane	Initial Seed Value
0	0x1008
1	0x1188
2	0x1248
3	0x1428

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_SEED1/2/3/4.

48.6.4. Interrupts

The Interrupt Status registers report error conditions.

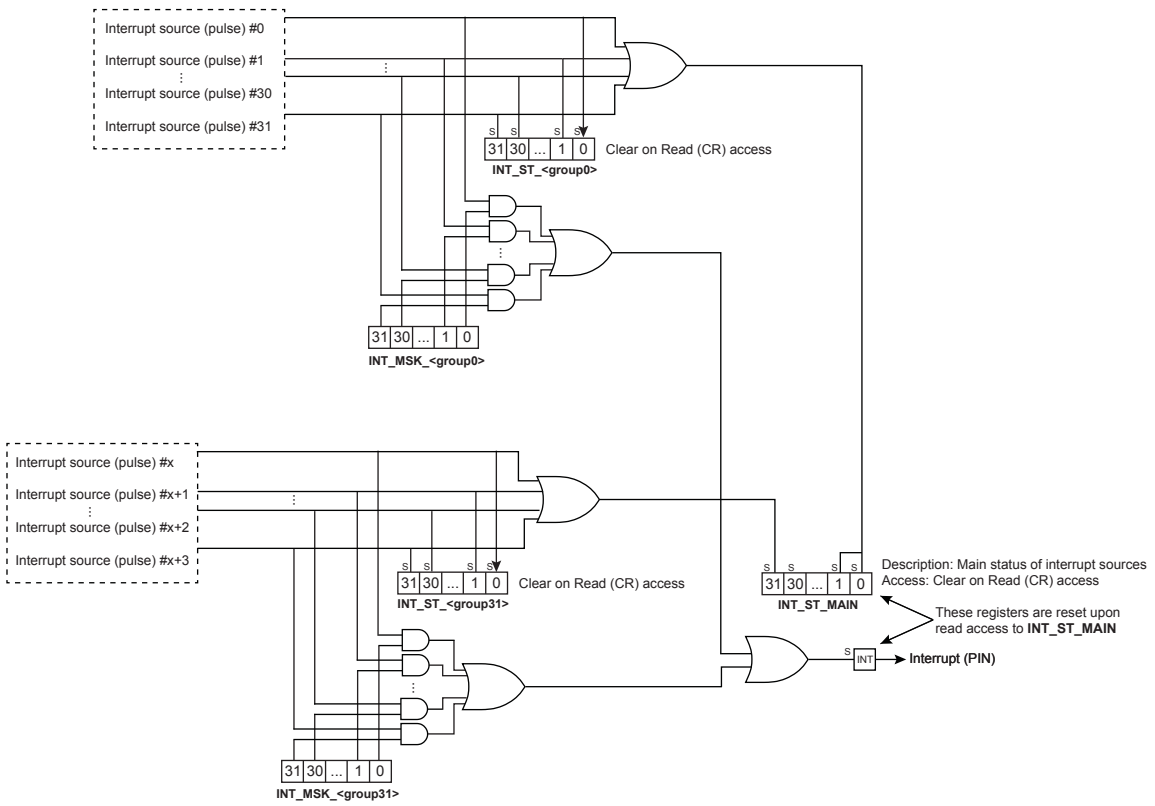
The triggering of the CSI interrupt line can be masked by programming the Interrupt Mask registers. By default, all errors are masked. When any bit of these registers is set to 1, it enables the interrupt for a specific error.

The Interrupt Status registers are cleared on read.

The Interrupt Force registers (CSI_INT_FORCE_<group>) are used for test purposes, and trigger interrupt events individually. Setting any bit of these registers to 1 triggers the corresponding interrupt.

The figure below shows the main parts of the interrupt mechanism.

Figure 48.4. Interrupt Mechanism



48.6.5. 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

48.7. 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[1:0]	
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_PKT	STATUS_INT_DPHY
		15:8								
		7:0						STATUS_INT_FRAME_FATAL	STATUS_INT_PKT_FATAL	STATUS_INT_PHY_FATAL
0x10 ... 0x3F	Reserved									
0x40	CSI_PHY_SHUTDOWNZ	31:24								
		23:16								
		15:8								
		7:0								PHY_SHUTDOWNZ
0x44	CSI_DPHY_RSTZ	31:24								
		23:16								
		15:8								
		7:0								DPHY_RSTZ
0x48	CSI_PHY_RX	31:24								
		23:16							PHY_RXCLKACTIVEHS	PHY_RXULPSSCLKNOT
		15:8								
		7:0					PHY_RXULPSESC_3	PHY_RXULPSESC_2	PHY_RXULPSESC_1	PHY_RXULPSESC_0
0x4C	CSI_PHY_STOPSTATE	31:24								
		23:16								PHY_STOPSTATECLK
		15:8								
		7:0					PHY_STOPSTADATA_3	PHY_STOPSTADATA_2	PHY_STOPSTADATA_1	PHY_STOPSTADATA_0
0x50	CSI_PHY_TEST_CTRL0	31:24								
		23:16								
		15:8								
		7:0							PHY_TESTCLK	PHY_TESTCLR

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x54	CSI_PHY_TEST_CTRL1	31:24								
		23:16								PHY_TESTEN
		15:8	PHY_TESTDOUT[7:0]							
		7:0	PHY_TESTDIN[7:0]							
0x58 ... 0xCB	Reserved									
0xCC	CSI_PHY_CAL	31:24								
		23:16								
		15:8								
		7:0								RXSKEWCALHS
0xD0 ... 0xDF	Reserved									
0xE0	CSI_INT_ST_PHY_FATAL	31:24								
		23:16								
		15:8								
		7:0					PHY_ERRSOTSYNCHS_3	PHY_ERRSOTSYNCHS_2	PHY_ERRSOTSYNCHS_1	PHY_ERRSOTSYNCHS_0
0xE4	CSI_INT_MSK_PHY_FATAL	31:24								
		23:16								
		15:8								
		7:0					MASK_ERRSOTSYNCHS_3	MASK_ERRSOTSYNCHS_2	MASK_ERRSOTSYNCHS_1	MASK_ERRSOTSYNCHS_0
0xE8	CSI_INT_FORCE_PHY_FATAL	31:24								
		23:16								
		15:8								
		7:0					FORCE_ERRSOTSYNCHS_3	FORCE_ERRSOTSYNCHS_2	FORCE_ERRSOTSYNCHS_1	FORCE_ERRSOTSYNCHS_0
0xEC ... 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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xF8	CSI_INT_FORCE_PKT_FATAL	31:24								
		23:16								FORCE_ERR_ECC_DOUBLE
		15:8								
		7:0					FORCE_VC3_ERR_CRC	FORCE_VC2_ERR_CRC	FORCE_VC1_ERR_CRC	FORCE_VC0_ERR_CRC
0xFC ... 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
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	CSI_INT_FORCE_FRAME_FATAL	31:24								
		23:16					FORCE_ERR_FRAME_DATA_VC3	FORCE_ERR_FRAME_DATA_VC2	FORCE_ERR_FRAME_DATA_VC1	FORCE_ERR_FRAME_DATA_VC0
		15:8					FORCE_ERR_F_SEQ_VC3	FORCE_ERR_F_SEQ_VC2	FORCE_ERR_F_SEQ_VC1	FORCE_ERR_F_SEQ_VC0
		7:0					FORCE_ERR_F_BNDRY_MATCH_VC3	FORCE_ERR_F_BNDRY_MATCH_VC2	FORCE_ERR_F_BNDRY_MATCH_VC1	FORCE_ERR_F_BNDRY_MATCH_VC0
0x010C ... 0x010F	Reserved									
0x0110	CSI_INT_ST_PHY	31:24								
		23:16					PHY_ERRESC_3	PHY_ERRESC_2	PHY_ERRESC_1	PHY_ERRESC_0
		15:8								
		7:0					PHY_ERRSOTHS_3	PHY_ERRSOTHS_2	PHY_ERRSOTHS_1	PHY_ERRSOTHS_0
0x0114	CSI_INT_MSK_PHY	31:24								
		23:16					MASK_PHY_ERRESC_3	MASK_PHY_ERRESC_2	MASK_PHY_ERRESC_1	MASK_PHY_ERRESC_0
		15:8								
		7:0					MASK_PHY_ERRSOTHS_3	MASK_PHY_ERRSOTHS_2	MASK_PHY_ERRSOTHS_1	MASK_PHY_ERRSOTHS_0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0118	CSI_INT_FORCE_PHY	31:24								
		23:16					FORCE_PHY_ERRESC_3	FORCE_PHY_ERRESC_2	FORCE_PHY_ERRESC_1	FORCE_PHY_ERRESC_0
		15:8								
		7:0					FORCE_PHY_ERRSOTHS_3	FORCE_PHY_ERRSOTHS_2	FORCE_PHY_ERRSOTHS_1	FORCE_PHY_ERRSOTHS_0
0x011C ... 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	CSI_INT_FORCE_PKT	31:24								
		23:16					FORCE_VC3_ERR_ECC_CORRECTED	FORCE_VC2_ERR_ECC_CORRECTED	FORCE_VC1_ERR_ECC_CORRECTED	FORCE_VC0_ERR_ECC_CORRECTED
		15:8								
		7:0					FORCE_ERR_ID_VC3	FORCE_ERR_ID_VC2	FORCE_ERR_ID_VC1	FORCE_ERR_ID_VC0
0x012C ... 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]							
0x030C	CSI_SCRAMBLING_SEED2	31:24								
		23:16								
		15:8	SCRAMBLE_SEED_LANE2[15:8]							
		7:0	SCRAMBLE_SEED_LANE2[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0310	CSI_SCRAMBLING_SEED3	31:24								
		23:16								
		15:8	SCRAMBLE_SEED_LANE3[15:8]							
		7:0	SCRAMBLE_SEED_LANE3[7:0]							

48.7.1. 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[1:0]	
Access							R/W	R/W
Reset							0	1

Bits 1:0 – N_LANES[1:0] Number of active data lanes
The update can be performed only when the D-PHY is in Stop state.

Value	Name	Description
0	1_LANE	One data lane
1	2_LANES	Two data lanes
2	3_LANES	Three data lanes
3	4_LANES	Four data lanes

48.7.2. 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.

48.7.3. 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_PKT	STATUS_INT_DPHY
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
						STATUS_INT_FRAME_FATAL	STATUS_INT_PKT_FATAL	STATUS_INT_PHY_FATAL
Access						R	R	R
Reset						0	0	0

Bit 17 – STATUS_INT_PKT CSI_INT_ST_PKT Register Event (cleared on read)
Indicates if an event occurred in the CSI_INT_ST_PKT register.

Bit 16 – STATUS_INT_DPHY CSI_INT_ST_DPHY Register Event (cleared on read)
Indicates if an event occurred in the CSI_INT_ST_DPHY register.

Bit 2 – STATUS_INT_FRAME_FATAL CSI_INT_ST_FRAME_FATAL Register Event (cleared on read)
Indicates if an event occurred in the CSI_INT_ST_FRAME_FATAL register.

Bit 1 – STATUS_INT_PKT_FATAL CSI_INT_ST_PKT_FATAL Register Event (cleared on Read)
Indicates if an event occurred in the CSI_INT_ST_PKT_FATAL register.

Bit 0 – STATUS_INT_PHY_FATAL CSI_INT_ST_PHY_FATAL Register Event (cleared on read)
Indicates if an event occurred in the CSI_INT_ST_PHY_FATAL register.

48.7.4. CSI D-PHY Shutdown Register

Name: CSI_PHY_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
								PHY_SHUTDOWNZ
Access								R/W
Reset								0

Bit 0 – PHY_SHUTDOWNZ Shutdown Input Buffer

Value	Description
0	Puts the D-PHY in Shutdown mode. All analog blocks are in Power-down mode and digital logic is cleared.
1	D-PHY exits Shutdown mode.

48.7.5. 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.

48.7.6. CSI D-PHY Receive Status Register

Name: CSI_PHY_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
							PHY_RXCLKACTIVEHS	PHY_RXULPSCLKNOT
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
					PHY_RXULPSESC_3	PHY_RXULPSESC_2	PHY_RXULPSESC_1	PHY_RXULPSESC_0
Access					R	R	R	R
Reset					0	0	0	0

Bit 17 – PHY_RXCLKACTIVEHS D-PHY Receives a DDR Clock

Value	Description
0	No DDR clock received.
1	Indicates that D-PHY clock lane is actively receiving a DDR clock.

Bit 16 – PHY_RXULPSSCLKNOT Clock Lane Power Status

Value	Description
0	Indicates that D-PHY Clock Lane module has entered Ultra-Low-Power (ULP) mode.
1	Clock Lane is not in ULP mode.

Bit 3 – PHY_RXULPSESC_3 Lane 3 Ultra-Low-Power Status

Value	Description
0	Indicates that D-PHY Clock Lane module has entered ULP mode.
1	Clock Lane is not in ULP mode.

Bit 2 – PHY_RXULPSESC_2 Data Lane 2 Ultra-Low-Power Status

Value	Description
0	Data lane 2 module is not in ULP mode.
1	Data lane 2 module has entered ULP mode.

Bit 1 – PHY_RXULPSESC_1 Data Lane 1 Ultra-Low-Power Status

Value	Description
0	Data lane 1 module is not in ULP mode.
1	Data lane 1 module has entered ULP mode.

Bit 0 – PHY_RXULPSESC_0 Data Lane 0 Ultra-Low-Power Status

Value	Description
0	Data lane 0 module is not in ULP mode.
1	Data lane 0 module has entered ULP mode.

48.7.7. CSI D-PHY Stop State Register

Name: CSI_PHY_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
								PHY_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
					PHY_STOPSTADATA_3	PHY_STOPSTADATA_2	PHY_STOPSTADATA_1	PHY_STOPSTADATA_0
Access					R	R	R	R
Reset					0	0	0	0

Bit 16 – PHY_STOPSTATECLK Clock Lane Stop State Status

Value	Description
0	Clock lane module is not in Stop state.
1	Clock lane module is in Stop state.

Bit 3 – PHY_STOPSTADATA_3 Data Lane 3 Stop State Status

Value	Description
0	Data lane 3 module is not in Stop state.
1	Data lane 3 module has entered Stop state.

Bit 2 – PHY_STOPSTADATA_2 Data Lane 2 Stop State Status

Value	Description
0	Data lane 2 module is not in Stop state.
1	Data lane 2 module has entered Stop state.

Bit 1 – PHY_STOPSTADATA_1 Data Lane 1 Stop State Status

Value	Description
0	Data lane 1 module is not in Stop state.
1	Data lane 1 module has entered Stop state.

Bit 0 – PHY_STOPSTADATA_0 Data Lane 0 Stop State Status

Value	Description
0	Data lane 0 module is not in Stop state.
1	Data lane 0 module has entered Stop state.

48.7.8. CSI D-PHY Analog Configuration Control Register

Name: CSI_PHY_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
							PHY_TESTCLK	PHY_TESTCLR
Access							R/W	R/W
Reset							0	1

Bit 1 – PHY_TESTCLK Analog Configuration Control Clock
The data is loaded on one edge of PHY_TESTCLK, thus it is mandatory to write a '0' immediately after writing a '1'. Refer to PHY_TEST_CTRL1.PHY_TESTDEN.

Value	Description
0	No effect.
1	Captures the PHY_TEST_CTRL1.PHY_TESTDIN value.

Bit 0 – PHY_TESTCLR Analog Configuration Clear
The reset is performed on the rising edge of PHY_TESTCLR, thus it is mandatory to write a '0' immediately after writing a '1'.

Value	Description
0	No effect.
1	Resets the analog configuration.

48.7.9. CSI D-PHY Analog Configuration Data Register

Name: CSI_PHY_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
								PHY_TESTEN
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	PHY_TESTDOUT[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
	PHY_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 – PHY_TESTEN Analog Configuration Code Selection

Value	Description
0	Transmits the high-speed bit rate code on the rising edge of CSI_PHY_TEST_CTRL0.PHY_TESTCLK.
1	Transmits the address (0x44) of the high-speed bit rate code on the falling edge of CSI_PHY_TEST_CTRL0.PHY_TESTCLK.

Bits 15:8 – PHY_TESTDOUT[7:0] Read Data Output for Test
Data output for reading data and other probing functionalities.

Bits 7:0 – PHY_TESTDIN[7:0] Analog Configuration Value or High-Speed Bit Rate Code
Value selected by PHY_TESTEN.

48.7.10. CSI D-PHY Calibration Status Register

Name: CSI_PHY_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 Calibration Status (cleared on read)

Value	Description
0	No calibration initiated since the last CSI_PHY_CAL read.
1	A calibration has been initiated since the last CSI_PHY_CAL read.

48.7.11. CSI D-PHY Fatal Error Interrupt Status Register

Name: CSI_INT_ST_PHY_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_PHY_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
					PHY_ERRSOTSYNCHS_3	PHY_ERRSOTSYNCHS_2	PHY_ERRSOTSYNCHS_1	PHY_ERRSOTSYNCHS_0
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – PHY_ERRSOTSYNCHS_3 Data Lane 3 Start Of Transmission Error Status (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 3 since the last read of CSI_INT_ST_PHY_FATAL.
1	Transmission error has occurred on data lane 3 since the last read of CSI_INT_ST_PHY_FATAL.

Bit 2 – PHY_ERRSOTSYNCHS_2 Data Lane 2 Start Of Transmission Error Status (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 2 since the last read of CSI_INT_ST_PHY_FATAL.
1	Transmission error has occurred on data lane 2 since the last read of CSI_INT_ST_PHY_FATAL.

Bit 1 – PHY_ERRSOTSYNCHS_1 Data Lane 1 Start Of Transmission Error Status (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_PHY_FATAL.
1	Transmission error has occurred on data lane 1 since the last read of CSI_INT_ST_PHY_FATAL.

Bit 0 – PHY_ERRSOTSYNCHS_0 Data Lane 0 Start Of Transmission Error Status (cleared on read)

Value	Description
0	No transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_PHY_FATAL.
1	Transmission error has occurred on data lane 0 since the last read of CSI_INT_ST_PHY_FATAL.

48.7.12. CSI D-PHY Fatal Error Interrupt Mask Register

Name: CSI_INT_MSK_PHY_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_ERRSOTSYNCHS_3	MASK_ERRSOTSYNCHS_2	MASK_ERRSOTSYNCHS_1	MASK_ERRSOTSYNCHS_0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – MASK_ERRSOTSYNCHS_3 Data Lane 3 Start Of Transmission Error Interrupt Mask

Bit 2 – MASK_ERRSOTSYNCHS_2 Data Lane 2 Start Of Transmission Error Interrupt Mask

Bit 1 – MASK_ERRSOTSYNCHS_1 Data Lane 1 Start Of Transmission Error Interrupt Mask

Bit 0 – MASK_ERRSOTSYNCHS_0 Data Lane 0 Start Of Transmission Error Interrupt Mask

48.7.13. CSI D-PHY Fatal Error Interrupt Force Register

Name: CSI_INT_FORCE_PHY_FATAL
Offset: 0xE8
Reset: 0x00000000
Property: Read/Write

Used for test purposes. Triggers CSI_INT_ST_PHY_FATAL interrupt events individually without the need to activate the conditions that trigger the interrupt sources.

Bit	31	30	29	28	27	26	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
					FORCE_ERRSOTSYNCHS_3	FORCE_ERRSOTSYNCHS_2	FORCE_ERRSOTSYNCHS_1	FORCE_ERRSOTSYNCHS_0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – FORCE_ERRSOTSYNCHS_3 Force Start Of Transmission Interrupt Error on Data Lane 3

Bit 2 – FORCE_ERRSOTSYNCHS_2 Force Start Of Transmission Interrupt Error on Data Lane 2

Bit 1 – FORCE_ERRSOTSYNCHS_1 Force Start Of Transmission Interrupt Error on Data Lane 1

Bit 0 – FORCE_ERRSOTSYNCHS_0 Force Start Of Transmission Interrupt Error on Data Lane 0

48.7.14. 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
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bit 16 – ERR_ECC_DOUBLE Unrecoverable Header Error (ECC Two Errors) (cleared on read)

Bit 3 – VC3_ERR_CRC Virtual Channel 3 Payload Checksum Error (cleared on read)

Bit 2 – VC2_ERR_CRC Virtual Channel 2 Payload Checksum Error (cleared on read)

Bit 1 – VC1_ERR_CRC Virtual Channel 1 Payload Checksum Error (cleared on read)

Bit 0 – VC0_ERR_CRC Virtual Channel 0 Payload Checksum Error (cleared on read)

48.7.15. 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 Unrecoverable Header Error (ECC Two Errors) Interrupt Mask

Bit 3 – MASK_VC3_ERR_CRC Virtual Channel 3 Payload Checksum Error Interrupt Mask

Bit 2 – MASK_VC2_ERR_CRC Virtual Channel 2 Payload Checksum Error Interrupt Mask

Bit 1 – MASK_VC1_ERR_CRC Virtual Channel 1 Payload Checksum Error Interrupt Mask

Bit 0 – MASK_VC0_ERR_CRC Virtual Channel 0 Payload Checksum Error Interrupt Mask

48.7.16. CSI Fatal Packet Force Interrupt Register

Name: CSI_INT_FORCE_PKT_FATAL
Offset: 0xF8
Reset: 0x00000000
Property: Read/Write

Used for test purposes. Triggers CSI_INT_ST_PKT_FATAL interrupt events individually without the need to activate the conditions that trigger the interrupt sources.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: The corresponding interrupt source is forced.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								FORCE_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
					FORCE_VC3_ERR_CRC	FORCE_VC2_ERR_CRC	FORCE_VC1_ERR_CRC	FORCE_VC0_ERR_CRC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 16 – FORCE_ERR_ECC_DOUBLE Force Header ECC Double Error Interrupt

Bit 3 – FORCE_VC3_ERR_CRC Force Virtual Channel 3 Payload Checksum Error Interrupt

Bit 2 – FORCE_VC2_ERR_CRC Force Virtual Channel 2 Payload Checksum Error Interrupt

Bit 1 – FORCE_VC1_ERR_CRC Force Virtual Channel 1 Payload Checksum Error Interrupt

Bit 0 – FORCE_VC0_ERR_CRC Force Virtual Channel 0 Payload Checksum Error Interrupt

48.7.17. 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 At Least One CRC Error in Last Received Frame of Virtual Channel 3 (cleared on read)

Bit 18 – ERR_FRAME_DATA_VC2 At Least One CRC Error in Last Received Frame of Virtual Channel 2 (cleared on read)

Bit 17 – ERR_FRAME_DATA_VC1 At Least One CRC Error in Last Received Frame of Virtual Channel 1 (cleared on read)

Bit 16 – ERR_FRAME_DATA_VC0 At Least One CRC Error in Last Received Frame of Virtual Channel 0 (cleared on read)

Bit 11 – ERR_F_SEQ_VC3 Incorrect Frame Sequence in Virtual Channel 3 (cleared on read)

Bit 10 – ERR_F_SEQ_VC2 Incorrect Frame Sequence in Virtual Channel 2 (cleared on read)

Bit 9 – ERR_F_SEQ_VC1 Incorrect Frame Sequence in Virtual Channel 1 (cleared on read)

Bit 8 – ERR_F_SEQ_VC0 Incorrect Frame Sequence 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)

48.7.18. CSI Frame Fatal Error Interrupt Mask Register

Name: CSI_INT_MSK_FRAME_FATAL
Offset: 0x104
Reset: 0x00000000
Property: Read/Write

Interrupt mask for CSI_INT_ST_FRAME_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_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 CRC Error in Last Received Frame of Virtual Channel 3 Interrupt Mask

Bit 18 – MASK_ERR_FRAME_DATA_VC2 CRC Error in Last Received Frame of Virtual Channel 2 Interrupt Mask

Bit 17 – MASK_ERR_FRAME_DATA_VC1 CRC Error in Last Received Frame of Virtual Channel 1 Interrupt Mask

Bit 16 – MASK_ERR_FRAME_DATA_VC0 CRC Error in Last Received Frame of Virtual Channel 0 Interrupt Mask

Bit 11 – MASK_ERR_F_SEQ_VC3 Incorrect Frame Sequence in Virtual Channel 3 Interrupt Mask

Bit 10 – MASK_ERR_F_SEQ_VC2 Incorrect Frame Sequence in Virtual Channel 2 Interrupt Mask

Bit 9 – MASK_ERR_F_SEQ_VC1 Incorrect Frame Sequence in Virtual Channel 1 Interrupt Mask

Bit 8 – MASK_ERR_F_SEQ_VC0 Incorrect Frame Sequence in Virtual Channel 0 Interrupt Mask

Bit 3 – MASK_ERR_F_BNDRY_MATCH_VC3 Virtual Channel 3 Error Matching Frame Start with Frame End Interrupt Mask

Bit 2 – MASK_ERR_F_BNDRY_MATCH_VC2 Virtual Channel 2 Error Matching Frame Start with Frame End Interrupt Mask

Bit 1 – MASK_ERR_F_BNDRY_MATCH_VC1 Virtual Channel 1 Error Matching Frame Start with Frame End Interrupt Mask

Bit 0 – MASK_ERR_F_BNDRY_MATCH_VC0 Virtual Channel 0 Error Matching Frame Start with Frame End Interrupt Mask

48.7.19. CSI Frame Fatal Error Interrupt Force Register

Name: CSI_INT_FORCE_FRAME_FATAL
Offset: 0x108
Reset: 0x00000000
Property: Read/Write

Used for test purposes. Triggers CSI_INT_ST_FRAME_FATAL interrupt events individually without the need to activate the conditions that trigger the interrupt sources.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: The corresponding interrupt source is forced.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FORCE_ERR_FRAME_DATA_VC3	FORCE_ERR_FRAME_DATA_VC2	FORCE_ERR_FRAME_DATA_VC1	FORCE_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
					FORCE_ERR_F_SEQ_VC3	FORCE_ERR_F_SEQ_VC2	FORCE_ERR_F_SEQ_VC1	FORCE_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
					FORCE_ERR_F_BNDRY_MATCH_VC3	FORCE_ERR_F_BNDRY_MATCH_VC2	FORCE_ERR_F_BNDRY_MATCH_VC1	FORCE_ERR_F_BNDRY_MATCH_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – FORCE_ERR_FRAME_DATA_VC3 Force CRC Error in Last Received Frame of Virtual Channel 3 Interrupt Error

Bit 18 – FORCE_ERR_FRAME_DATA_VC2 Force CRC Error in Last Received Frame of Virtual Channel 2 Interrupt Error

Bit 17 – FORCE_ERR_FRAME_DATA_VC1 Force CRC Error in Last Received Frame of Virtual Channel 1 Interrupt Error

Bit 16 – FORCE_ERR_FRAME_DATA_VC0 Force CRC Error in Last Received Frame of Virtual Channel 0 Interrupt Error

Bit 11 – FORCE_ERR_F_SEQ_VC3 Force Incorrect Frame Sequence in Virtual Channel 3 Interrupt Error

Bit 10 – FORCE_ERR_F_SEQ_VC2 Force Incorrect Frame Sequence in Virtual Channel 2 Interrupt Error

Bit 9 – FORCE_ERR_F_SEQ_VC1 Force Incorrect Frame Sequence in Virtual Channel 1 Interrupt Error

Bit 8 – FORCE_ERR_F_SEQ_VC0 Force Incorrect Frame Sequence in Virtual Channel 0 Interrupt Error

Bit 3 – FORCE_ERR_F_BNDRY_MATCH_VC3 Force Virtual Channel 3 Error Matching Frame Start with Frame End Interrupt Error

Bit 2 – FORCE_ERR_F_BNDRY_MATCH_VC2 Force Virtual Channel 2 Error Matching Frame Start with Frame End Interrupt Error

Bit 1 – FORCE_ERR_F_BNDRY_MATCH_VC1 Force Virtual Channel 1 Error Matching Frame Start with Frame End Interrupt Error

Bit 0 – FORCE_ERR_F_BNDRY_MATCH_VC0 Force Virtual Channel 0 Error Matching Frame Start with Frame End Interrupt Error

48.7.20. CSI D-PHY Interrupt Status Register

Name: CSI_INT_ST_PHY
Offset: 0x110
Reset: 0x00000000
Property: Read-only

Reading CSI_INT_ST_PHY 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
					PHY_ERRESC_3	PHY_ERRESC_2	PHY_ERRESC_1	PHY_ERRESC_0
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
					PHY_ERRSOTHS_3	PHY_ERRSOTHS_2	PHY_ERRSOTHS_1	PHY_ERRSOTHS_0
Access					R	R	R	R
Reset					0	0	0	0

Bit 19 – PHY_ERRESC_3 Start of Transmission Error on Data Lane 3 (synchronization can still be achieved) (cleared on read)

Bit 18 – PHY_ERRESC_2 Start of Transmission Error on Data Lane 2 (synchronization can still be achieved) (cleared on read)

Bit 17 – PHY_ERRESC_1 Start of Transmission Error on Data Lane 1 (synchronization can still be achieved) (cleared on read)

Bit 16 – PHY_ERRESC_0 Start of Transmission Error on Data Lane 0 (synchronization can still be achieved) (cleared on read)

Bit 3 – PHY_ERRSOTHS_3 Start of Transmission Error on Data Lane 3 (no synchronization achieved) (cleared on read)

Bit 2 – PHY_ERRSOTHS_2 Start of Transmission Error on Data Lane 2 (no synchronization achieved) (cleared on read)

Bit 1 – PHY_ERRSOTHS_1 Start of Transmission Error on Data Lane 1 (no synchronization achieved) (cleared on read)

Bit 0 – PHY_ERRSOTHS_0 Start of Transmission Error on Data Lane 0 (no synchronization achieved) (cleared on read)

48.7.21. CSI D-PHY Interrupt Mask Register

Name: CSI_INT_MSK_PHY
Offset: 0x114
Reset: 0x00000000
Property: Read/Write

Interrupt mask for CSI_INT_MSK_PHY 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_PHY_ERRESC_3	MASK_PHY_ERRESC_2	MASK_PHY_ERRESC_1	MASK_PHY_ERRESC_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
					MASK_PHY_ERRSOTHS_3	MASK_PHY_ERRSOTHS_2	MASK_PHY_ERRSOTHS_1	MASK_PHY_ERRSOTHS_0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – MASK_PHY_ERRESC_3 Start of Transmission Error on Data Lane 3 (synchronization can still be achieved) Interrupt Mask

Bit 18 – MASK_PHY_ERRESC_2 Start of Transmission Error on Data Lane 2 (synchronization can still be achieved) Interrupt Mask

Bit 17 – MASK_PHY_ERRESC_1 Start of Transmission Error on Data Lane 1 (synchronization can still be achieved) Interrupt Mask

Bit 16 – MASK_PHY_ERRESC_0 Start of Transmission Error on Data Lane 0 (synchronization can still be achieved) Interrupt Mask

Bit 3 – MASK_PHY_ERRSOTHS_3 Start of Transmission Error on Data Lane 3 (no synchronization achieved) Interrupt Mask

Bit 2 – MASK_PHY_ERRSOTHS_2 Start of Transmission Error on Data Lane 2 (no synchronization achieved) Interrupt Mask

Bit 1 – MASK_PHY_ERRSOTHS_1 Start of Transmission Error on Data Lane 1 (no synchronization achieved) Interrupt Mask

Bit 0 – MASK_PHY_ERRSOTHS_0 Start of Transmission Error on Data Lane 0 (no synchronization achieved) Interrupt Mask

48.7.22. CSI D-PHY Interrupt Force Register

Name: CSI_INT_FORCE_PHY
Offset: 0x118
Reset: 0x00000000
Property: Read/Write

Used for test purposes. Triggers INT_ST_PHY interrupt events individually without the need to activate the conditions that trigger the interrupt sources.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: The corresponding interrupt source is forced.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FORCE_PHY_ERRESC_3	FORCE_PHY_ERRESC_2	FORCE_PHY_ERRESC_1	FORCE_PHY_ERRESC_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
					FORCE_PHY_ERRSOTHS_3	FORCE_PHY_ERRSOTHS_2	FORCE_PHY_ERRSOTHS_1	FORCE_PHY_ERRSOTHS_0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – FORCE_PHY_ERRESC_3 Force Start of Transmission Error on Data Lane 3 (synchronization can still be achieved) Interrupt

Bit 18 – FORCE_PHY_ERRESC_2 Force Start of Transmission Error on Data Lane 2 (synchronization can still be achieved) Interrupt

Bit 17 – FORCE_PHY_ERRESC_1 Force Start of Transmission Error on Data Lane 1 (synchronization can still be achieved) Interrupt

Bit 16 – FORCE_PHY_ERRESC_0 Force Start of Transmission Error on Data Lane 0 (synchronization can still be achieved) Interrupt

Bit 3 – FORCE_PHY_ERRSOTHS_3 Force Start of Transmission Error on Data Lane 3 (no synchronization achieved) Interrupt

Bit 2 – FORCE_PHY_ERRSOTHS_2 Force Start of Transmission Error on Data Lane 2 (no synchronization achieved) Interrupt

Bit 1 – FORCE_PHY_ERRSOTHS_1 Force Start of Transmission Error on Data Lane 1 (no synchronization achieved) Interrupt

Bit 0 – FORCE_PHY_ERRSOTHS_0 Force Start of Transmission Error on Data Lane 0 (no synchronization achieved) Interrupt

48.7.23. CSI Packet Interrupt Status Register

Name: CSI_INT_ST_PKT
Offset: 0x120
Reset: 0x00000000
Property: Read-only

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 Header Error Detected and Corrected on Virtual Channel 3 (cleared on read)

Bit 18 – VC2_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 2 (cleared on read)

Bit 17 – VC1_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 1 (cleared on read)

Bit 16 – VC0_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 0 (cleared on read)

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)

48.7.24. CSI Packet Interrupt Mask Register

Name: CSI_INT_MSK_PKT
Offset: 0x124
Reset: 0x00000000
Property: Read/Write

Interrupt mask for CSI_INT_MSK_PKT 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_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 Header Error Detected and Corrected on Virtual Channel 3 Interrupt Mask

Bit 18 – MASK_VC2_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 2 Interrupt Mask

Bit 17 – MASK_VC1_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 1 Interrupt Mask

Bit 16 – MASK_VC0_ERR_ECC_CORRECTED Header Error Detected and Corrected on Virtual Channel 0 Interrupt Mask

Bit 3 – MASK_ERR_ID_VC3 Unrecognized or Unimplemented Data Type Detected in Virtual Channel 3 Interrupt Mask

Bit 2 – MASK_ERR_ID_VC2 Unrecognized or Unimplemented Data Type Detected in Virtual Channel 2 Interrupt Mask

Bit 1 – MASK_ERR_ID_VC1 Unrecognized or Unimplemented Data Type Detected in Virtual Channel 1 Interrupt Mask

Bit 0 – MASK_ERR_ID_VC0 Unrecognized or Unimplemented Data Type Detected in Virtual Channel 0 Interrupt Mask

48.7.25. CSI Packet Interrupt Force Register

Name: CSI_INT_FORCE_PKT
Offset: 0x128
Reset: 0x00000000
Property: Read/Write

Used for test purposes. Triggers CSI_INT_ST_PKT interrupt events individually without the need to activate the conditions that trigger the interrupt sources.

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: The corresponding interrupt source is forced.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FORCE_VC3_ERR_ECC_CORRECTED	FORCE_VC2_ERR_ECC_CORRECTED	FORCE_VC1_ERR_ECC_CORRECTED	FORCE_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
					FORCE_ERR_ID_VC3	FORCE_ERR_ID_VC2	FORCE_ERR_ID_VC1	FORCE_ERR_ID_VC0
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 19 – FORCE_VC3_ERR_ECC_CORRECTED Force Start of Transmission Error on Data Lane 3 (synchronization can still be achieved) Interrupt

Bit 18 – FORCE_VC2_ERR_ECC_CORRECTED Force Start of Transmission Error on Data Lane 2 (synchronization can still be achieved) Interrupt

Bit 17 – FORCE_VC1_ERR_ECC_CORRECTED Force Start of Transmission Error on Data Lane 1 (synchronization can still be achieved) Interrupt

Bit 16 – FORCE_VC0_ERR_ECC_CORRECTED Force Start of Transmission Error on Data Lane 0 (synchronization can still be achieved) Interrupt

Bit 3 – FORCE_ERR_ID_VC3 Force Start of Transmission Error on Data Lane 3 (no synchronization achieved) Interrupt

Bit 2 – FORCE_ERR_ID_VC2 Force Start of Transmission Error on Data Lane 2 (no synchronization achieved) Interrupt

Bit 1 – FORCE_ERR_ID_VC1 Force Start of Transmission Error on Data Lane 1 (no synchronization achieved) Interrupt

Bit 0 – FORCE_ERR_ID_VC0 Force Start of Transmission Error on Data Lane 0 (no synchronization achieved) Interrupt

48.7.26. 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 Data Descrambling Enable

Value	Description
0	No data de-scrambling.
1	Activates the data de-scrambling.

48.7.27. 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] Data Lane 0 Descrambler Seed
Data Lane 0 descrambler seed.

48.7.28. 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] Data Lane 1 Descrambler Seed
Data Lane 1 descrambler seed.

48.7.29. CSI Lane 2 Scrambling Seed Register

Name: CSI_SCRAMBLING_SEED2
Offset: 0x30C
Reset: 0x00001248
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_LANE2[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	1	0

Bit	7	6	5	4	3	2	1	0
	SCRAMBLE_SEED_LANE2[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	0	0	0

Bits 15:0 – SCRAMBLE_SEED_LANE2[15:0] Data Lane 2 Descrambler Seed
Data Lane 2 descrambler seed.

48.7.30. CSI Lane 3 Scrambling Seed Register

Name: CSI_SCRAMBLING_SEED3
Offset: 0x310
Reset: 0x00001428
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_LANE3[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	1	0	0

Bit	7	6	5	4	3	2	1	0
	SCRAMBLE_SEED_LANE3[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	1	0	0	0

Bits 15:0 – SCRAMBLE_SEED_LANE3[15:0] Data Lane 3 Descrambler Seed
Data Lane 3 descrambler seed.

49. CSI-2 Demultiplexer Controller (CSI2DC)

49.1. 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 49.1. 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)

49.2. Embedded Characteristics

- CSI-2 Version 1.3 Specification-compliant Demultiplexer
- Four Virtual Channels

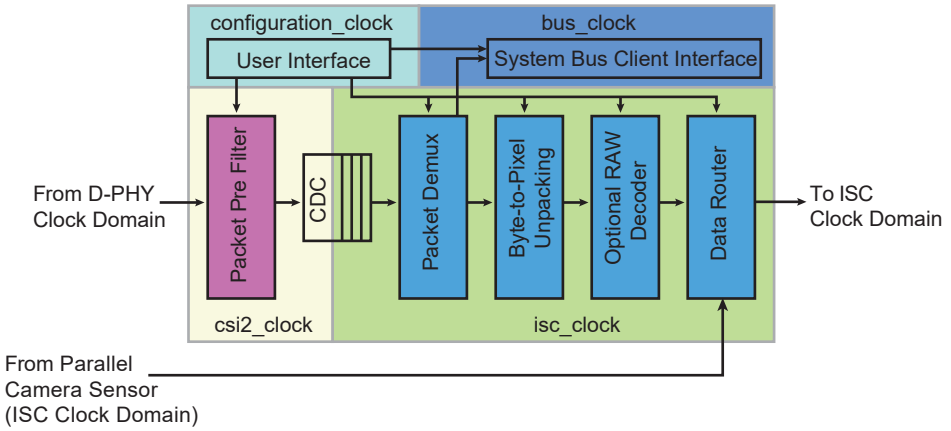
- Primary and Secondary Image Format
- RGB, YUV, RAW, Compressed RAW, User-defined Byte-based Packets
- Recommended Memory Storage Format or Single Pixel Per Clock Cycle
- Non-Image Data Packet and Generic Short Packet Support
- Image Data Snoop Controller (four entries)
- Programmable Video Pipeline Filter
- Programmable Data Pipeline Filter
- Progressive and Interlaced Content
- Programmable DMA Client Interface

49.3. 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 peripheral clock): 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 49.1. CSI2DC Block Diagram



49.4. I/O Lines Description

Table 49.2. I/O Lines Description

Signal Name	Description	Type
MIPI_CLKP	MIPI D-PHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI D-PHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI D-PHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_DP2	MIPI D-PHY differential output data lane 2	Input/Output
MIPI_DN2		
MIPI_DP3	MIPI D-PHY differential output data lane 3	Input/Output
MIPI_DN3		

Table 49.2. I/O Lines Description (continued)

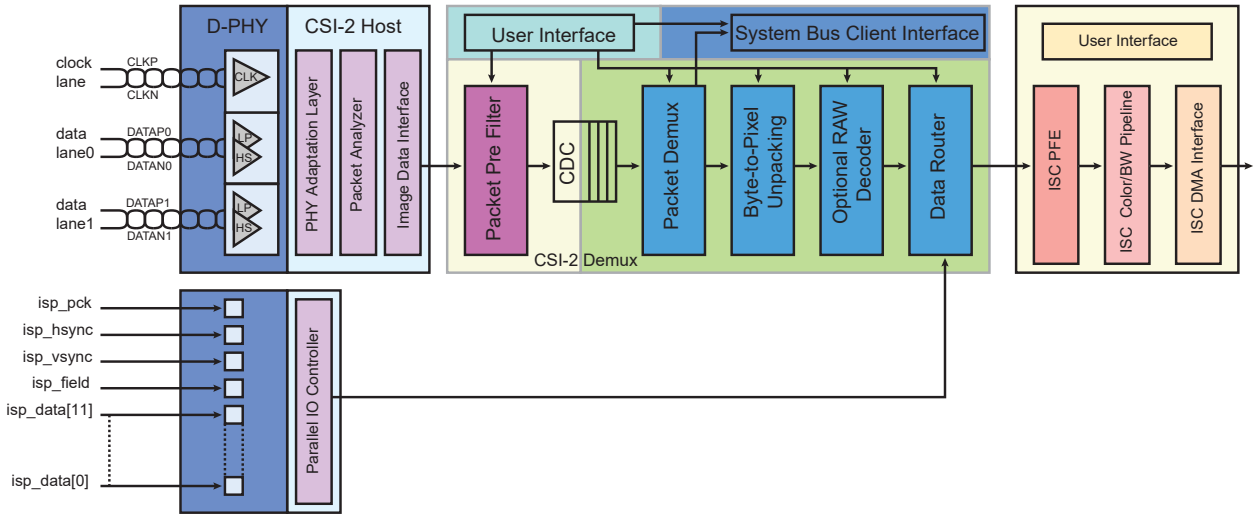
Signal Name	Description	Type
MIPI_REXT	Calibration reference resistor	Input/Output

49.5. 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 49.2. CSI2DC System Integration

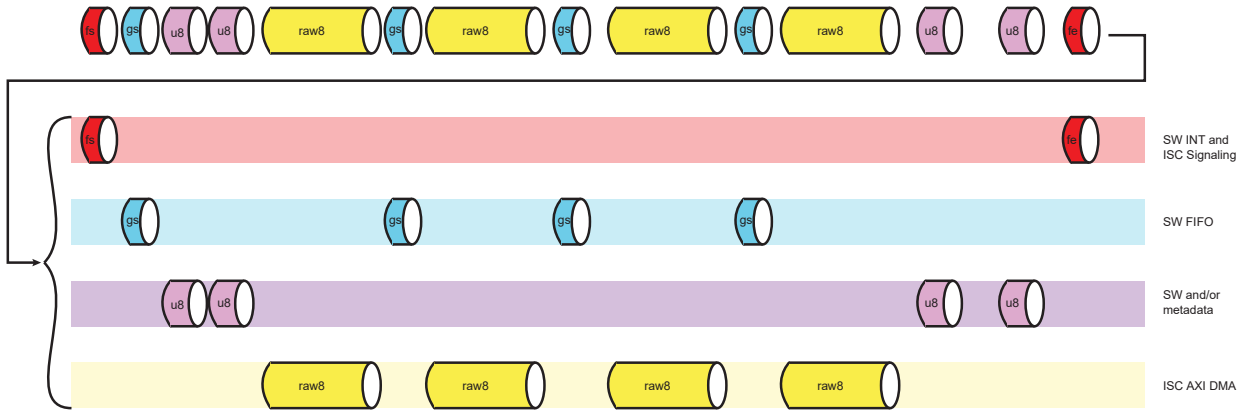


When a packet is received, a snoop controller can be used to monitor CSI-2 traffic and identify that the desired information is successfully transmitted and received in the CSI2DC.

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. The video pipe must be enabled to route data from the camera to the ISC.

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. 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 data pipe must be enabled to route data from the camera to the system bus client interface.

Figure 49.3. Packet Demultiplexing Example



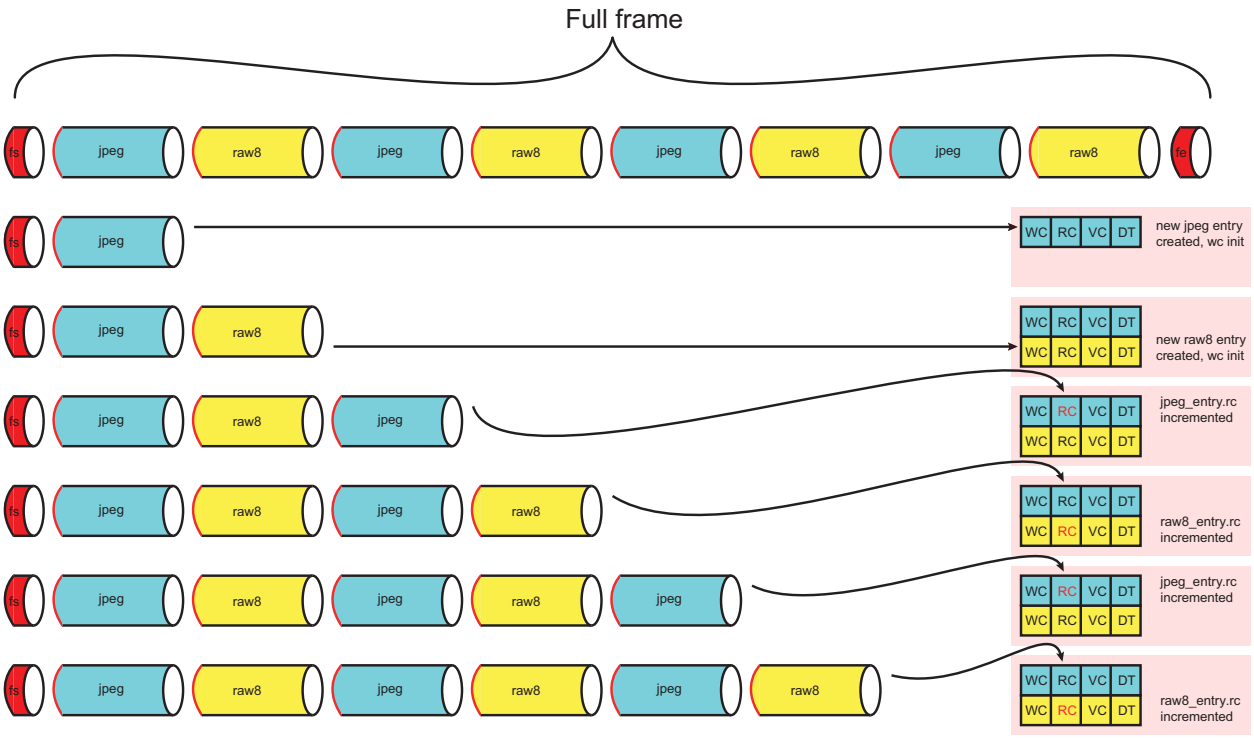
49.5.1. Image Data Snoop (IDS) Controller

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 49.3. 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 49.4. Image Data Snoop Controller



49.5.2. 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 49.4. 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.

49.5.3. Generic Short Packet Demux

Generic Short Packets (GSP) 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 49.5. 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

49.5.4. Generic Long Packet Demux

The Generic Long Packet (GLP) contains three packet subclasses:

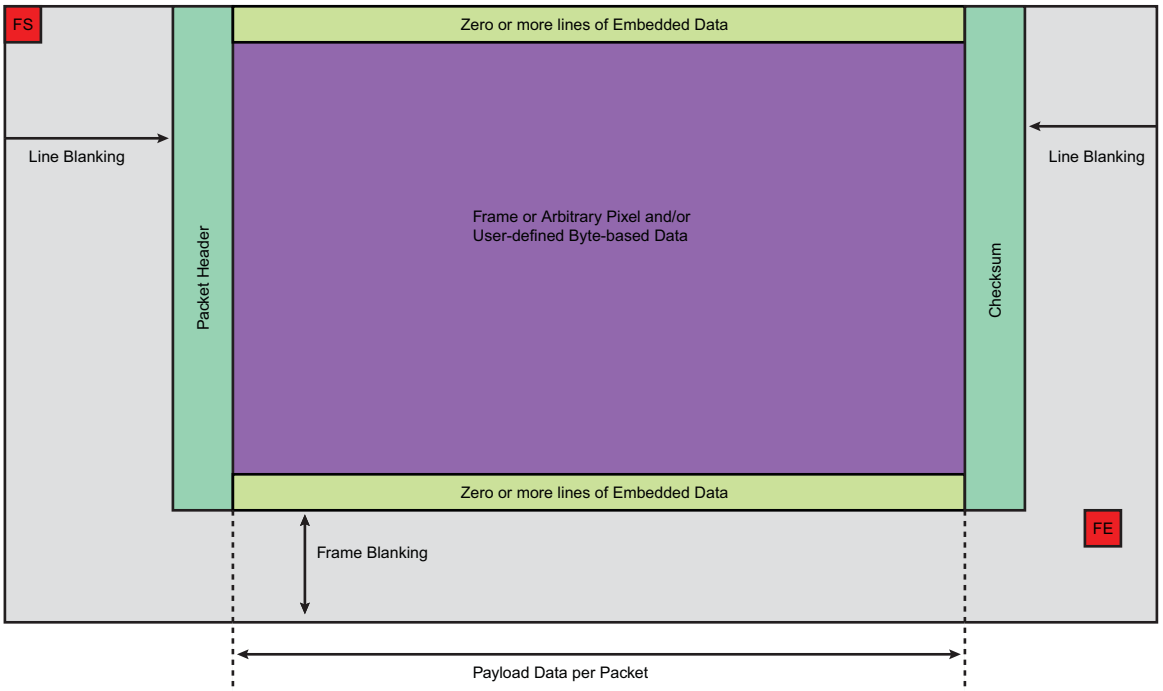
- NULL packet, data type set to 0x10
- Blanking data, data type set to 0x11
- Embedded 8-bit non-image data, data type set to 0x12

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 49.6. 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 49.5. Generic Long Packet Embedded Information Packet



49.5.5. 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 49.7. 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

49.5.5.1. YUV 420 8-bit Legacy Mode

YUV 420 legacy data type is set to 0x1A.

Note: For this mode, ISC only supports 32-bit packed acquisition.

Figure 49.6. YUV 420 8-bit Legacy Mapping

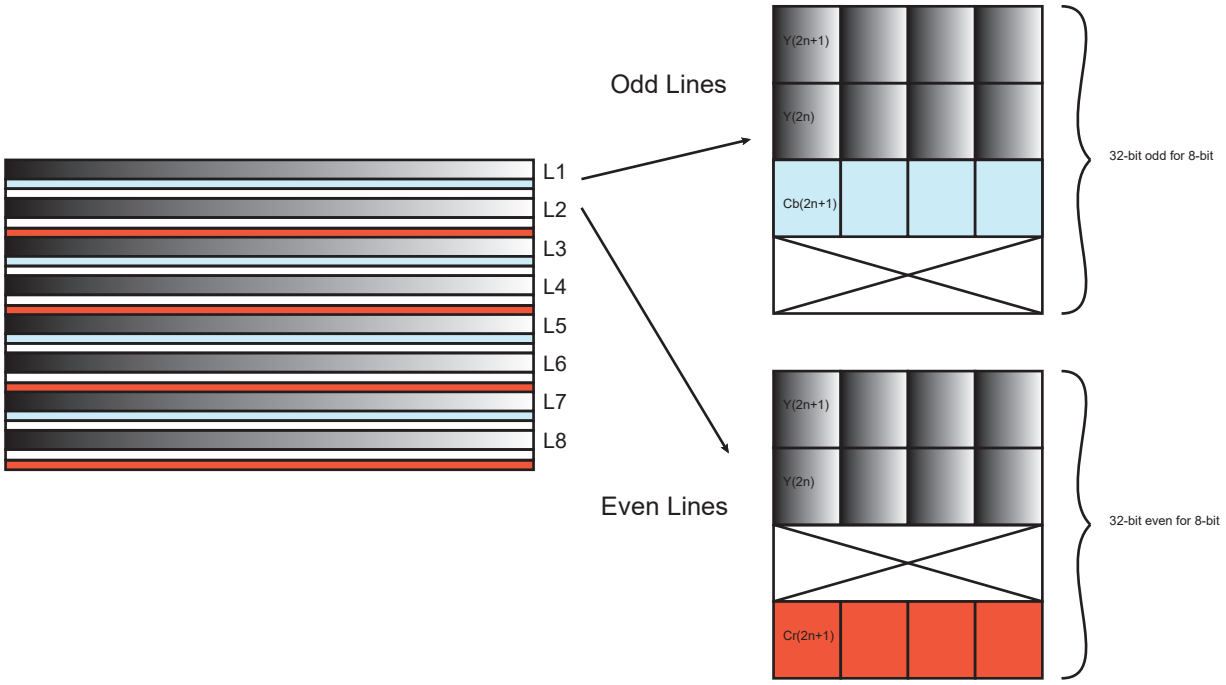


Table 49.8. YUV420 8-bit (Legacy) Odd Line Format (1, 3, 5, etc.) Mapping

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 49.9. 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 49.10. 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 49.11. 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							
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

49.5.5.2.YUV 420 8-bit Mode

Data type is set to 0x18 or 0x1C.

Figure 49.7. YUV 420 8-bit Mode

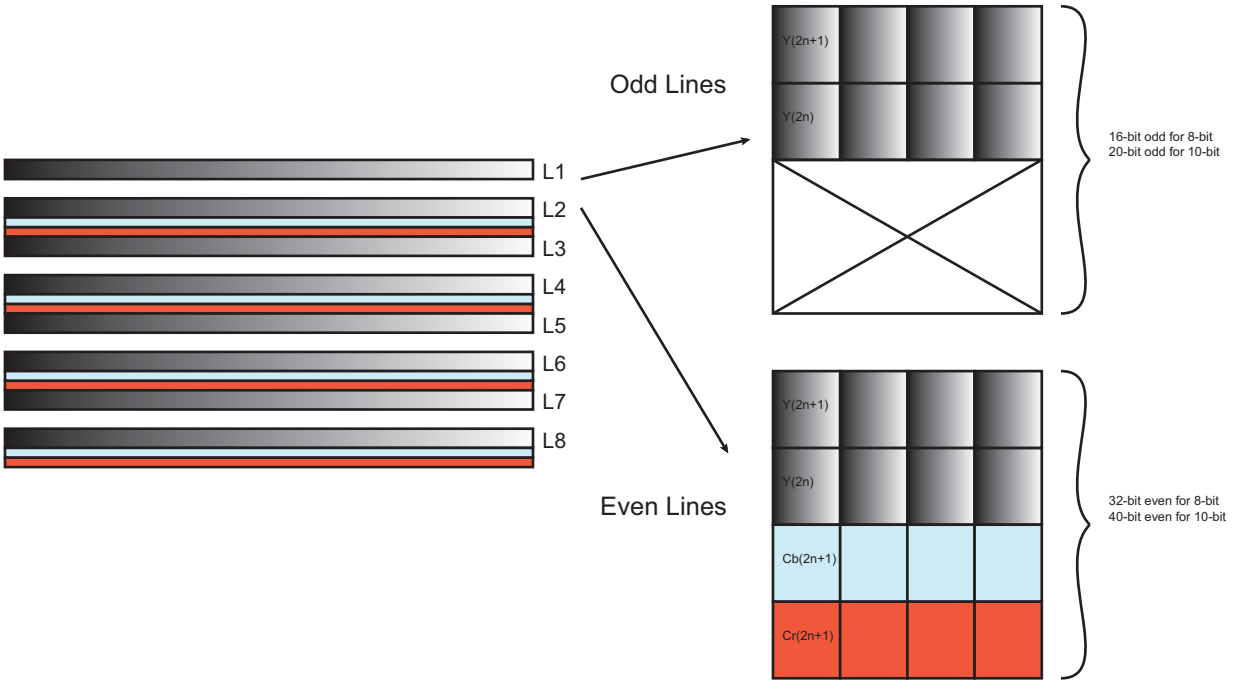


Table 49.12. YUV420 8-bit Odd Line Format Mapping

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]	{U1[7:0], 2'b00} (Cb component)
demux_data[9:0]	Invalid data

Table 49.13. 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]	Invalid data
demux_data[9:0]	{V1[7:0], 2'b00} (Cr component)

Table 49.14. 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 49.15. 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

49.5.5.3.YUV 420 10-bit Mode

Data type is set to 0x19 or 0x1D.

Table 49.16. 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 49.17. 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 49.18. 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]			
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 49.19. 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]			
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

49.5.5.4.YUV 422 8-bit Mode

Data type is set to 0x1E.

Figure 49.8. YUV 422 8-bit Mode Bit

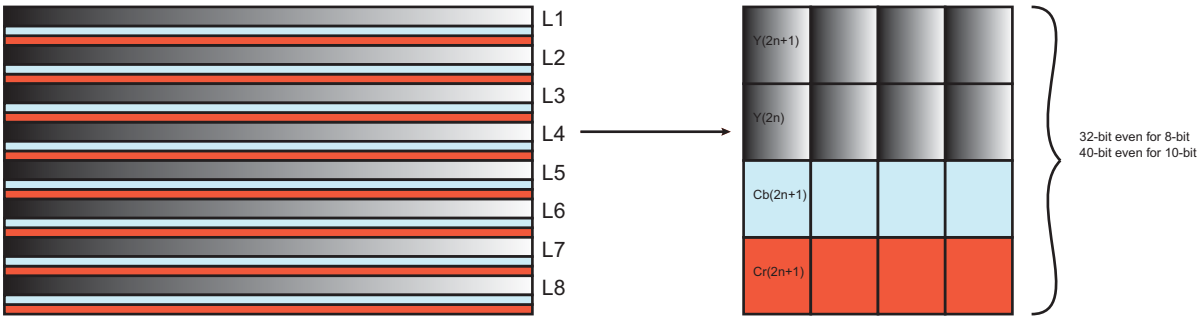


Table 49.20. YUV422 8-bit Mapping

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}

Table 49.20. YUV422 8-bit Mapping (continued)

DEMUX_DATA Slice	YUV 422 8-Bit Data Mapping
demux_data[9:0]	{V1[7:0], 2'b00} (Cr component)

Table 49.21. 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

49.5.5.5.YUV 422 10-bit Mode

Data type is set to 0x1F.

Table 49.22. 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 49.23. 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]							
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

49.5.6. RGB Packet Demux

Table 49.24. 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

49.5.6.1.RGB888

Data type is set to 0x24.

Table 49.25. 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 49.26. 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 49.27. 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

49.5.6.2.RGB666

Data type is set to 0x23.

Table 49.28. 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 49.29. 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 49.30. 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

49.5.6.3.RGB565

Data type is set to 0x22.

Table 49.31. 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 49.32. 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]}

49.5.6.4.RGB555

Data Type is set to 0x21.

Table 49.33. 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 49.34. 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]}

49.5.6.5.RGB444

Data Type is set to 0x20.

Table 49.35. 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 49.36. 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]}

49.5.7. RAW Packet Demux

Table 49.37. 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

49.5.7.1.RAW6

Data type is set to 0x28.

Table 49.38. 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 49.39. 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

49.5.7.2.RAW7

Data type is set to 0x29.

Table 49.40. 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}
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 49.41. 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

49.5.7.3.RAW8

Data type is set to 0x2A.

Table 49.42. 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 49.43. 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

49.5.7.4.RAW10

Data type is set to 0x2B.

Table 49.44. 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]}
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 49.45. 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]			
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

49.5.7.5.RAW12

Data type is set to 0x2C.

Table 49.46. 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 49.47. 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

49.5.7.6.RAW14

Data type is set to 0x2D.

Table 49.48. 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 49.49. 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]							
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

49.5.8. 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 49.50. 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.

49.5.9. 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 49.51. Decompression Mode Field Mapping to X-Y-Z Naming Convention

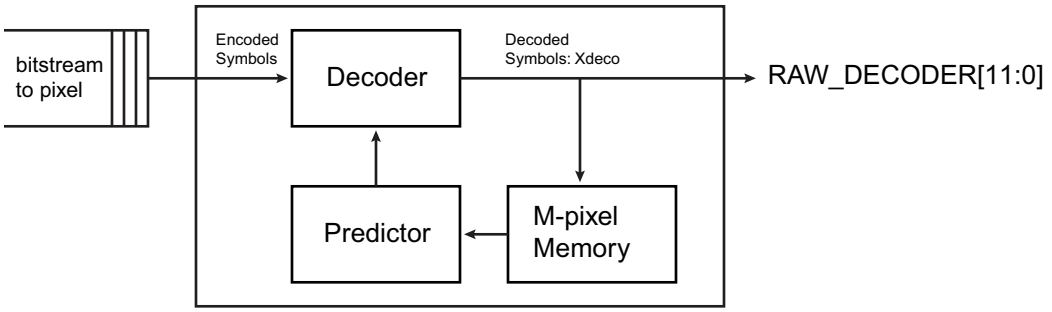
CSI2DC_VPCFG.DM	X-Y-Z CSI-2 Naming Rule
DECODER8TO12	12-8-12
DECODER7TO12	12-7-12
DECODER6TO12	12-6-12
DECODER8TO10	10-8-10
DECODER7TO10	10-7-10
DECODER6TO10	10-6-10

The data compression schemes specified in CSI-2 Annex E are lossy and are designed to encode each line independently.

Table 49.52. 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 49.9. RAW Decompression Decoder



49.6. 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]							

Register Summary (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]			

Register Summary (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]							
0xB8	CSI2DC_IDSEW0R3	31:24								
		23:16								
		15:8								
		7:0	VC[1:0]		DT[5:0]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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
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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]					

49.6.1. 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.

49.6.2. 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.

49.6.3. 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.

49.6.4. 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

49.6.5. 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

49.6.6. 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

49.6.7. 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.

Value	Description
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.

49.6.8. 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.
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.

49.6.9. 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

49.6.10. 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.

49.6.11. 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 Status

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. This bit is reset after the register read operation.

Bits 15:12 – LE[3:0] Line End Interrupt Status

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. This bit is reset after the register read operation.

Bits 11:8 – LS[3:0] Line Start Interrupt Status

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. This bit is reset after the register read operation.

Bits 7:4 – FE[3:0] Frame End Interrupt Status

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. This bit is reset after the register read operation.

Bits 3:0 – FS[3:0] Frame Start Interrupt Status

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.
1	A bit set at position i in the field FS indicates that a Frame Start interrupt is pending for virtual channel i. This bit is reset after the register read operation.

49.6.12. 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

49.6.13. 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

49.6.14. 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

49.6.15. 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

49.6.16. 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

49.6.17. 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

49.6.18. 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

49.6.19. 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

49.6.20. 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.

49.6.21. 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.

49.6.22. 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.

49.6.23. 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. This bit is reset after the register read operation.

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. This bit is reset after the register read operation.

49.6.24. CSI2DC GSP Status 0 Register

Name: CSI2DC_GSPS0R
Offset: 0x6C
Reset: 0x00080000
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	1	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.

49.6.25. CSI2DC GSP Status 1 Register

Name: CSI2DC_GSPS1R
Offset: 0x70
Reset: 0x00080000
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	1	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.

49.6.26. CSI2DC GSP Status 2 Register

Name: CSI2DC_GSPS2R
Offset: 0x74
Reset: 0x00080000
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	1	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.

49.6.27. CSI2DC GSP Status 3 Register

Name: CSI2DC_GSPS3R
Offset: 0x78
Reset: 0x00080000
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	1	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.

49.6.28. 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.

49.6.29. 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

49.6.30. 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.

49.6.31. 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. This bit is reset after the register read operation.

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. This bit is reset after the register read operation.

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. This bit is reset after the register read operation.

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.

Value	Description
1	A bit set at position i in the field NU indicates that a null packet interrupt is pending for virtual channel i. This bit is reset after the register read operation.

49.6.32. 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.

49.6.33. 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.

49.6.34. 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.

49.6.35. 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.

49.6.36. 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

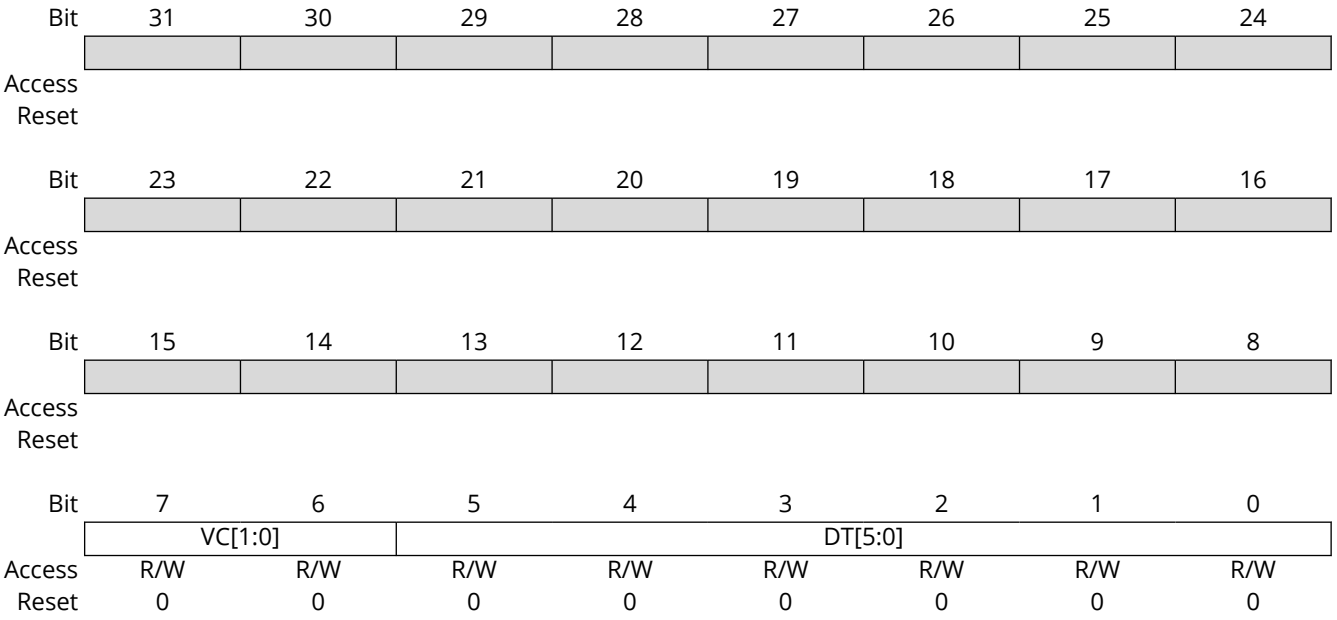
Value	Description
0	No effect.
1	Indicates that the IDS controller captured more than 4 entries. This bit is reset after the register read operation.

Bits 3:0 – IDS[3:0] Image Data Snoop Interrupt Status

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. This bit is reset after the register read operation.

49.6.37. CSI2DC IDS Entry Word 0 Register

Name: CSI2DC_IDSEW0Rx
Offset: 0xA0 + x*0x08 [x=0..3]
Reset: 0x00000000
Property: Read/Write



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.

49.6.38. 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.

49.6.39. 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.

49.6.40. 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.

49.6.41. 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

49.6.42. 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

49.6.43. 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

49.6.44. 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 since the last clear operation of the register.
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 since the last clear operation of the register.
1	Data overflow in the clock domain crossing FIFO.

Bit 4 – RXOVF1 Bank 1 Overflow

Value	Description
0	No overflow detected since the last clear operation of the register.
1	An overflow occurred in bank 1.

Bit 3 – RXOVF0 Bank 0 Overflow

Value	Description
0	No overflow detected since the last clear operation of the register.
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.

49.6.45. 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.

49.6.46. 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.

49.6.47. 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.

49.6.48. 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.

49.6.49. 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

49.6.50. 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

49.6.51. 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

49.6.52. 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 Status

Value	Description
0	No packet overflow since the last read of the register.
1	A packet overflow has been detected.

Bit 4 – LTE Packet Longer Than Expected Interrupt Status

Value	Description
0	No packet longer than expected since the last read of the register.
1	A packet longer than expected has been detected.

Bit 3 – STE Packet Shorter Than Expected Interrupt Status

Value	Description
0	No packet shorter than expected since the last read of the register.
1	A packet shorter than expected has been detected.

Bit 2 – CTLOVF Control Buffer Overflow Interrupt Status

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 Status

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.

49.6.53. 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.

49.6.54. 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
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.

49.6.55. 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

49.6.56. 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.

49.6.57. CSI2DC Video Pipe Data Type Remap Register

Name: CSI2DC_VPDTRR
Offset: 0x108
Reset: 0x00000000
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
		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.

50. Image Sensor Controller (ISC)

50.1. 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 Host DMA interface.

50.2. 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 2560 x 1920
- 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
- Register Write Protection

50.3. ISC Block Diagram and Use Cases

50.3.1. Functional Diagrams

Figure 50.1. ISC Block Diagram

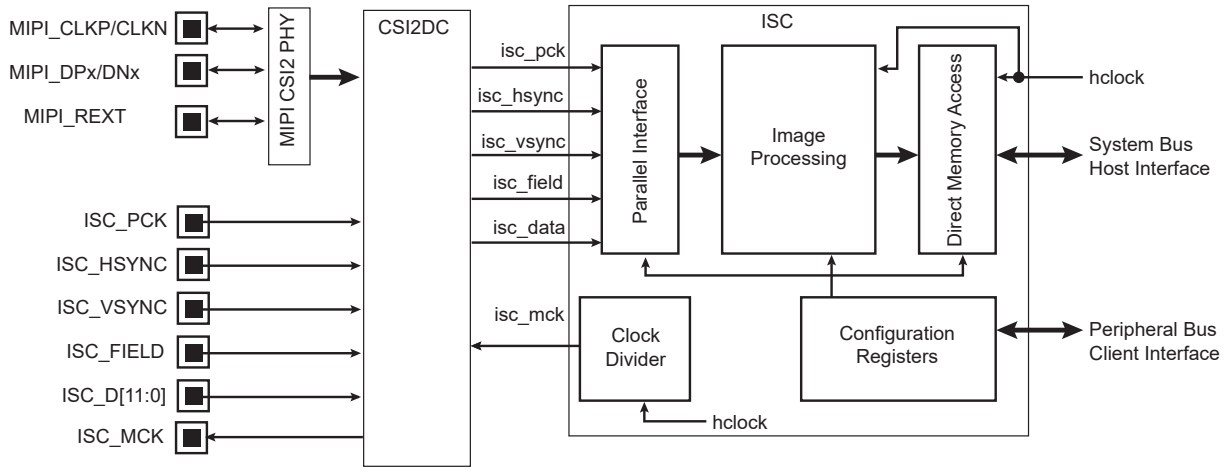
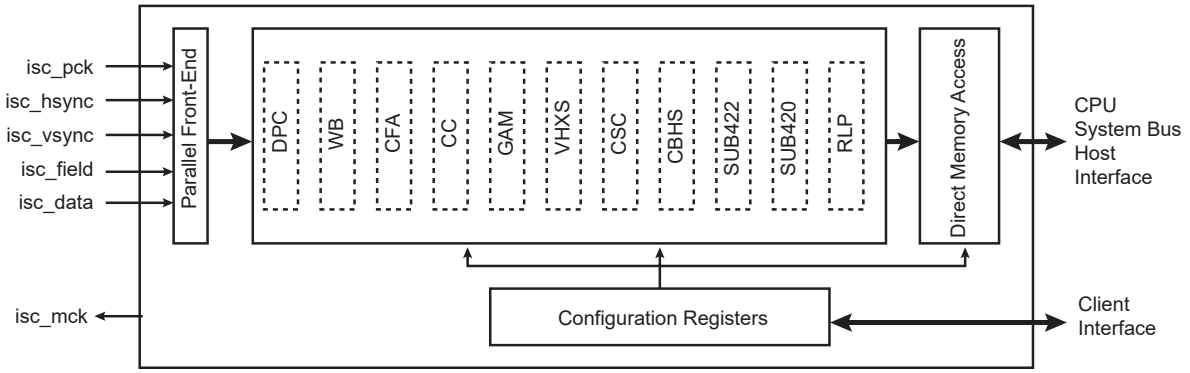


Figure 50.2. ISC Raw Bayer Signal Processor



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.

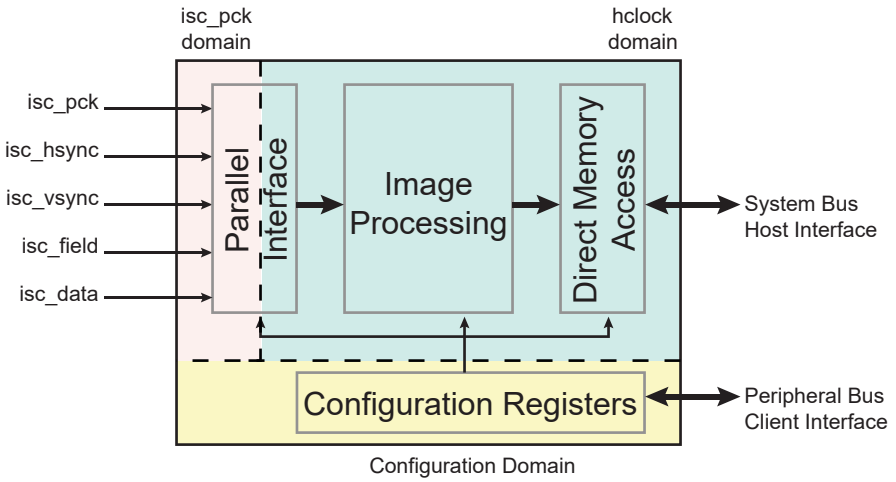
50.4. I/O Lines Description

Table 50.1. 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 D-PHY differential output clock lane	Input/Output
MIPI_CLKN		
MIPI_DP0	MIPI D-PHY differential output data lane 0	Input/Output
MIPI_DN0		
MIPI_DP1	MIPI D-PHY differential output data lane 1	Input/Output
MIPI_DN1		
MIPI_DP2	MIPI D-PHY differential output data lane 2	Input/Output
MIPI_DN2		
MIPI_DP3	MIPI D-PHY differential output data lane 3	Input/Output
MIPI_DN3		
MIPI_REXT	Calibration reference resistor	Input/Output

50.4.1. Clock Domain Diagram

Figure 50.3. Clock Domain Hierarchy



50.4.2. Typical Use Cases

Figure 50.4. Raw Bayer Sensor without Embedded Image Processor

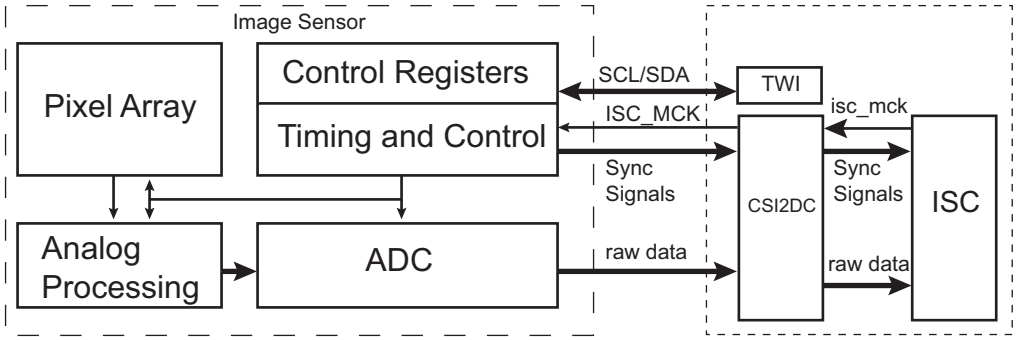


Figure 50.5. MIPI CSI2 Serial Link

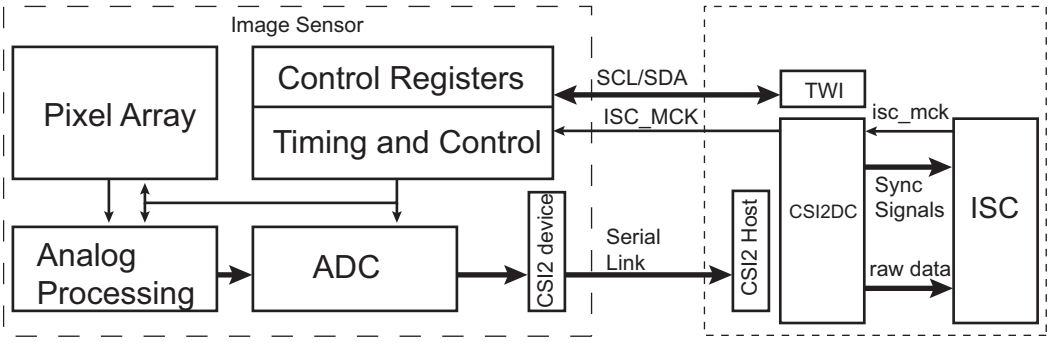


Figure 50.6. Raw Bayer Sensor with Embedded Image Processor

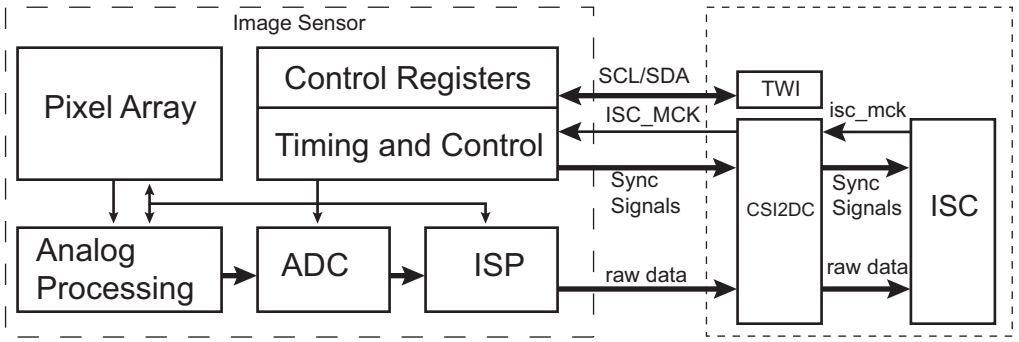


Figure 50.7. BT656 Video Interface Sensor

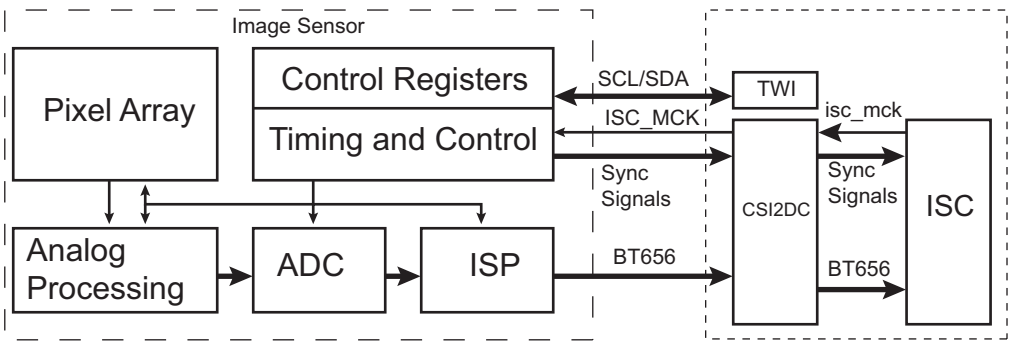


Figure 50.8. Sensor with JPEG Output

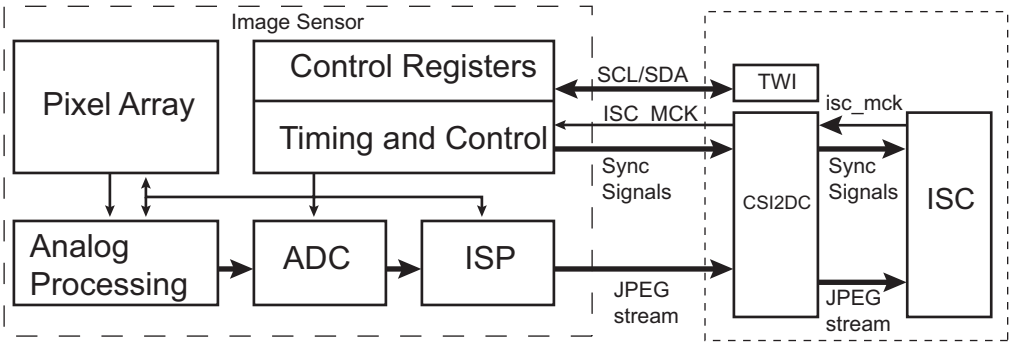
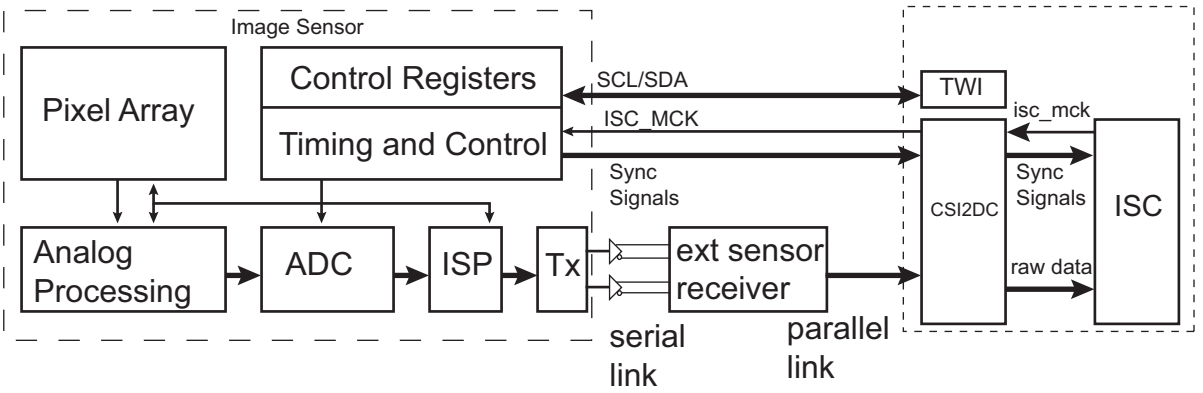


Figure 50.9. Serial CMOS Sensor with External Parallel Bridge



50.5. Product Dependencies

50.5.1. 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).

50.5.2. 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)".

50.5.3. 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.

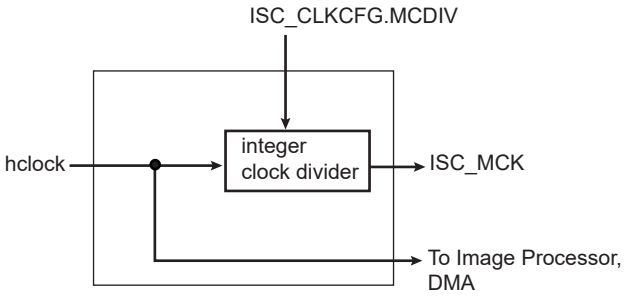
50.6. Functional Description

50.6.1. 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 the Block Diagram.

ISC_MCK has one programmable clock divider (ISC_CLKCFG.MCDIV). The clock is enabled using ISC_CLKEN.MCEN.

Figure 50.10. Clock Divider Block Diagram



The ISC is designed to accept input signals that are asynchronous to hclock.

Synchronization is done internally as long as the following relationship holds:

- isc_pck frequency is lower than or equal to hclock frequency.

50.6.1.1. Software Requirement

A software write operation to ISC_CLKEN or ISC_CLKDIS requires double clock domain synchronization and is not permitted when ISC_CLKSR.SIP is asserted.

50.6.2. Parallel Interface Timing Description

The parallel interface protocol supports two operating modes.

Figure 50.11. Free-Running Pixel Clock

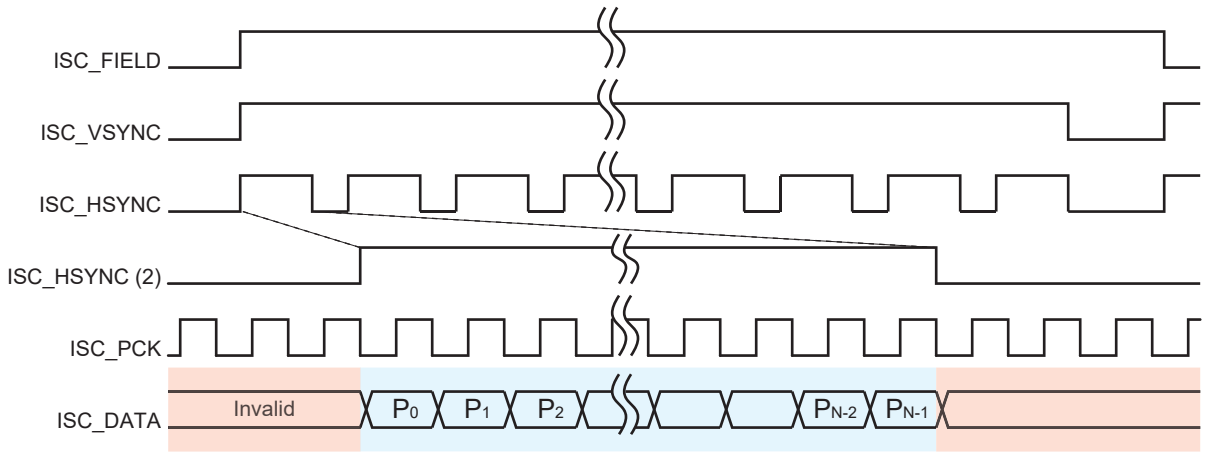
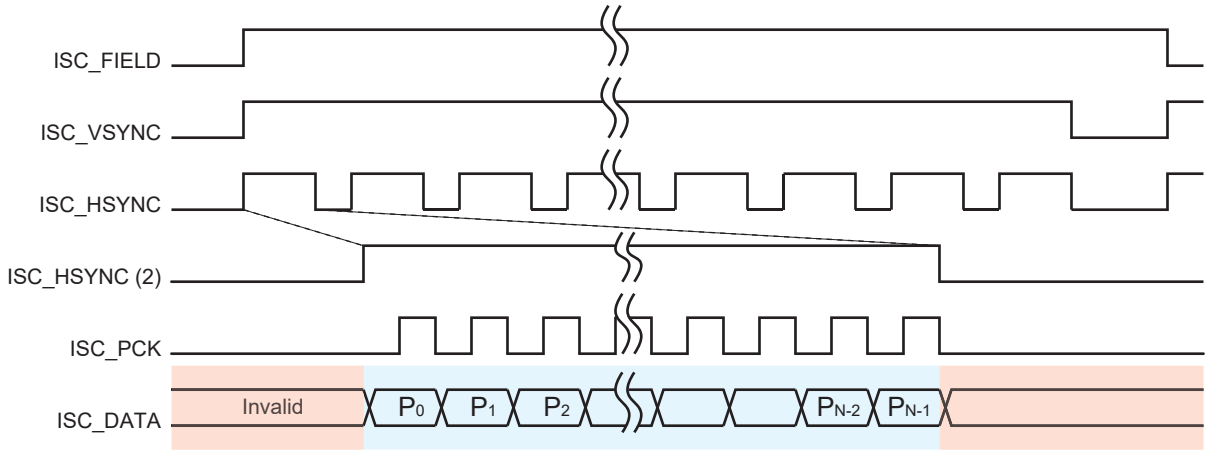


Figure 50.12. Gated Pixel Clock



50.6.3. BT.601/656/1120 Embedded Timing Synchronization Operation

The ISC module supports embedded synchronization decoding. When 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 CCIR10_8N is set, the bitstream is 10 bits wide, otherwise it is only 8 bits wide. When ISC_PFE_CFG0.CCIR_CRC is set, the decoder automatically corrects the error.

Figure 50.13. Field/Segment Timing Relationship for Interlaced and Segmented Frame Systems

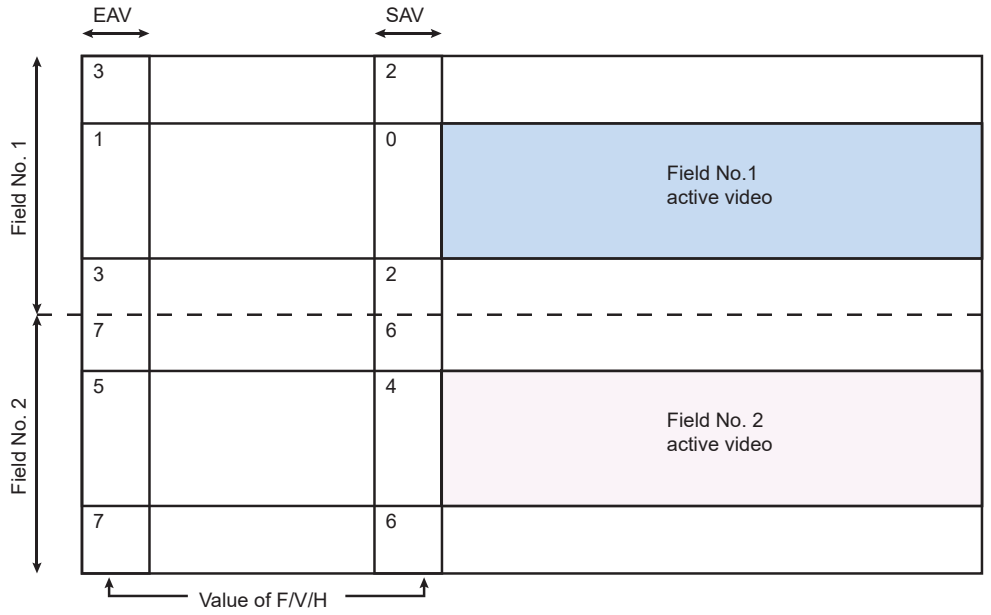
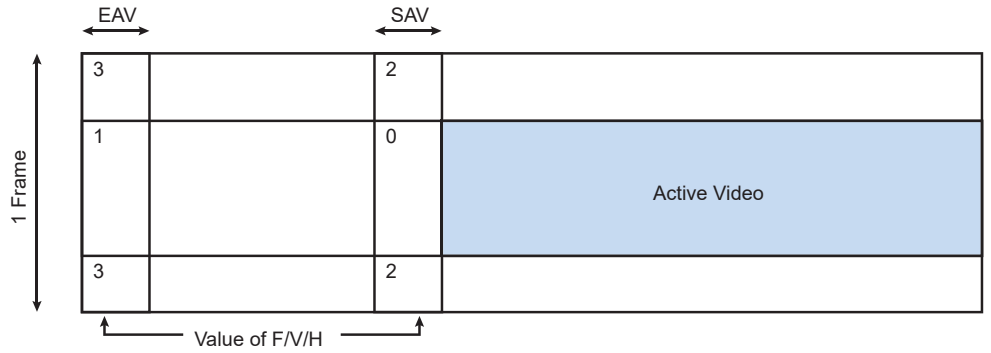


Figure 50.14. Frame Timing Relationship for Progressive Systems



50.6.4. Parallel Interface External Sensor Connections

50.6.4.1. 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
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

50.6.4.2. 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

50.6.4.3. 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

50.6.4.4. Monochrome Parallel Interface

The table below shows how to connect the data bus of a Monochrome sensor.

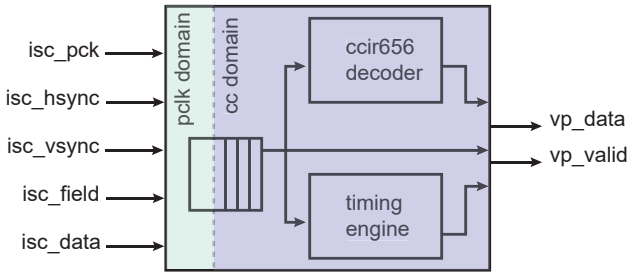
Interface	Mono 12-bit	Mono 11-bit	Mono 10-bit	Mono 9-bit	Mono 8-bit
ISC_DATA[11](MSB)	DOUT[11]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]
ISC_DATA[10]	DOUT[10]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]
ISC_DATA[9]	DOUT[9]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]
ISC_DATA[8]	DOUT[8]	DOUT[7]	DOUT[6]	DOUT[5]	DOUT[4]
ISC_DATA[7]	DOUT[7]	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

50.6.5. MIPI Interface Mapping

When operating with a MIPI interface, the CSI2DC maps the pixel content on the input data bus according to the data format. Refer to the Functional Description in the section “CSI-2 Demultiplexer Controller (CSI2DC)”.

50.6.6. Parallel Front End (PFE) Module

Figure 50.15. 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]
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 ISC_PFE_CFG0.CONT is cleared, the ISC transfers a single image to memory,

Figure 50.16. Single Shot Mode

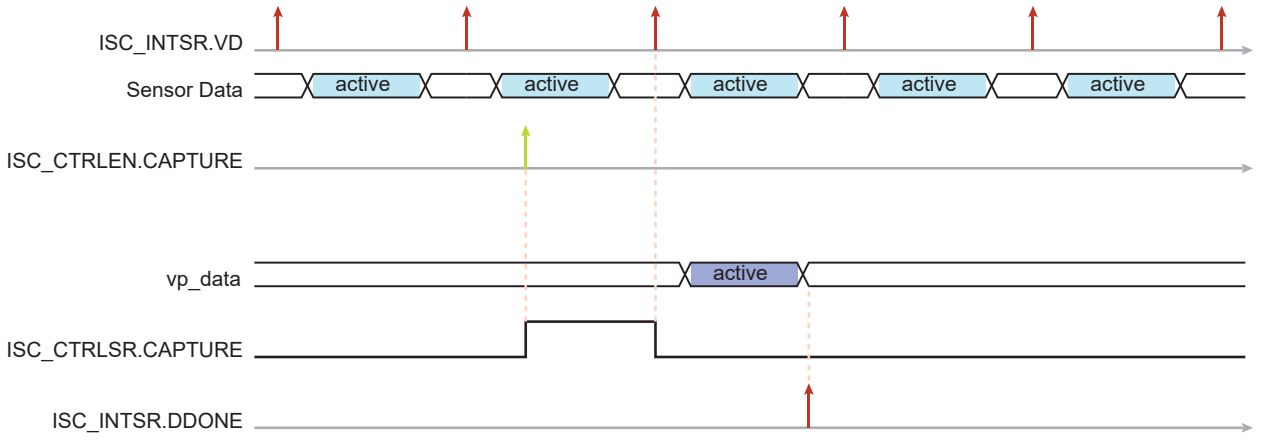
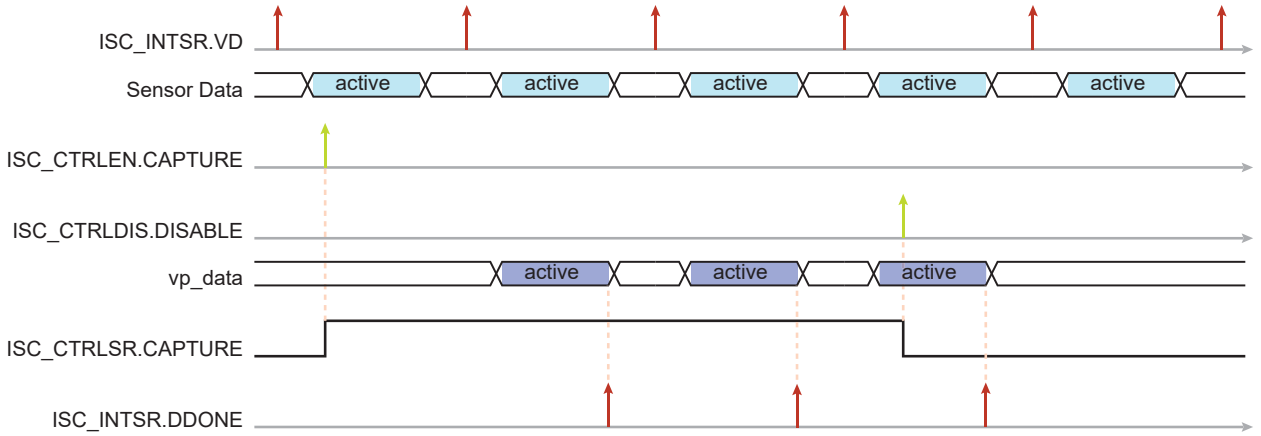
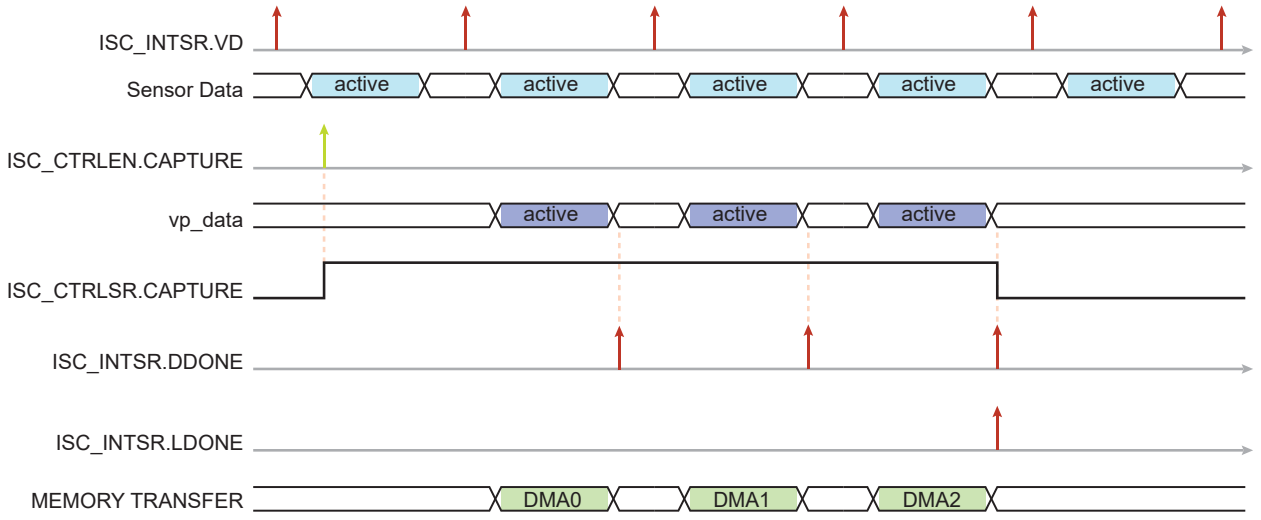


Figure 50.17. Continuous Acquisition Mode



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. ISC_INTSR.DDONE is set at the end of the DMA data transfer.

Figure 50.18. Continuous Acquisition, DMA Terminated



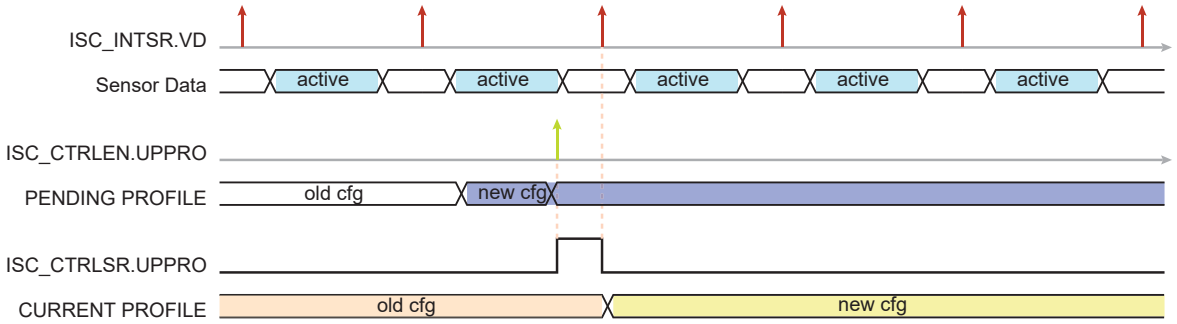
The linked list DMA transfer is terminated when an item of the list is programmed with ISC_DCTRL.DE cleared or when ISC_DNDA.NDA is equal to zero. This configuration also clears ISC_CTRLSR.CAPTURE and sets the ISC_INTSR.LDONE interrupt flag.

The linked list DMA transfer starts if ISC_DCTRL.DE is set and if ISC_DNDA.NDA is different from zero.

50.6.6.1. 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 50.19. Update Profile Timing Diagram



50.6.6.2.Software Requirements

Writing to ISC_CTRLEN or ISC_CTRLDIS requires a double domain synchronization, so it is forbidden to write these registers when ISC_CTRLSR.SIP is asserted.

50.6.7. Defective Pixel Correction (DPC)

The ISC can detect and correct defective pixels on Raw Bayer formats up to 12-bit width. 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.

50.6.8. 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.

The GDC module operates on Raw Bayer formats up to 12-bit width.

50.6.9. 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.

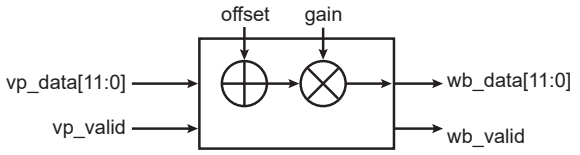
The BLC module operates on Raw Bayer formats up to 12-bit width.

50.6.10. 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 (up to 12-bit width supported), 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 50.20. WB Block Diagram



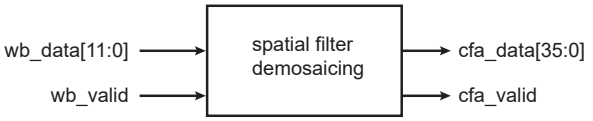
There are four {gain, offset} sets for each component. The output value is clipped.

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)

50.6.11. 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 50.21. 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.

50.6.11.1. 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

50.6.11.2.Frame Size Requirement when Edge Interpolation is On, ISC_CFA_CFG.EITPOL Set

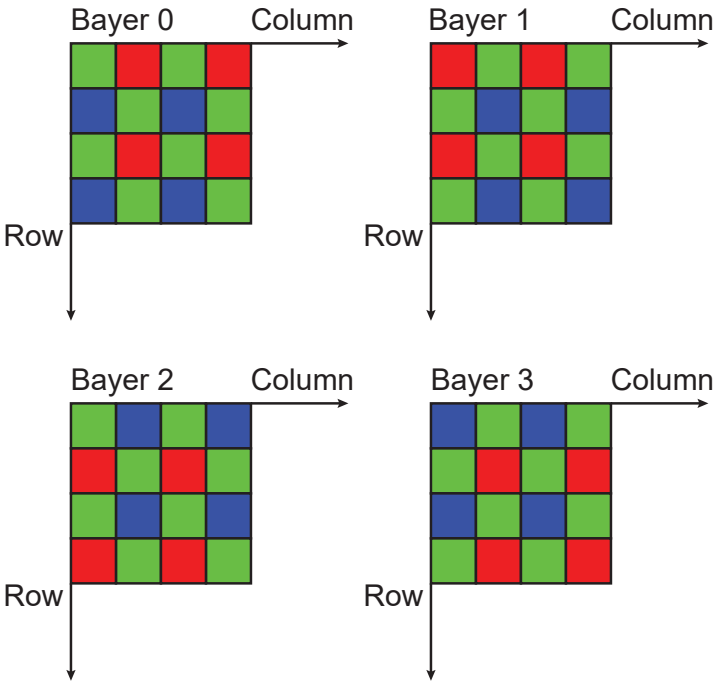
- Minimum number of rows (in): 3
- Minimum number of columns (in): 3
- Number of rows after CFA: Number of rows (in)
- Number of columns after CFA: Number of columns (in)

50.6.11.3.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.

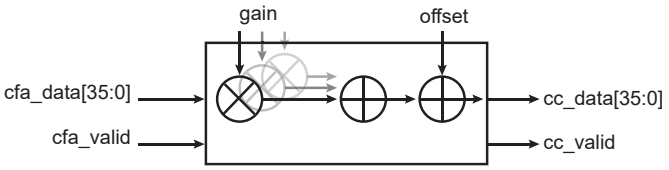
Figure 50.22. Supported Color Filter Array Patterns



50.6.12. 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 50.23. CC Block Diagram



There are three {gain, offset} sets for color component R, G, B.

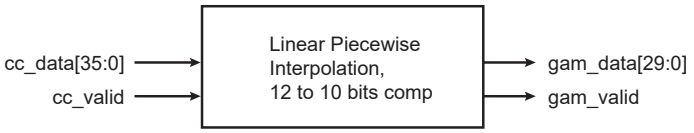
ISC_CC_CTRL.ENABLE	CC_DATA Slice	Value
0	cc_data[35:24]	cfa_data[11:0]
	cc_data[23:12]	cfa_data[11:0]
	cc_data[11:0]	cfa_data[11:0]
1	cc_data[35:24]	R=clipped(sum(cfa_data_x * gain_Rx) + offset_R)
	cc_data[23:12]	G=clipped(sum(cfa_data_x * gain_gx) + offset_g)
	cc_data[11:0]	B=clipped(sum(cfa_data_x * gain_Bx) + offset_B)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \text{RRGAIN} & \text{RGGAIN} & \text{RBGAIN} \\ \text{GRGAIN} & \text{GGGAIN} & \text{GBGAIN} \\ \text{BRGAIN} & \text{BGGAIN} & \text{BBGAIN} \end{bmatrix} \times \begin{bmatrix} \text{cfa\_data}[35:24] \\ \text{cfa\_data}[23:12] \\ \text{cfa\_data}[11:0] \end{bmatrix} + \begin{bmatrix} \text{ROFST} \\ \text{GOFST} \\ \text{BOFST} \end{bmatrix}$$

50.6.13. Gamma Curve (GAM) Module

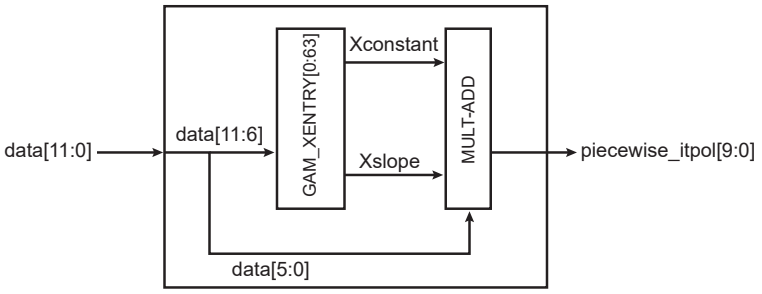
The GAM module samples the cc_data[35:0] bus when cc_valid is asserted, and generates gam_data[29:0] 30-bit width data along with the validity signal gam_valid. Imaging devices have non-linear characteristics, but the transfer function is approximated by a power function. The intensity of each of the 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 50.24. 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 50.25. 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.

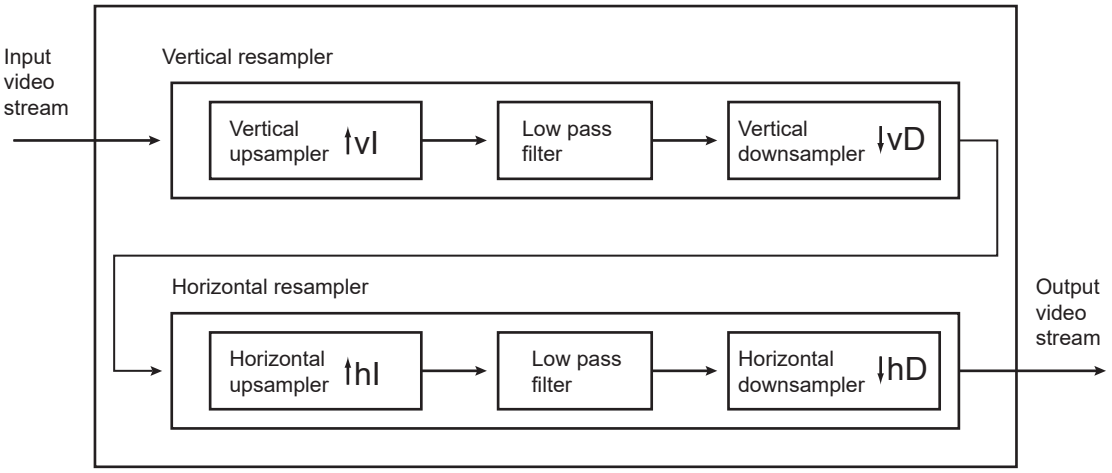
50.6.14. Scaler Function

The video pipeline includes a hardware scaler that allows an image resize in both the horizontal and the vertical directions. It integrates a two-tap or four-tap filter architecture with programmable polyphase coefficients or fixed bilinear interpolation coefficients.

50.6.14.1. Video Scaler Description

The scaling operation is based on a vertical and horizontal resampling algorithm. The sampling rate of the original image is increased when the video is upscaled, and decreased when the video is downscaled. The horizontal and vertical low pass filters are both designed to minimize the aliasing effect.

Figure 50.26. Video Resampler Architecture



The horizontal and vertical resamplers include a 16-phase 4-tap filter equivalent to a 64-tap FIR illustrated in the figures below.

Figure 50.27. Horizontal Resampler Filter Architecture

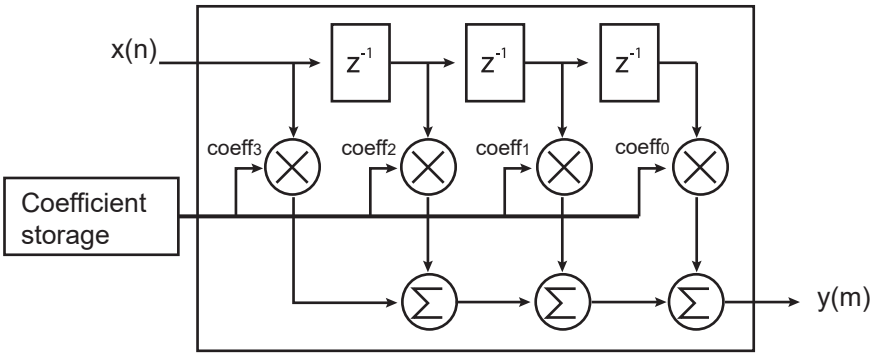
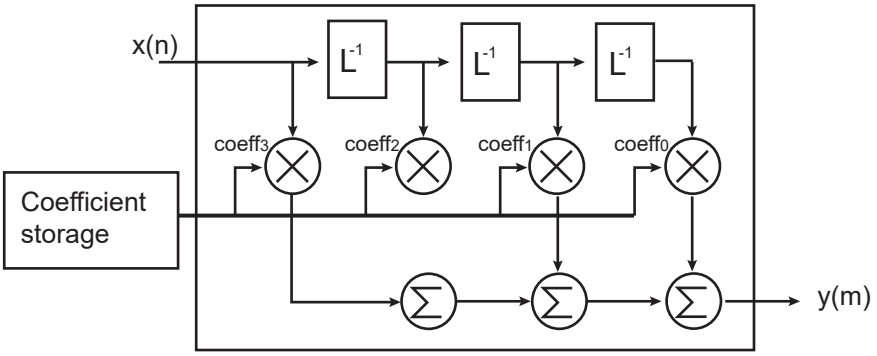


Figure 50.28. Vertical Resampler Filter Architecture



50.6.14.2.Horizontal Scaler Main Configuration

ISC_VHXS_SS.XS indicates the horizontal size minus one of the image in the system memory. ISC_VHXS_DS.XD contains the horizontal size minus one of the output image.

ISC_VHXS_CTRL.HXSEN is used to activate the horizontal scaler. The scaling factor is programmed in ISC_HXS_FACT.HFACT. The following equation calculates the horizontal scaling factor (HFACT) value:

$$\mathrm{HFACT} = \mathrm{round}\left(\frac{2^{20} \times \mathrm{IMAGE\_X\_SIZE\_IN}}{\mathrm{IMAGE\_X\_SIZE\_OUT}}\right)$$

The value HFACT is 24 bits wide (4-bit integer part, 20-bit fractional part), hence the horizontal downsampling capacity is limited to 16.0.

50.6.14.3.Vertical Scaler

ISC_VHXS_SS.YS indicates the vertical size minus one of the image in the system memory. ISC_VHXS_DS.YD contains the vertical size minus one of the output image.

ISC_VHXS_CTRL.VXSEN is used to activate the vertical scaler. The scaling factor is programmed in ISC_VXS_FACT.VFACT. The following equation calculates the vertical scaling factor (VFACT) value:

$$\mathrm{VFACT} = \mathrm{round}\left(\frac{2^{20} \times \mathrm{IMAGE\_Y\_SIZE\_IN}}{\mathrm{IMAGE\_Y\_SIZE\_OUT}}\right)$$

The value VFACT is 24 bits wide (4-bit integer part, 20-bit fractional part), hence the vertical downsampling capacity is limited to 16.0.

50.6.14.4.Input/Output Pixel Alignment

Depending on application requirements, output pixel alignment according to input pixels can be tuned for the horizontal and vertical scaler. This tuning modifies the position of the output image within the range of the filter length. Two different types of configuration fields are used, Filter Taps

Shift and Filter Init Phase Offset. Each of them produces a filter spatial response shift as described below.

Filter taps shift (T_{shift}) determines which set of input pixels is used at filter input to process a given output pixel. The following equation shows how the output pixel is calculated through the filter:

$$Pout\left(m\right) = \sum_{k=0}^{N-1} P_{in}\left(n + N - 1 - k - T_{shift}\right) \times h(k,p)$$

With :

P_{in}—input pixel component value (any R, G or B in RGB)

N—the filter number of taps (N=SCALER_FILT_NUM_PHASES)

T_{shift}—Filter taps shift

h(k,p)—Filter taps value depending on tap number k, and phase index p

P_{out}—output pixel component value (R, G or B in RGB mode)

The output image is shifted by the value T_{shift}/FACT, with FACT=HFACT for horizontal scaling and FACT=VFACT for vertical scaling. The following figures illustrate a use case of upsampling by 4 (VFACT=0.25x220):

Figure 50.29. Filter Taps Shift

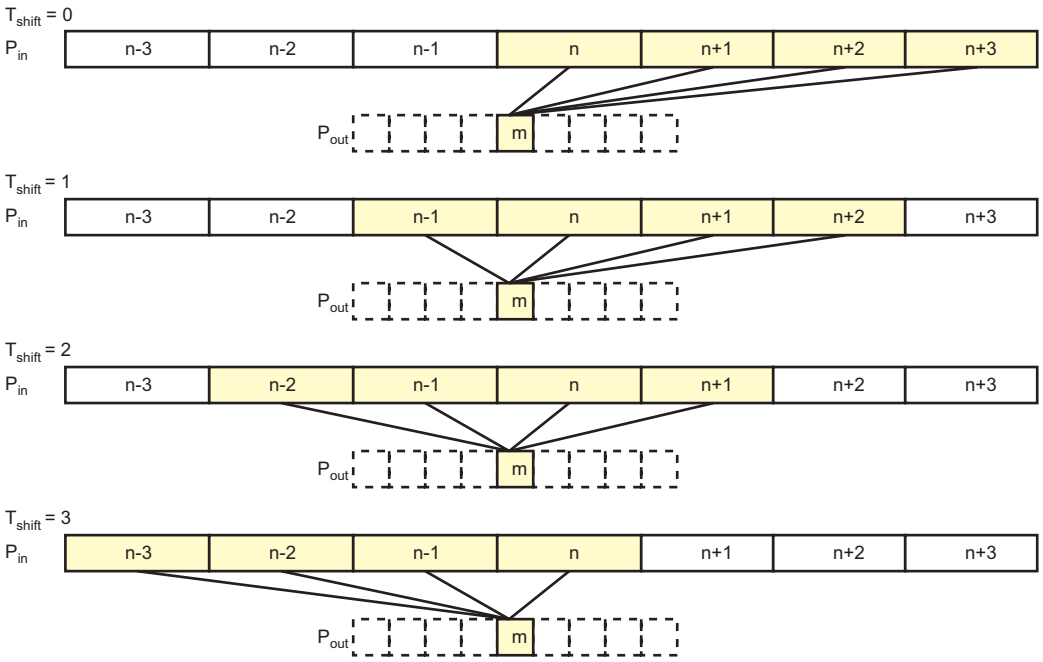


Figure 50.30. Filter Taps Shift Results for Different Horizontal T_{shift} Values (Upsampling by 4)

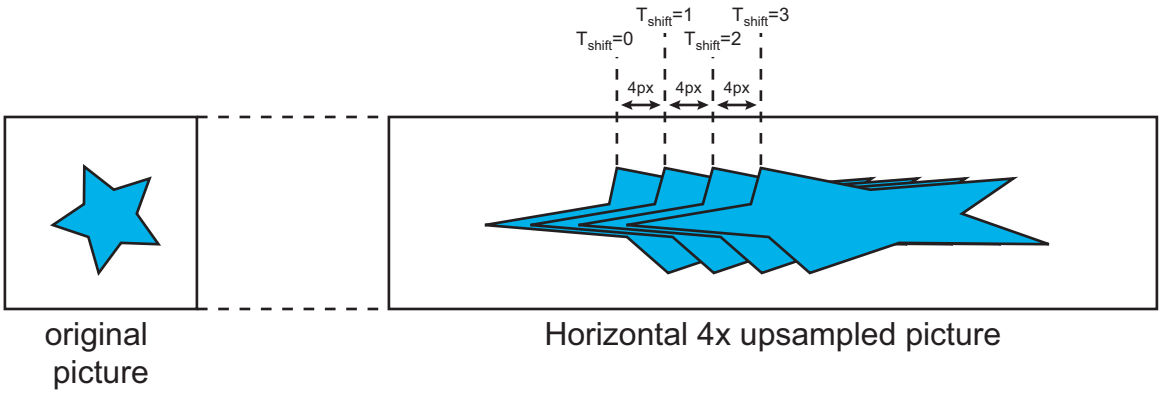
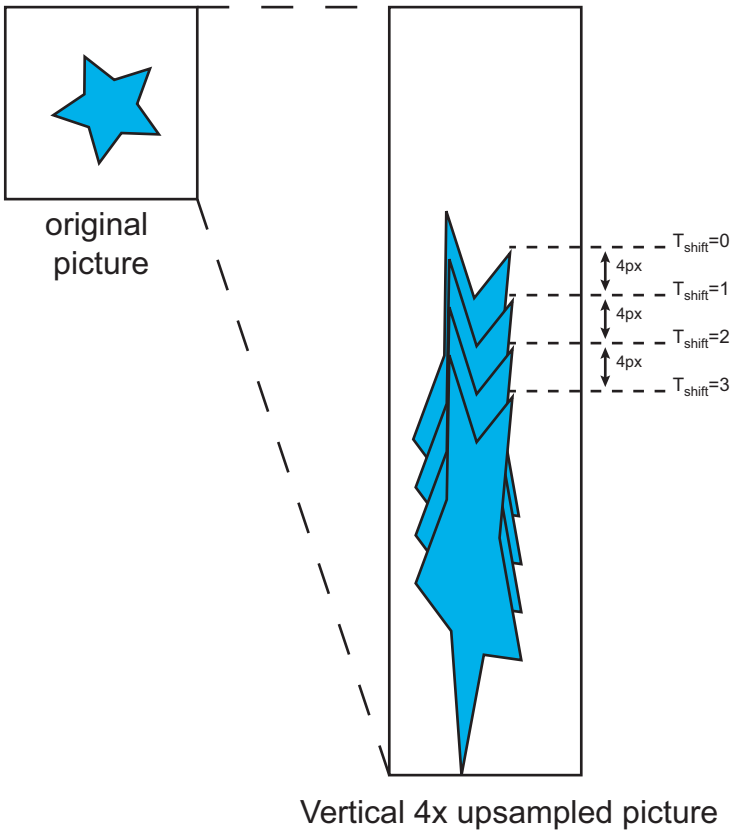


Figure 50.31. Filter Taps Shift Results for Different Vertical T_{shift} Values (Upsampling by 4)

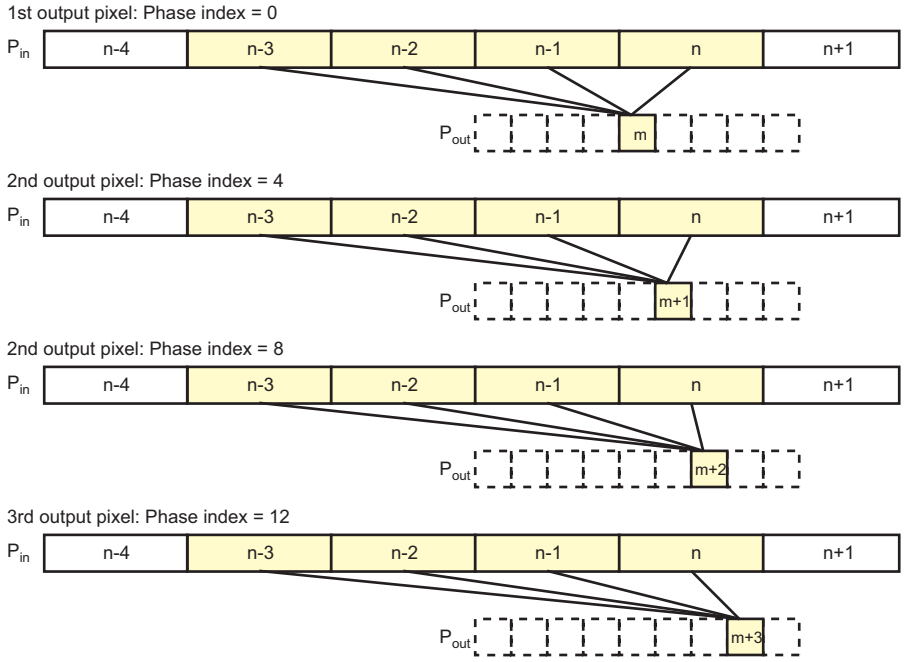


Filter taps shift (T_{shift}) is controlled by ISC_HXS_CFG.FILTCFG for the horizontal scaler, and ISC_VXS_CFG.FILTCFG for the vertical scaler.

During the resampling process, a filter tap phase is selected for each output pixel, representing the fractional part of the output pixel spatial position, according to the input pixel width. An input pixel width is subsampled in values of 16 phases, each phase p corresponding to a subset of filter taps. Phase 0 corresponds to 0 fractional shift. Incrementing the phase index by 1 corresponds to a spatial shift of 1/16 pixel width.

The following figures illustrate a use case of upsampling by 4 (four output pixels for one input pixel), with a fixed T_{shift} value.

Figure 50.32. Filter Phase Index (T_{shift} = 3)



A phase offset adds a fractional shift to all pixels of the frame. This is illustrated in the figures below for the use case when upsizing by 4.

Figure 50.33. Horizontal Initial Phase Offset On Upsizing by 4

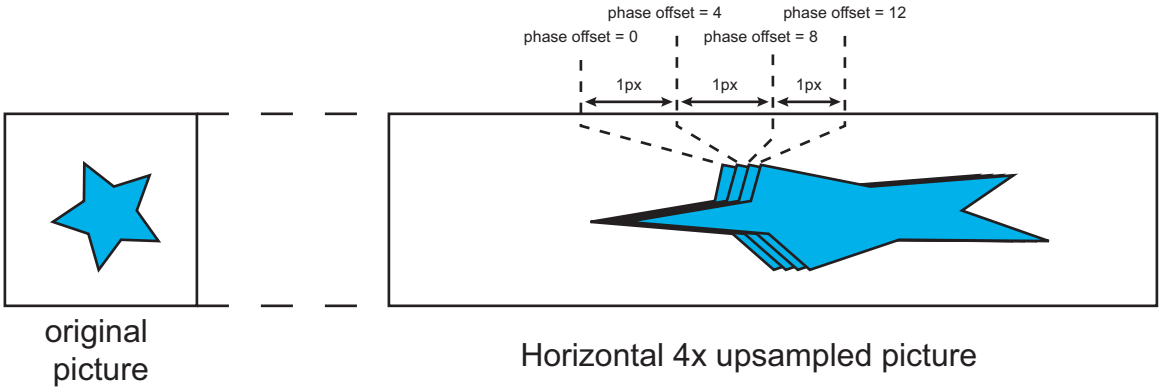
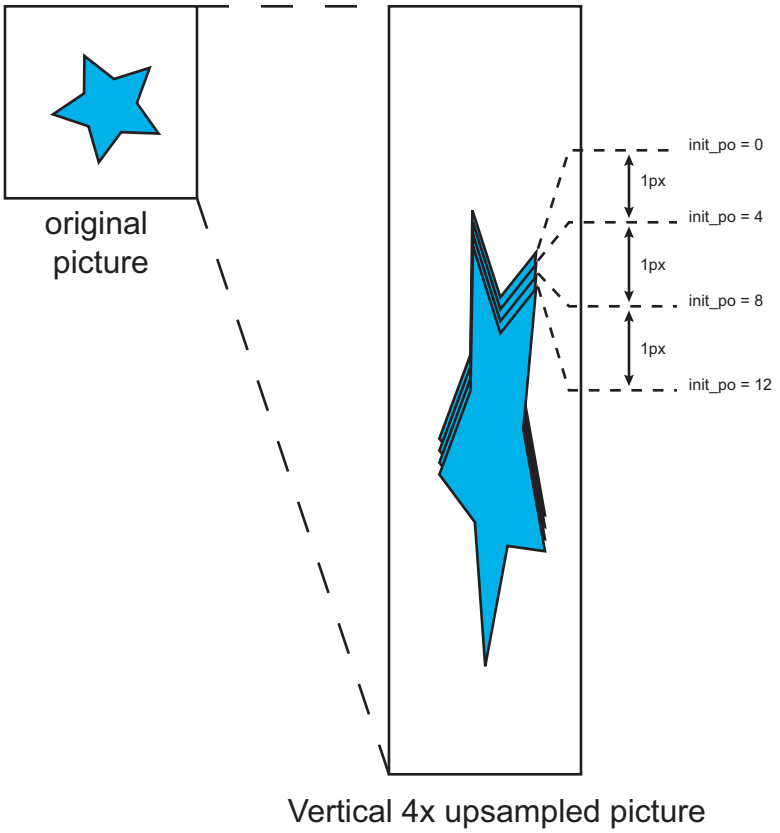


Figure 50.34. Vertical Initial Phase Offset on Upsizing by 4

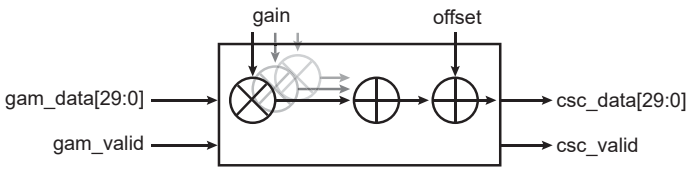


The initial phase offset is controlled by ISC_HXS_CFG.OFFSET for the horizontal scaler, and ISC_VXS_CFG.OFFSET for the vertical scaler.

50.6.15. 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 50.35. CSC Block Diagram



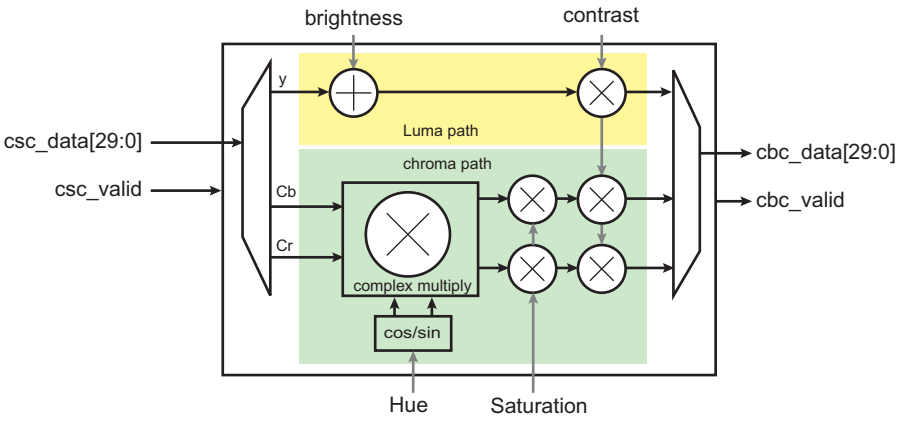
ISC_CSC_CTRL.ENABLE	CSC_DATA Slice	Value
0	csc_data[29:0]	gam_data[29:0]
1	csc_data[29:20]	Y = clipped(sum(gam_data_x * gain_Yx) + offset_y << 2)
	csc_data[19:10]	Cb = clipped(sum(gam_data_x * gain_Cbx)+offset_cb << 2)
	csc_data[9:0]	Cr = clipped(sum(gam_data_x * gain_Crx)+offset_cr << 2)

$$\begin{bmatrix} Y \\ CB \\ CR \end{bmatrix} = \begin{bmatrix} YR & YG & YB \\ CBR & CBG & CBB \\ CRR & CRG & CRB \end{bmatrix} \times \begin{bmatrix} gam\_data[29:20] \\ gam\_data[19:10] \\ gam\_data[9:0] \end{bmatrix} + \begin{bmatrix} YOFST \\ CBOFST \\ CROFST \end{bmatrix}$$

50.6.16. Contrast, Brightness, Hue and Saturation

This module is for YUV formatting purposes. 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 50.36. CBHS Block Diagram

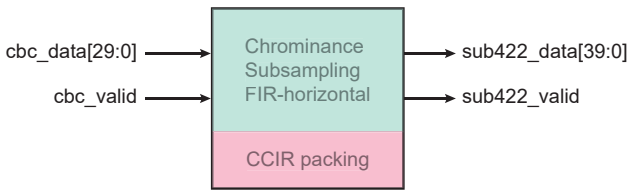


ISC_CBC_CTRL.ENABLE	ISC_CBC_CFG.CCIR	CBC_DATA Slice	Value
0	0	cbc_data[29:0]	csc_data[29:0]
1	0	cbc_data[29:20]	Y = clipped(contrast * (csc_data[29:20]+brightness))
		cbc_data[19:10]	Cb = clipped(saturation * contrast * (csc_data[19:10]*cos(hue)+csc_data[9:0]*sin(hue))
		cbc_data[9:0]	Cr = clipped(saturation * 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

50.6.17. 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 50.37. 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 50.38. Cosited Filter Configuration

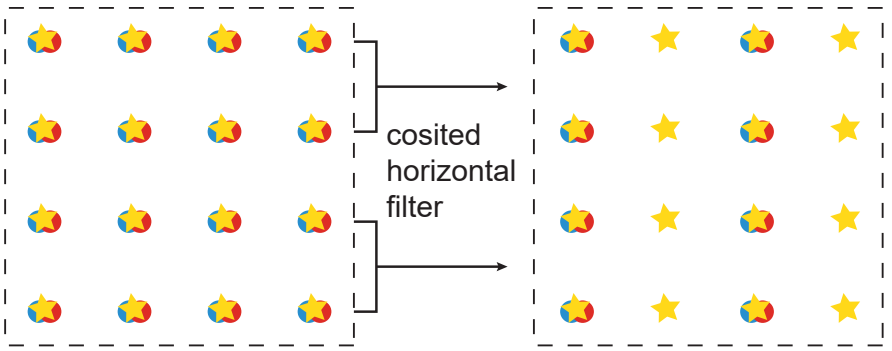
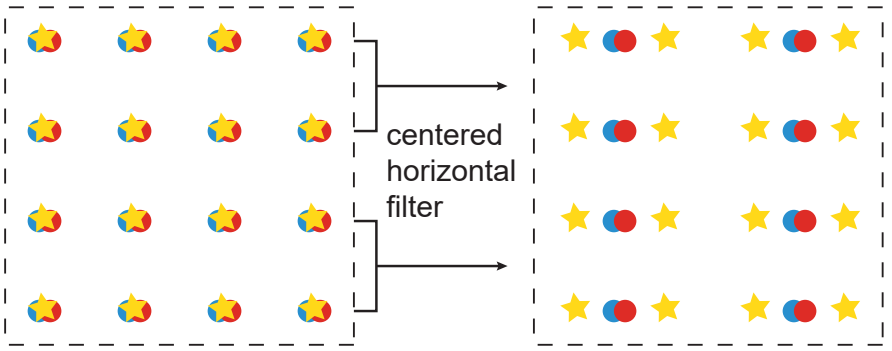


Figure 50.39. Centered Filter Configuration



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.

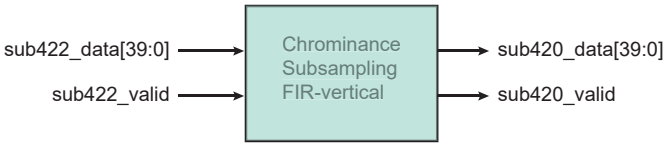
SUB422_DATA Slice	Even Line Length	Odd Line Length
sub422_data[39:30]	Y(n)	Y(n-1)

4:4:4 To 4:2:2 Chrominance Horizontal Subsampler (SUB422) Module (continued)		
SUB422_DATA Slice	Even Line Length	Odd Line Length
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)

50.6.18. 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 50.40. 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]
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 50.41. Vertical Chrominance Filter for Progressive Content (Cosited Chrominance Example)

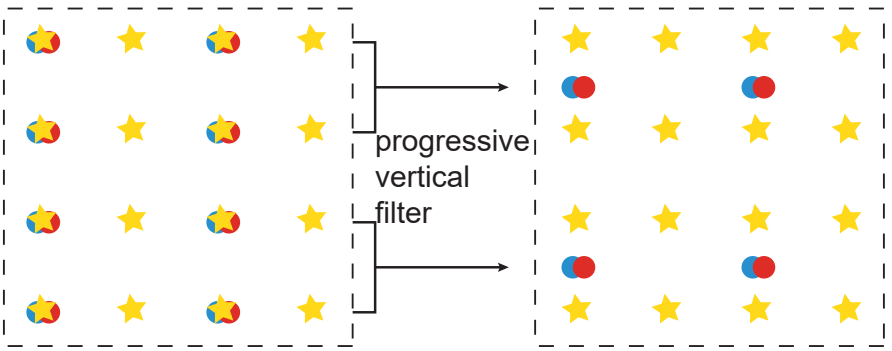


Figure 50.42. Field-dependent Chrominance Filter for Interlaced Content (Cosited Chrominance Example)

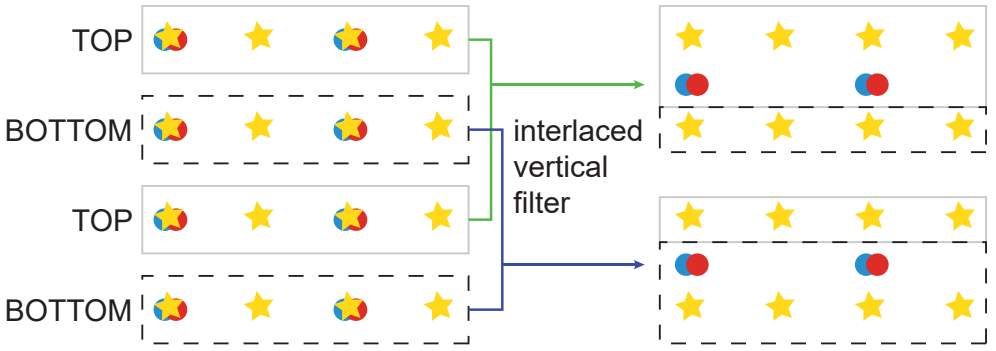


Table 50.2. Filter Configuration

ISC_SUB420_CTRL.FILTER	Field	Filter Configuration
0	progressive	{1, 1}
1	0 (TOP)	{3, 1}
	1 (BOTTOM)	{1, 3}

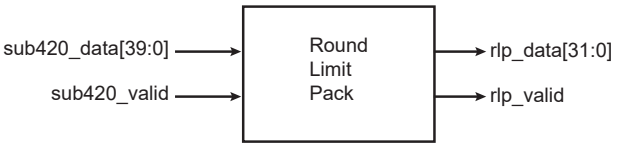
Table 50.3. Output Line Length Configuration

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

50.6.19. 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 50.43. 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[31:10]	0
	rlp_data[9:0]	Y = sub420_data[29:20])

Rounding, Limiting and Packing (RLP) Module (continued)		
ISC_RLP_CFG	RLP_DATA Slice	Value
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]
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]

The round_limit function provides a simple method to round and limit the range of both Luminance and Chrominance signals.

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

50.6.20. 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 ISC_DCTRL, and sets ISC_DCTRL.DONE to one and ISC_DCTRL.FIELD to the value of the frame field when interlaced content is being used. This 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 50.44. DMA Host 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]
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.

50.6.20.1.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

50.6.20.2.Descriptor Memory Mapping

Three descriptor views are available :

- Descriptor view 0: used when the pixel or data stream is packed
- Descriptor view 1: used for YCbCr semi-planar pixel stream
- Descriptor view 2: used for YCbCr planar pixel stream

Table 50.4. ISC_DCTRL.DVIEW = 0

Address	Register
<current descriptor address>+0x00	ISC_DCTRL
<current descriptor address>+0x04	ISC_DNDA
<current descriptor address>+0x08	ISC_DAD0
<current descriptor address>+0x0C	ISC_DST0

Table 50.5. ISC_DCTRL.DVIEW = 1

Address	Register
<current descriptor address>+0x00	ISC_DCTRL
<current descriptor address>+0x04	ISC_DNDA
<current descriptor address>+0x08	ISC_DAD0
<current descriptor address>+0x0C	ISC_DST0
<current descriptor address>+0x10	ISC_DAD1
<current descriptor address>+0x14	ISC_DST1

Table 50.6. ISC_DCTRL.DVIEW = 2

Address	Register
<current descriptor address>+0x00	ISC_DCTRL
<current descriptor address>+0x04	ISC_DNDA
<current descriptor address>+0x08	ISC_DAD0
<current descriptor address>+0x0C	ISC_DST0
<current descriptor address>+0x10	ISC_DAD1
<current descriptor address>+0x14	ISC_DST1
<current descriptor address>+0x18	ISC_DAD2
<current descriptor address>+0x1C	ISC_DST2

50.6.20.3.Example: Memory Mapping for 16-bit Packed, DMA Interface IMODE = 1 at ISC_DAD0.AD0 Location

Table 50.7. 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 50.8. 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 50.9. 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 50.10. 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 50.11. 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]											

50.6.20.4.Example: Memory Mapping for 12-bit YC420SP, DMA Interface IMODE = 5

Table 50.12. 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 50.13. 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]															

50.6.20.5.Example: Memory Mapping for 12-bit YC420P, DMA Interface IMODE = 6

Table 50.14. 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 50.15. 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 50.16. 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]							

50.6.21. 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 50.45. Histogram Block Diagram

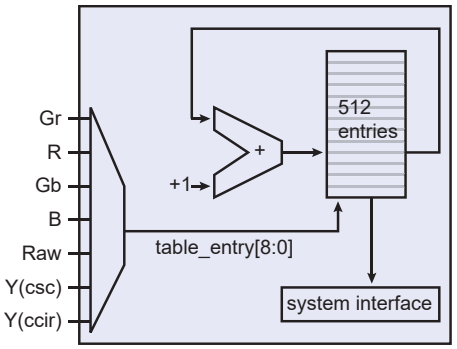
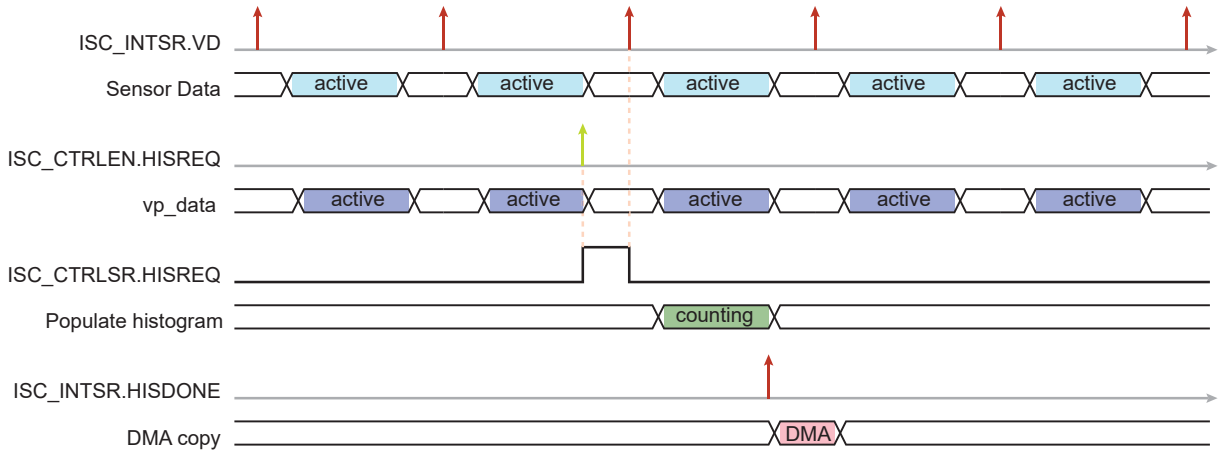


Figure 50.46. Histogram Request Timing Diagram



50.6.22. Register Write Protection

To prevent any single software error from corrupting ISC behavior, certain registers in the address space can be write-protected by setting the bits WPCFGEN, WPITEN, WPCREN in ISC_WPMR.

If a write access to a write-protected register is detected, the Write Protection Violation Status (WPVS) flag in ISC_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 ISC_WPSR.

50.7. 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
0x38 ... 0x3F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]	
0x78	ISC_CC_CTRL	31:24								
		23:16								
		15:8								
		7:0								ENABLE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]							
0x0398	ISC_VHXS_CTRL	31:24								
		23:16								
		15:8								
		7:0							HXSEN	VXSEN

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]			
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]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]				
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]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]			
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]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]				
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]				

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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_CONT	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]							
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									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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								
		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								
		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]							
0x053C	ISC_DST2	31:24								
		23:16								
		15:8	ST2[15:8]							
		7:0	ST2[7:0]							
0x0540	ISC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCREN	WPITEN	WPCFGEN
0x0544	ISC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
0x0548 ... 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]							

50.7.1. 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.

This register can only be written if WPCREN is cleared in ISC_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							FUPPRO	
Reset							W	
							–	

Bit	7	6	5	4	3	2	1	0
Access					HISCLR	HISREQ	UPPRO	CAPTURE
Reset					W	W	W	W
					–	–	–	–

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

50.7.2. 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.

This register can only be written if WPCREN is cleared in ISC_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
								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

50.7.3. 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.

Value	Description
1	Capture is pending.

50.7.4. ISC Parallel Front End Configuration 0 Register

Name: ISC_PFE_CFG0
Offset: 0x0C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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 come from the physical parallel interface.
1	Input data come 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.

50.7.5. ISC Parallel Front End Configuration 1 Register

Name: ISC_PFE_CFG1
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.6. ISC Parallel Front End Configuration 2 Register

Name: ISC_PFE_CFG2
Offset: 0x14
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.7. ISC Clock Enable Register

Name: ISC_CLKEN
Offset: 0x18
Reset: –
Property: Write-only

This register can only be written if WPCREN is cleared in ISC_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
							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.

50.7.8. 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.

This register can only be written if WPCREN is cleared in ISC_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							W	W
Reset							–	–

Bit	7	6	5	4	3	2	1	0
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

50.7.9. ISC Clock Status Register

Name: ISC_CLKSR
Offset: 0x20
Reset: 0x00000001
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	1

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.

50.7.10. ISC Clock Configuration Register

Name: ISC_CLKCFG
Offset: 0x24
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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

$$f_{mc} = \frac{f_{mcref}}{MCDIV + 1}$$

50.7.11. 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.

This register can only be written if WPITEN is cleared in ISC_WPMR.

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

50.7.12. 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.

This register can only be written if WPITEN is cleared in ISC_WPMR.

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

50.7.13. 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 corresponding source of interrupt is disabled.

1: The corresponding source of 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

50.7.14. 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.

Value	Description
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.
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.

50.7.15. ISC Defective Pixel Control Register

Name: ISC_DPC_CTRL
Offset: 0x40
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
						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.

50.7.16. ISC Defective Pixel Configuration Register

Name: ISC_DPC_CFG
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.17. ISC Defective Pixel Correction Median Threshold Register

Name: ISC_DPC_THRESHM
Offset: 0x48
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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

50.7.18. ISC Defective Pixel Correction Closest Threshold Register

Name: ISC_DPC_THRESHC
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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

50.7.19. ISC Defective Pixel Correction Average Threshold Register

Name: ISC_DPC_THRESHA
Offset: 0x50
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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

50.7.20. 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)
Shows the number of active pixel substitutions in the previous frame. It is updated on a frame-by-frame basis.

50.7.21. ISC White Balance Control Register

Name: ISC_WB_CTRL
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
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.

50.7.22. ISC White Balance Configuration Register

Name: ISC_WB_CFG
Offset: 0x5C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
							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).

50.7.23. ISC White Balance Offset for R, GR Register

Name: ISC_WB_O_RGR
Offset: 0x60
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.24. ISC White Balance Offset for B and GB Register

Name: ISC_WB_O_BGB
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.25. ISC White Balance Gain for R, GR Register

Name: ISC_WB_G_RGR
Offset: 0x68
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.26. ISC White Balance Gain for B, GB Register

Name: ISC_WB_G_BGB
Offset: 0x6C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.27. ISC Color Filter Array Control Register

Name: ISC_CFA_CTRL
Offset: 0x70
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
								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.

50.7.28. ISC Color Filter Array Configuration Register

Name: ISC_CFA_CFG
Offset: 0x74
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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).

50.7.29. ISC Color Correction Control Register

Name: ISC_CC_CTRL
Offset: 0x78
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
								ENABLE
Access								R/W
Reset								0

Bit 0 – ENABLE Color Correction Enable

Value	Description
0	Color correction is disabled.
1	Color correction is enabled.

50.7.30. ISC Color Correction RR RG Register

Name: ISC_CC_RR_RG
Offset: 0x7C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.31. ISC Color Correction RB OR Register

Name: ISC_CC_RB_OR
Offset: 0x80
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.32. ISC Color Correction GR GG Register

Name: ISC_CC_GR_GG
Offset: 0x84
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.33. ISC Color Correction GB OG Register

Name: ISC_CC_GB_OG
Offset: 0x88
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.34. ISC Color Correction BR BG Register

Name: ISC_CC_BR_BG
Offset: 0x8C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.35. ISC Color Correction BB OB Register

Name: ISC_CC_BB_OB
Offset: 0x90
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.36. ISC Gamma Correction Control Register

Name: ISC_GAM_CTRL
Offset: 0x94
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
				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.

50.7.37. 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

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.38. 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

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.39. 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

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.40. ISC VHXS Control Register

Name: ISC_VHXS_CTRL
Offset: 0x398
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
							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.

50.7.41. ISC VHXS Source Size Register

Name: ISC_VHXS_SS
Offset: 0x39C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.42. ISC VHXS Destination Size Register

Name: ISC_VHXS_DS
Offset: 0x3A0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.43. ISC VXS Scaling Factor Register

Name: ISC_VXS_FACT
Offset: 0x3A4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.44. ISC HXS Scaling Factor Register

Name: ISC_HXS_FACT
Offset: 0x3A8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.45. ISC VXS Configuration Register

Name: ISC_VXS_CFG
Offset: 0x3AC
Reset: 0x80000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.46. ISC HXS Configuration Register

Name: ISC_HXS_CFG
Offset: 0x3B0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.47. ISC VXS TAP10 Phase x Register

Name: ISC_VXS_TAP10PHIx
Offset: 0x03B4 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.48. ISC VXS TAP32 Phase x Register

Name: ISC_VXS_TAP32PHIx
Offset: 0x03B8 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.49. ISC HXS TAP10 Phase x Register

Name: ISC_HXS_TAP10PHIx
Offset: 0x0434 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.50. ISC HXS TAP32 Phase x Register

Name: ISC_HXS_TAP32PHIx
Offset: 0x0438 + x*0x08 [x=0..15]
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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.

50.7.51. ISC Color Space Conversion Control Register

Name: ISC_CSC_CTRL
Offset: 0x4B4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
								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.

50.7.52. ISC Color Space Conversion YR YG Register

Name: ISC_CSC_YR_YG
Offset: 0x4B8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.53. ISC Color Space Conversion YB OY Register

Name: ISC_CSC_YB_OY
Offset: 0x4BC
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.54. ISC Color Space Conversion CBR CBG Register

Name: ISC_CSC_CBR_CBG
Offset: 0x4C0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.55. ISC Color Space Conversion CBB OCB Register

Name: ISC_CSC_CBB_OCB
Offset: 0x4C4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.56. ISC Color Space Conversion CRR CRG Register

Name: ISC_CSC_CRR_CRG
Offset: 0x4C8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.57. ISC Color Space Conversion CRB OCR Register

Name: ISC_CSC_CRB_OCR
Offset: 0x4CC
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_WPMR.

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)

50.7.58. ISC Control Register

Name: ISC_CBHS_CTRL
Offset: 0x4D0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
								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.

50.7.59. ISC CBHS Configuration Register

Name: ISC_CBHS_CFG
Offset: 0x4D4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
						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.

50.7.60. ISC CBHS Brightness Register

Name: ISC_CBHS_BRIGHT
Offset: 0x4D8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
						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.

50.7.61. ISC CBHS Contrast Register

Name: ISC_CBHS_CONT
Offset: 0x4DC
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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.

50.7.62. ISC Hue Register

Name: ISC_CBHS_HUE
Offset: 0x4E0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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								R/W
Reset								0

Bit	7	6	5	4	3	2	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 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.

50.7.63. ISC CBHS Saturation Register

Name: ISC_CBHS_SAT
Offset: 0x4E4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
					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.

50.7.64. ISC Subsampling 4:4:4 to 4:2:2 Control Register

Name: ISC_SUB422_CTRL
Offset: 0x4E8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
								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.

50.7.65. ISC Subsampling 4:4:4 to 4:2:2 Configuration Register

Name: ISC_SUB422_CFG
Offset: 0x4EC
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
			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.

50.7.66. ISC Subsampling 4:2:2 to 4:2:0 Control Register

Name: ISC_SUB420_CTRL
Offset: 0x4F0
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
			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.

50.7.67. ISC Rounding, Limiting and Packing Configuration Register

Name: ISC_RLP_CFG
Offset: 0x4F4
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
	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
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.

50.7.68. ISC Histogram Control Register

Name: ISC_HIS_CTRL
Offset: 0x4F8
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
Access								R/W
Reset								0

Bit 0 – ENABLE Histogram Sub Module Enable

Value	Description
0	Histogram disabled.
1	Histogram enabled.

50.7.69. ISC Histogram Configuration Register

Name: ISC_HIS_CFG
Offset: 0x4FC
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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								RAR
Reset								R/W
								0

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

50.7.70. 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
3	YC422SP	32 bits, dual channel
4	YC422P	32 bits, triple channel
5	YC420SP	32 bits, dual channel
6	YC420P	32 bits, triple channel

50.7.71. ISC DMA Control Register

Name: ISC_DCTRL
Offset: 0x520
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCREN is cleared in ISC_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
	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
Appears in the descriptor located in memory 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
Only relevant for interlaced content.
Appears in the descriptor located in memory 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	Packed frame buffer (see ISC_DCTRL.DVIEW = 0)
1	SEMIPLANAR	Semi planar frame buffer (see ISC_DCTRL.DVIEW = 1)
2	PLANAR	Planar frame buffer (see ISC_DCTRL.DVIEW = 2)
3	–	Reserved

Bit 0 – DE Descriptor Enable

Value	Description
0	Descriptor disabled.
1	Descriptor enabled.

50.7.72. 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

50.7.73. 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

50.7.74. ISC DMA Stride 0 Register

Name: ISC_DST0
Offset: 0x52C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
	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 15:0 – ST0[15:0] Channel 0 Stride

50.7.75. 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

50.7.76. ISC DMA Stride 1 Register

Name: ISC_DST1
Offset: 0x534
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
	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 15:0 – ST1[15:0] Channel 1 Stride

50.7.77. 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

50.7.78. ISC DMA Stride 2 Register

Name: ISC_DST2
Offset: 0x53C
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFGEN is cleared in ISC_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
	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 15:0 – ST2[15:0] Channel 2 Stride

50.7.79. 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

50.7.80. ISC Write Protection Mode Register

Name: ISC_WPMR
Offset: 0x540
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	WPCFGEN
Access						R/W	R/W	R/W
Reset						0	0	0

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

Value	Name	Description
0x584953	PASSWD	Writing any other value in this field aborts the write operation of the WPCFGEN,WPITEN,WPCREN bits. Always reads as 0.

Bit 2 – WPCREN Write Protection Control Registers Enable

Value	Description
0	Disables the write protection of control registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).
1	Enables the write protection of control enable/disable registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).

Bit 1 – WPITEN Write Protection Interrupt Registers Enable

Value	Description
0	Disables the write protection of interrupt enable/disable registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).
1	Enables the write protection of interrupt enable/disable registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).

Bit 0 – WPCFGEN Write Protection Configuration Registers Enable

Value	Description
0	Disables the write protection of configuration registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).
1	Enables the write protection of configuration registers if WPKEY corresponds to 0x584953 (“XIS” in ASCII).

50.7.81. ISC Write Protection Mode Register

Name: ISC_WPSR
Offset: 0x544
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/W	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
	WPVSRC[7:0]							
Access	R/W	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
								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 occurred since the last read of ISI_WPSR.
1	A write protection violation has occurred since the last read of ISI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Audio Subsystem

51. Overview

51.1. Components

The audio subsystem is made of the following components. Unless otherwise specified, all except the CLASSD controller are designed to handle audio sample rates up to 192 kHz.

- 1x 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.
- 1x Inter-IC Sound Multi Channel Controller (I2SMCC)
  - The maximum bit clock (I2SMCC_SCK) speed is 25 MHz in both Host and Client modes.
  - In TDM256 mode, this corresponds to 8x 32-bit channels at 96 kHz.
- 1x Low-jitter audio PLL with one dedicated AUDIOCLK output pin
- 1x Stereo Class D Controller (CLASSD)
  - Fixed PWM frequency operation in the 750 to 800 kHz range
  - Targeted to low-end audio applications
  - Embedded interpolation filters to accommodate standard audio sampling rates

51.2. Product Dependencies

51.2.1. Clocks

To generate audio frequencies with high accuracy, the device embeds a fractional audio PLL that can serve both the internal (GCLK generation used for CLASSD) and external needs (MCK generation on external audio components).

51.2.2. Interrupts

Refer to the table Peripheral Identifiers.

51.2.3. Reset

Audio peripherals are connected to the processor and peripherals reset line.

51.2.4. 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.

51.3. Special Functions in SFR

None.

52. Audio Class D Amplifier (CLASSD)

52.1. Description

The Audio Class D Amplifier (CLASSD) is a digital input, Pulse Width Modulated (PWM) output stereo Class D amplifier. It features a high-quality interpolation filter embedding a digitally-controlled gain, an equalizer and a de-emphasis filter.

On its input side, the CLASSD is compatible with most common audio data rates. On the output side, its PWM output can drive either:

- high-impedance single-ended or differential output loads (Audio DAC application) or,
- external MOSFETs through an integrated non-overlapping circuit (Class D power amplifier application).

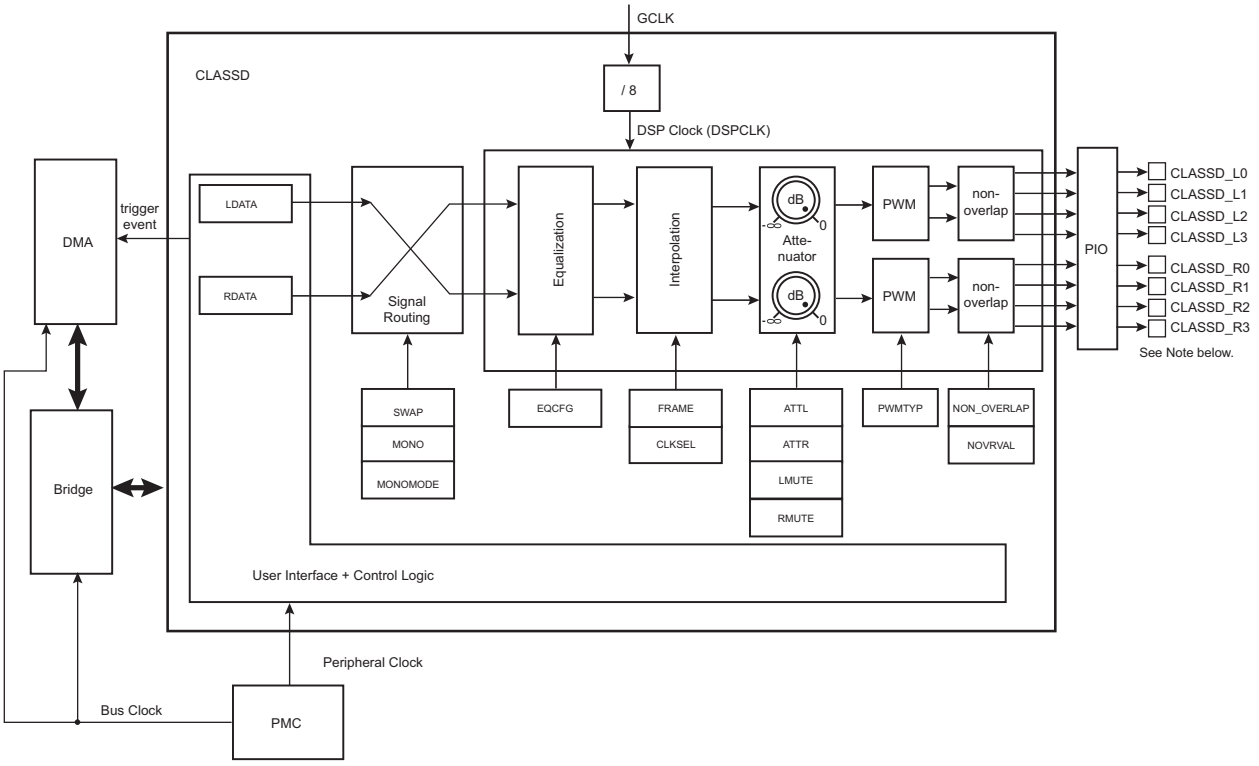
Note: CLASSD is stereo but depending on the available I/O at the product level, only one (right or left) channel may be accessed. Refer to the product "Pin Description" table.

52.2. Embedded Characteristics

- PWM Class D Amplifier
- 16-bit Audio Data
- DSP Clocks: 12.288 and 11.2896 MHz
- Input Sampling Rates: 8, 16, 32, 48, 96, 22.05, 44.1, 88.2 kHz
- 3-band Equalizer
- De-emphasis Filter
- Digital Volume Control
- Differential or Single-ended Outputs
- Non-overlapping Circuit to Control External MOSFETs
- Supports DMA

52.3. Block Diagram

Figure 52.1. CLASSD Block Diagram



Note:
CLASSD is stereo but depending on the available I/O at the product level, only one (right or left) channel may be accessed.
Refer to the product pin description table.

52.4. Pin Name List

Table 52.1. Output Pins Assignment Versus Application Use Cases

Pin	External MOS Driver (NON_OVERLAP = 1)		Direct Load (NON_OVERLAP = 0)		Type
	Full H-Bridge (PWMTYP = 1)	Half H-Bridge (PWMTYP = 0)	Differential Load (PWMTYP = 1)	Single-Ended Load (PWMTYP = 0)	
	Use Case 1	Use Case 2	Use Cases 3A & 3B	Use Cases 4A & 4B	
CLASSD_L0	gate_pmos_leftp	gate_pmos_left	leftp	left	Output
CLASSD_L1	gate_nmos_leftp	gate_nmos_left	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_L2	gate_pmos_leftn	Not used (fixed to 1)	leftn	Not used (fixed to 0)	Output
CLASSD_L3	gate_nmos_leftn	Not used (fixed to 1)	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_R0	gate_pmos_rightp	gate_pmos_right	rightp	right	Output
CLASSD_R1	gate_nmos_rightp	gate_nmos_right	Not used (fixed to 0)	Not used (fixed to 0)	Output
CLASSD_R2	gate_pmos_rightn	Not used (fixed to 1)	rightn	Not used (fixed to 0)	Output
CLASSD_R3	gate_nmos_rightn	Not used (fixed to 1)	Not used (fixed to 0)	Not used (fixed to 0)	Output

52.5. Product Dependencies

52.5.1. 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 CLASSD pins to their peripheral functions.

52.5.2. Power Management

The CLASSD is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the CLASSD Peripheral Clock and provide a generic clock (GCLK).

The fields NOVRVAL, NON_OVERLAP and PWMTYP in CLASSD_MR, and DSPCLKFREQ and FREQ in CLASSD_INTPMR, must be configured prior to applying the GCLK.

52.5.3. Interrupt

The CLASSD has an interrupt line connected to the interrupt controller. Handling the CLASSD interrupt requires programming the interrupt controller before configuring the CLASSD.

52.6. Functional Description

52.6.1. Interpolator

52.6.1.1. Clock Configuration

The interpolator accepts input sampling frequencies (f_s) and the input DSP clock (DSPCLK) that can be configured in the CLASSD Interpolator Mode Register. GCLK must be configured in the PMC according to the desired DSPCLK so that DSPCLK = GCLK / 8.

The following table provides authorized DSPCLK / f_s ratios and associated filter types.

Table 52.2. Authorized DSPCLK / f_s Ratios & Filter Types

f_s	DSPCLK	
	12.288 MHz	11.2896 MHz
8 kHz	2	–
16 kHz	2	–
32 kHz	2	–
48 kHz	1	–
96 kHz	3	–
22.05 kHz	–	1
44.1 kHz	–	1
88.2 kHz	–	3

Note: Each dash (–) indicates a configuration that is not authorized and that raises the CFGERR flag in CLASSD_INTSR.

52.6.1.2. CLASSD Frequency Response

Interpolation is performed with a combination of Infinite Impulse Response (IIR) and Cascaded Integrator-Comb (CIC) filters. Given the input configuration, the coefficients of the filters are redefined to optimize their transfer function to optimize the audio bandwidth. The different types of filters are defined in section Clock Configuration.

Figure 52.2. Type 1 Frequency Response

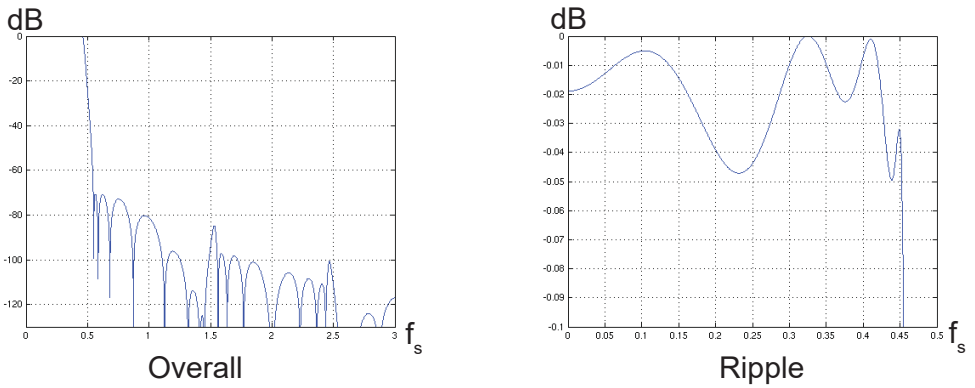


Figure 52.3. Type 2 Frequency Response

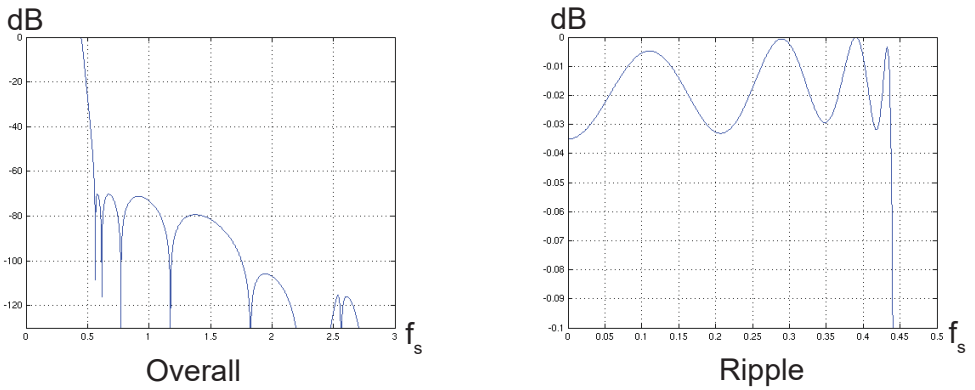
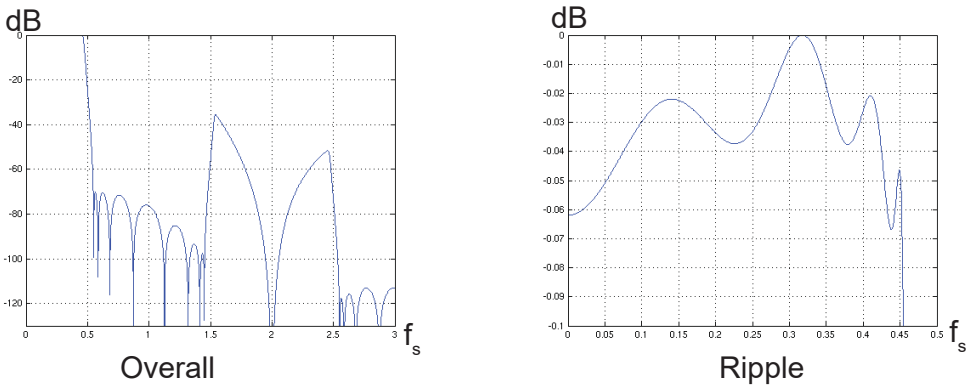


Figure 52.4. Type 3 Frequency Response



52.6.2. Equalizer

The CLASSD offers 12 pre-programmed equalization filters.

A zero-cross detection system is used to modify the equalizer on-the-fly with minimum disturbance on the output signal.

Programming of the equalization filter is detailed in section CLASSD Interpolator Mode Register.

The following figures show the frequency response of the equalizer function implemented in the D/A channels.

Figure 52.5. Bass Filters Response

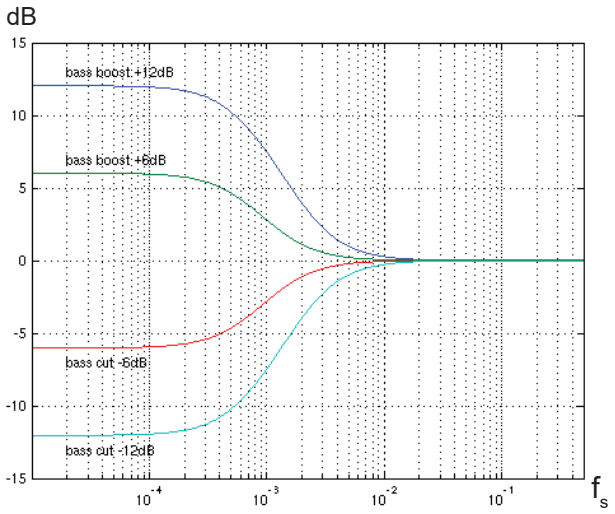


Figure 52.6. Medium Filters Response

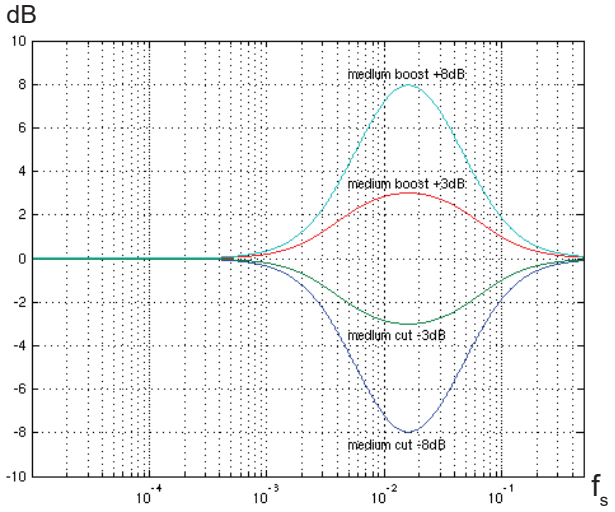
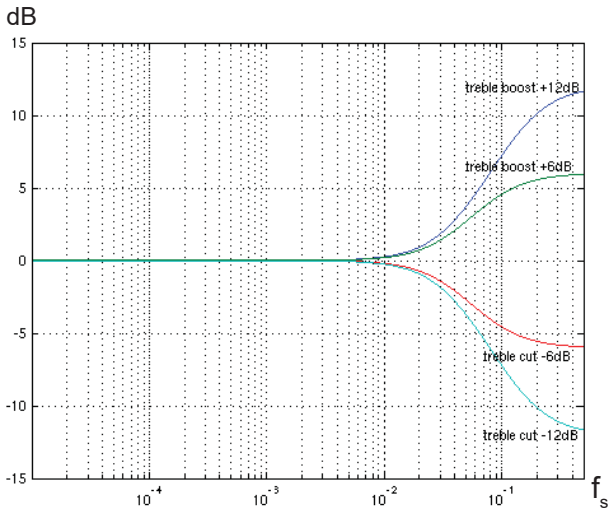


Figure 52.7. Treble Filters Response



52.6.3. De-emphasis Filter Frequency Response

The CLASSD includes a de-emphasis filter which can be enabled for 32, 44.1 or 48 kHz sampling frequencies.

The response and the error generated by the digital approximation of the filter are illustrated in the following figures.

Figure 52.8. De-emphasis Filter: Frequency Response & Error (f_s = 32 kHz)

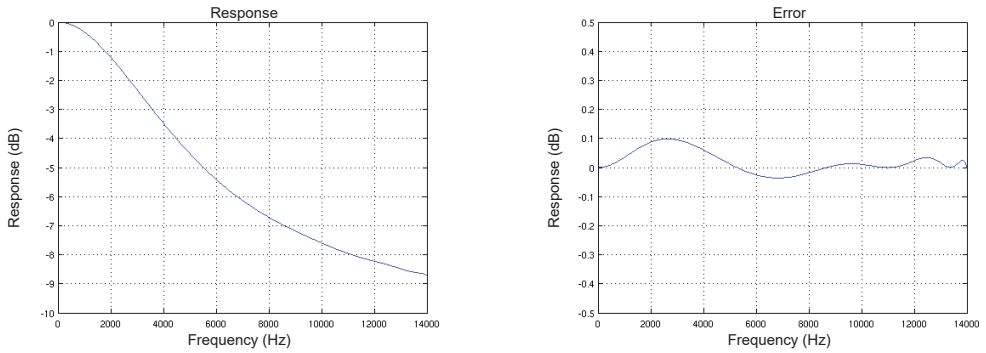


Figure 52.9. De-emphasis Filter: Frequency Response & Error (f_s = 44.1 kHz)

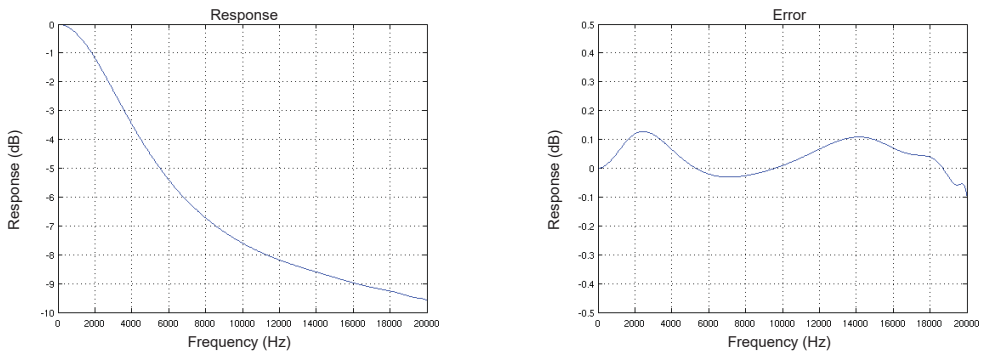
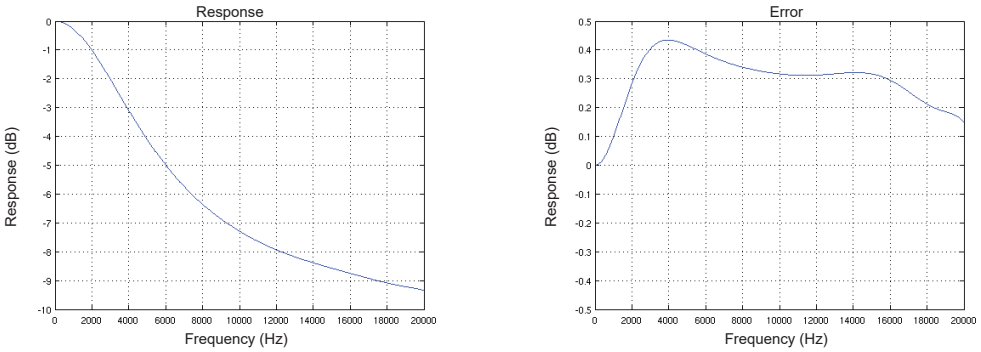


Figure 52.10. De-emphasis Filter: Frequency Response & Error (f_s = 48 kHz)



52.6.4. Attenuator and Recommended Input Levels

The CLASSD features a digital attenuator with an attenuation range of 0–77 dB and a step size of 1 dB. When attenuation greater than 77 dB is programmed, the attenuator mutes the channel.

To avoid saturations in the PWM stage, it is recommended to avoid input levels greater than 1 dB below the digital full scale (-1 dBFS). This can done by programming a minimum attenuation of 1 dB.

52.6.5. Pulse Width Modulator (PWM)

The CLASSD Pulse Width Modulator generates fixed frequency pulse width modulated output signals. For the 44.1 kS/s and 48 kS/s standard audio sample rates, the PWM output frequency is set to 16 × f_s: 705.6 kHz and 768 kHz respectively. For 8, 16, 24 and 96 kS/s, the 16× (interpolation) ratio is adapted to keep the output frequency at 768 kHz. In the same way, the output frequency is 705.6 kHz for the 22.05 and 88.2 kS/s cases.

The CLASSD functions either as a DAC loaded by a medium-to-high resistive load (e.g., 1 kΩ to 100 kΩ) or as a Class D power amplifier controller driving an external power stage. Depending on the value of CLASSD_MR.NON_OVERLAP, the CLASSD drives:

- Single-ended or differential resistive loads (NON_OVERLAP = 0)
- Full or Half MOSFET H-bridges (NON_OVERLAP = 1)

When driving an external power stage (NON_OVERLAP = 1), the CLASSD generates the signals to control complementary MOSFET pairs (PMOS and NMOS) with a non-overlapping delay between the NMOS and PMOS controls to avoid short circuit current. The non-overlapping delay can be adjusted in the CLASSD_MR.NOVRVAL field.

The CLASSD can have a single-ended or a differential output. A specific pulse width modulation type is associated to each case. For single-ended output (CLASSD_MR.PWMTYP = 0), the PWM acts only on the falling edge of the PWM waveform (trailing edge PWM). For differential output (CLASSD_MR.PWMTYP = 1), both the rising and the falling edges of the PWM waveform are modulated (symmetric PWM). Modulation principles are illustrated in the following figures for both types of PWM. In particular, when describing a null input, if PWMTYP = 0 (trailing edge PWM), the output waveform is a square wave with 50% duty cycle. With the same input and PWMTYP = 1, the differential output waveform is zero. This difference removes the classical L-C low-pass filter when PWMTYP = 1.

Figure 52.11. Output Waveform Modulation Principle for PWMTYP = 0

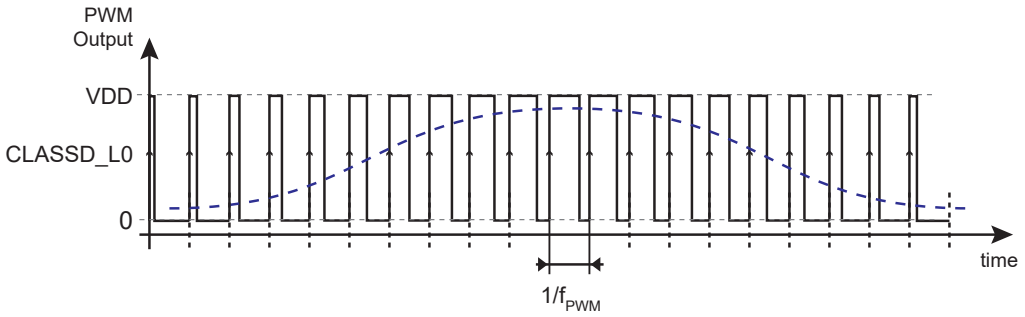
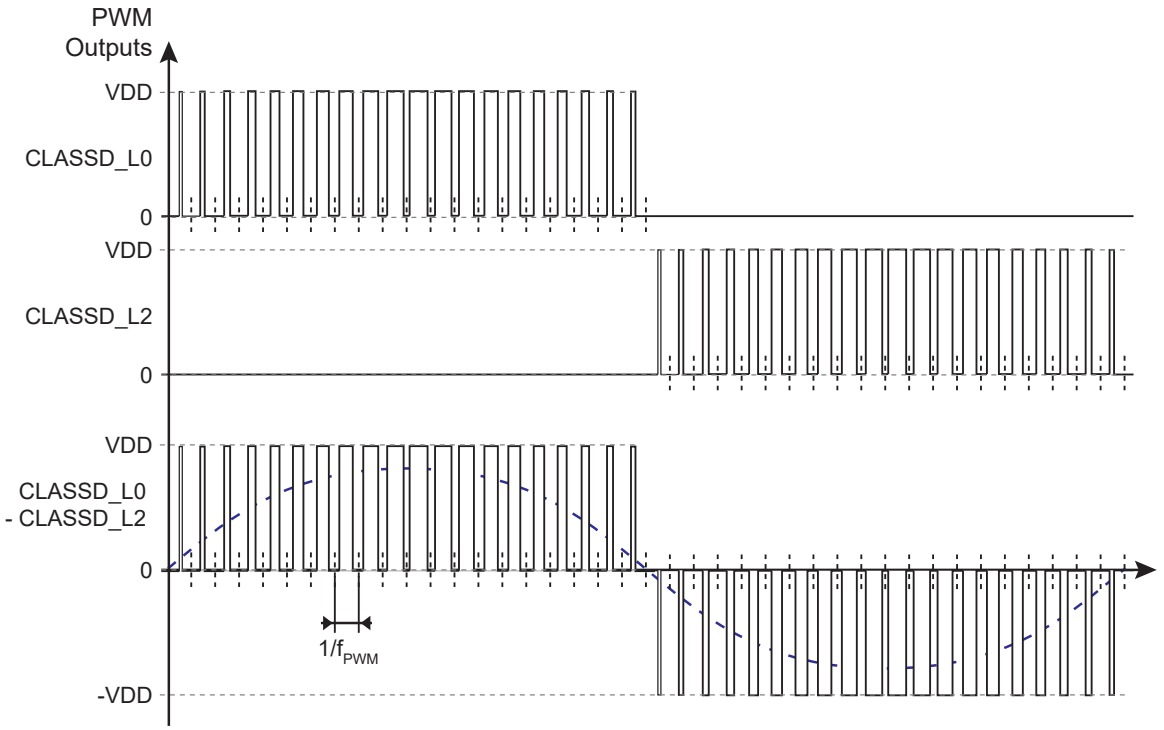
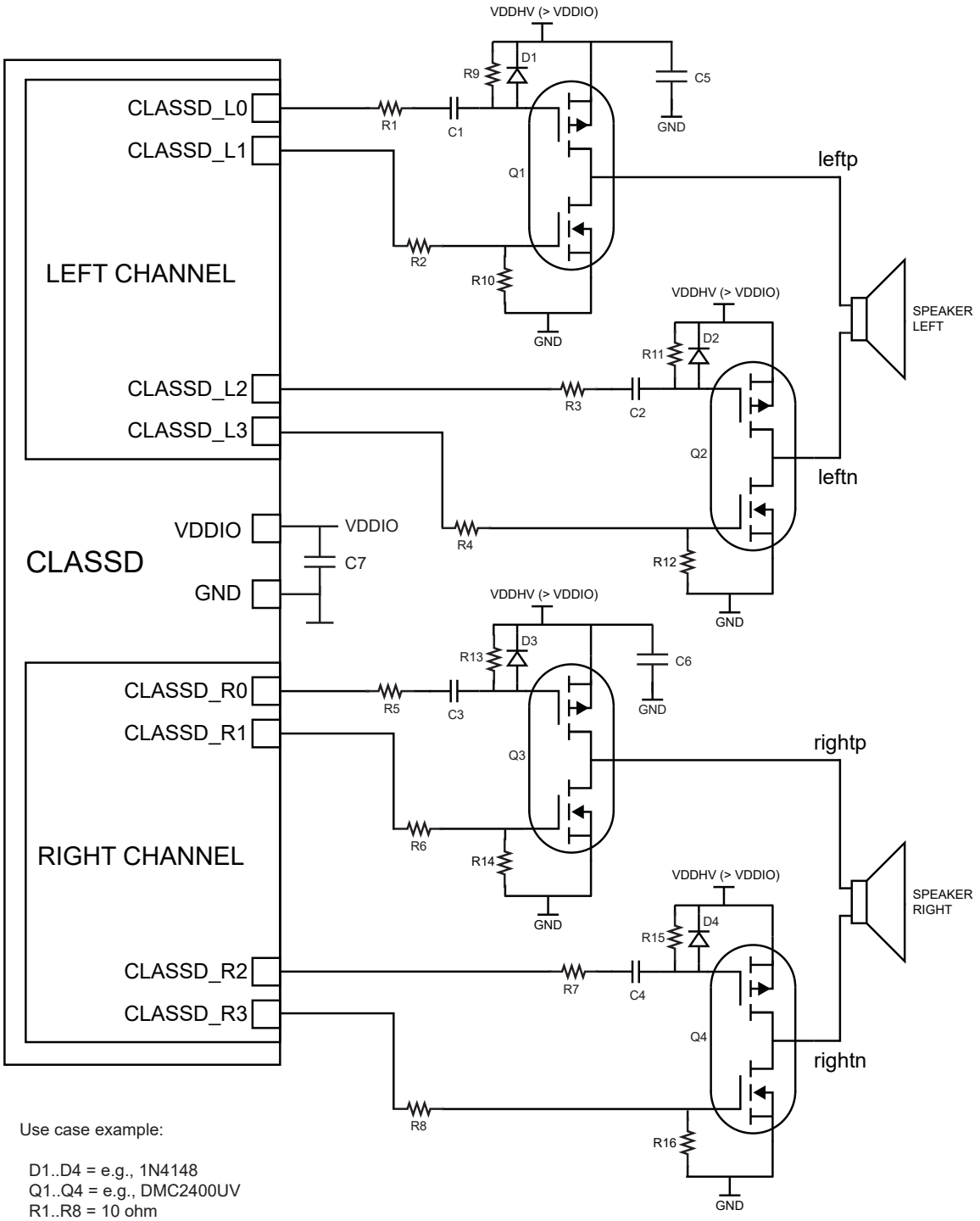


Figure 52.12. Output Waveform Modulation Principle for PWMTYP = 1 (Only Left Channel Pins Shown)



52.6.6. Application Schematics For Use Case Examples

Figure 52.13. Use Case 1: Stereo Class D Amplifier With External Differential Power Stage

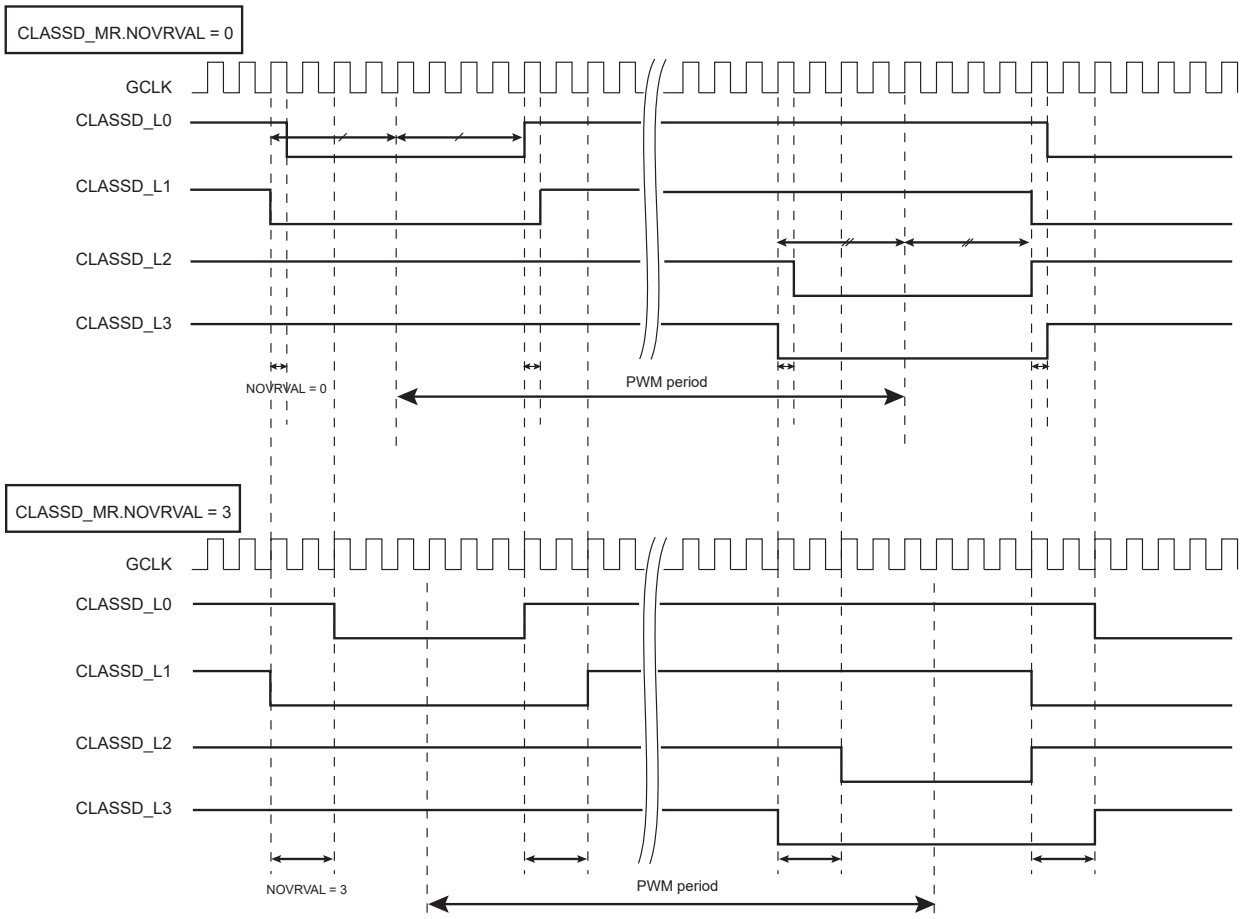


Use case example:

D1..D4 = e.g., 1N4148
Q1..Q4 = e.g., DMC2400UV
R1..R8 = 10 ohm
R9..R16 = 10 kohm
C1..C4 = 10 nF
C5..C6 = 10 µF
C7 = 1 µF

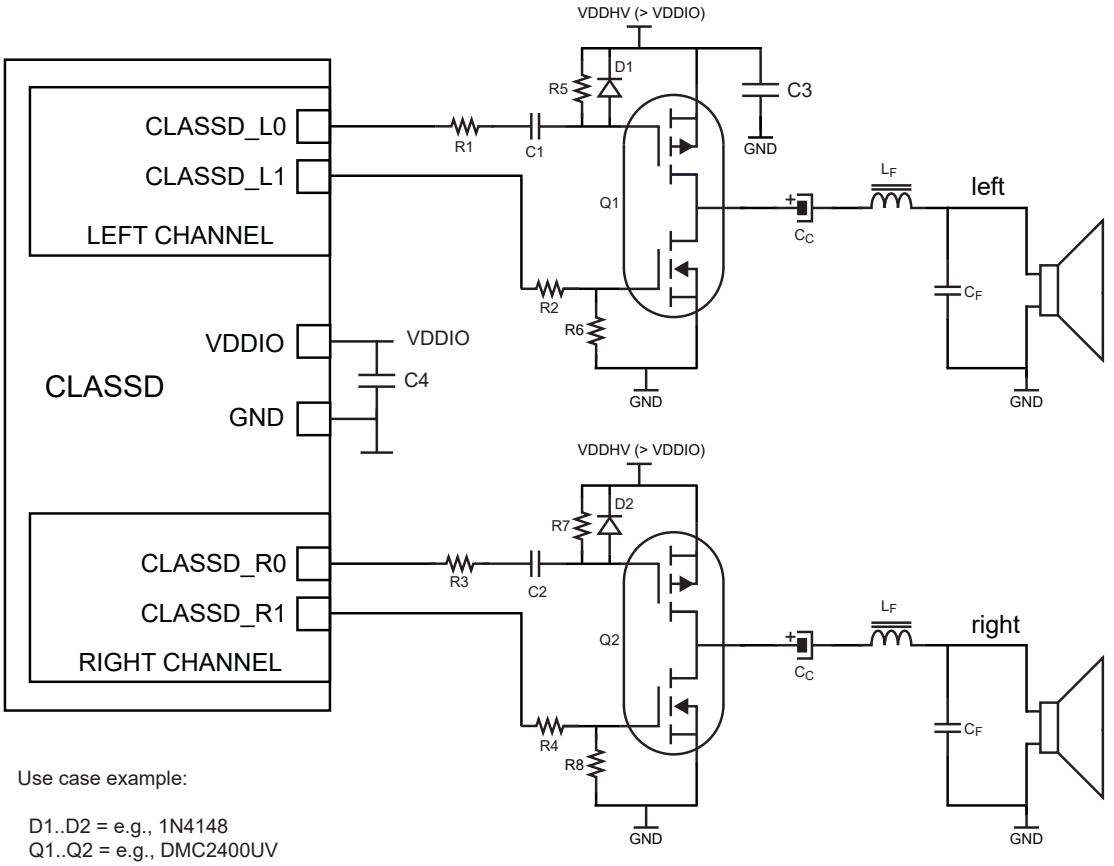
Figure 52.14. Use Case 1: Waveforms

CLASSD_MR.PWMTYP = 1, CLASSD_MR.NON_OVERLAP = 1



In Use Case 1, the external power stages are made of complementary low-cost MOSFETs. In addition to the R_{DSON} and drain breakdown voltage characteristics, the choice of these components is driven by a low gate threshold voltage, a low input capacitance, a low total gate charge and a fast turn-on time characteristics. Series resistance (10 Ω) added to the gates of the MOSFETs are optional and may be adjusted to optimize the gate drive. They help to limit the output current peaks driven by the I/Os into the MOSFET gates in some cases. The 10 kΩ resistors ensure an off condition when not driven and the capacitor/diode network (C1..C2/D1..D2) shifts the PMOS drive from the typical V_{DDIO} level (3.3V) to a higher supply voltage (e.g., a 5V power domain).

Figure 52.15. Use Case 2: Stereo Class D Amplifier With External Single-ended Power Stage



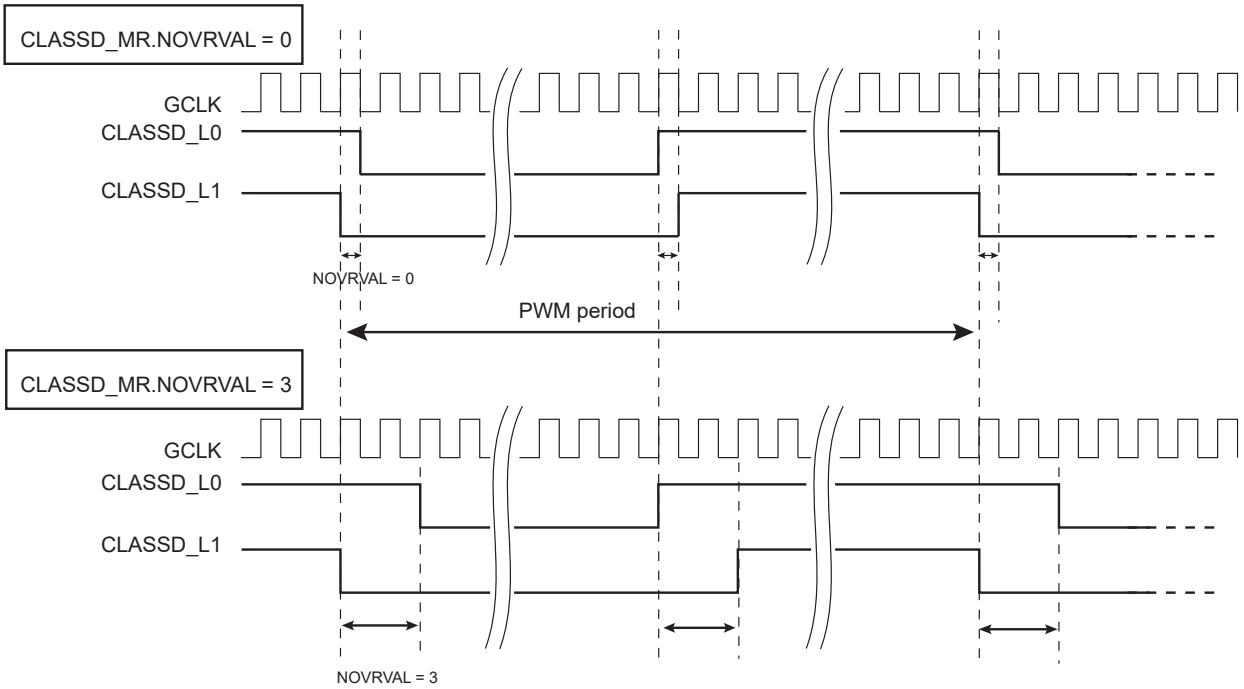
Use case example:

D1..D2 = e.g., 1N4148
Q1..Q2 = e.g., DMC2400UV
R1..R4 = 10 ohm
R5..R8 = 10 kohm
C1..C2 = 10 nF
C3 = 10 µF
C4 = 1 µF

In the Use Case 2 application schematic, the drive network of the MOSFETs gates follows the principles described in Use Case 1.

Figure 52.16. Use Case 2: Waveforms

CLASSD_MR.PWMTYP = 0, CLASSD_MR.NON_OVERLAP = 1



A coupling capacitor (C_C) and an L-C low-pass filter (L_F, C_F) are added to the output of the power stage to remove both the DC and the high frequency components of the PWM signal. C_C with the resistive part of the speaker (R_{SPK}) forms a C-R high pass filter with a corner frequency of $f_{HP} = 1 / (2 \times PI \times C_C \times R_{SPK})$.

L_F, C_F and R_{SPK} form a second-order low-pass filter of corner frequency $f_C = 1 / (2 \times PI \times sqrt (L_F \times C_F))$ and of quality factor $Q = R_{SPK} \times sqrt (C_F / L_F)$. As a numerical example, consider the case f_{HP} = 200 Hz, f_C = 30 kHz, Q = 0.707 (maximum flat response) with R_{SPK} = 8 Ω. This leads to C_C = 100 µF, L_F = 60 µH, C_F = 470 nF.

Figure 52.17. Use Case 3A: Stereo Audio DAC With Active Differential-to-Single Low-Pass Filter

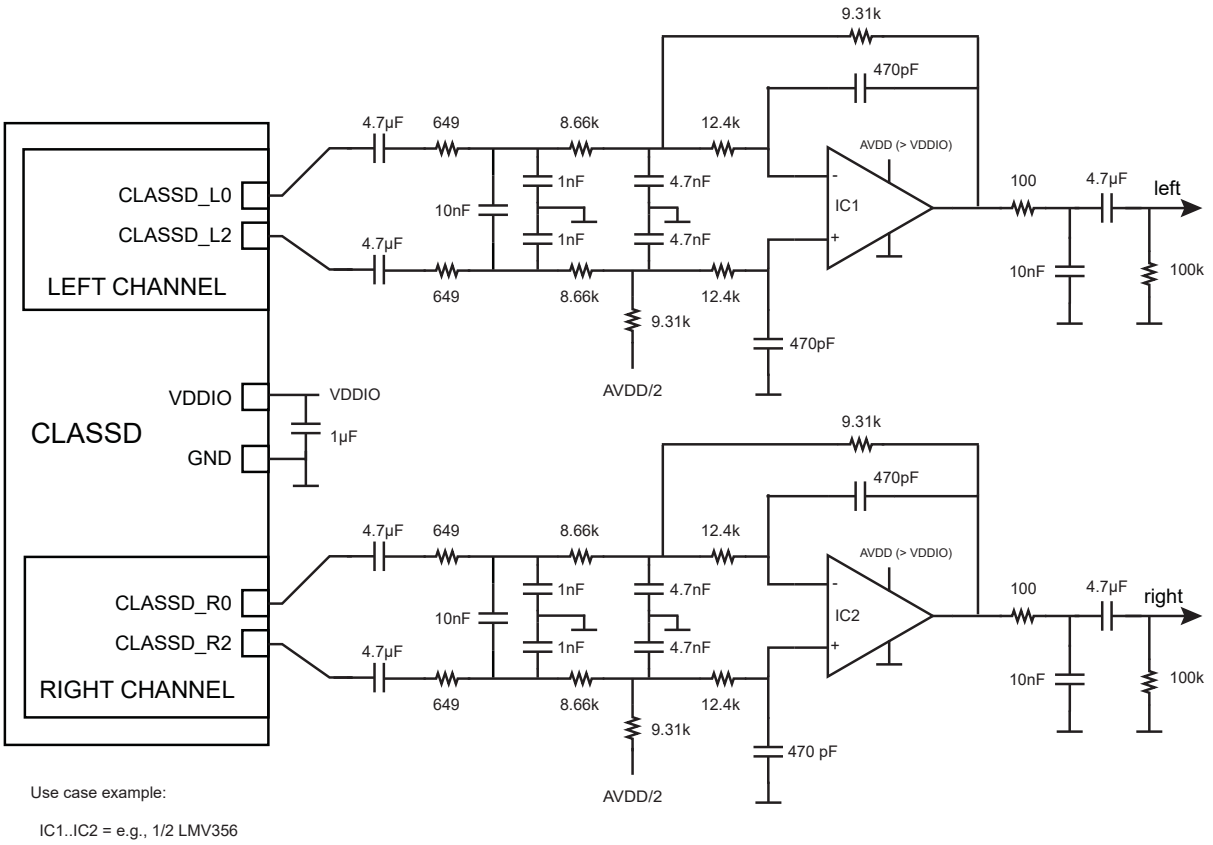


Figure 52.18. Use Case 3B: Stereo Audio DAC With Simple Passive Low-Pass Filter and Differential Outputs

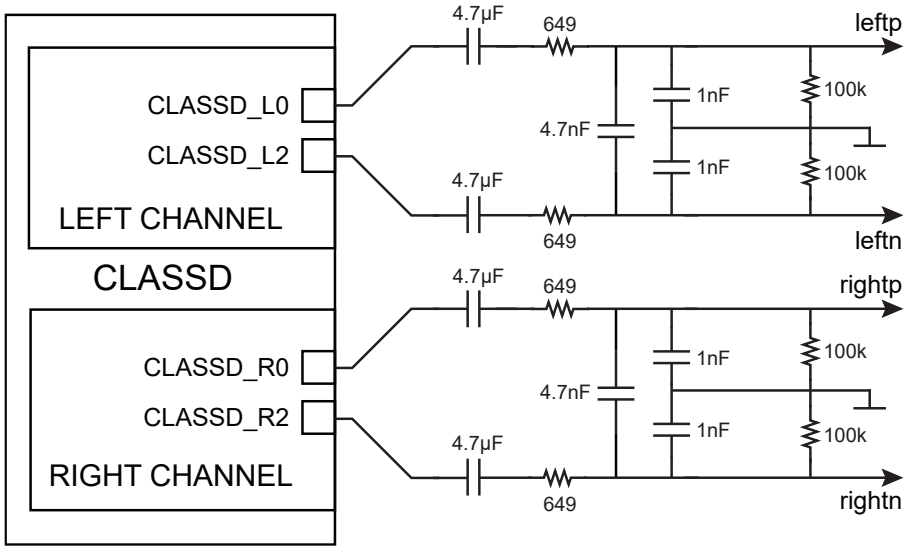
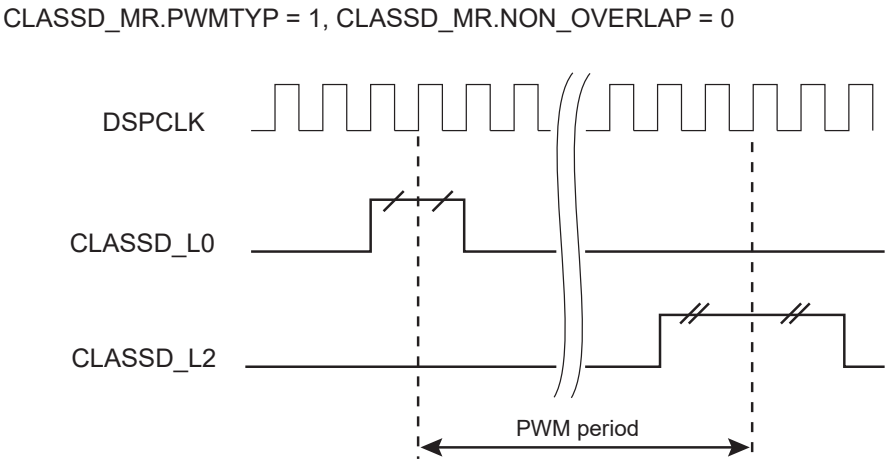


Figure 52.19. Use Cases 3A and 3B: Waveforms



In Use Case 3A, the CLASSD is used as an audio DAC. In this case, the differential outputs of the CLASSD are used. The application schematic suggested in figure "Use Case 3A: Stereo Audio DAC With Active differential to Single Low-Pass Filter" above implements a third-order 10 kHz low-pass Butterworth filter and makes the differential to single-ended conversion. Note that in this schematic, the AVDD/2 point needs to be fed at low impedance (e.g., a buffered voltage). A simpler schematic (Use Case 3B) may also be possible, as shown in figure "Use Case 3B: Stereo Audio DAC With Simple Passive Low-Pass Filter and Differential Outputs" above, at the cost of higher out-of-band noise and differential outputs which may be acceptable in some applications.

Figure 52.20. Use Case 4A: Stereo Audio DAC With Active Low-Pass Filter and Single-ended Outputs

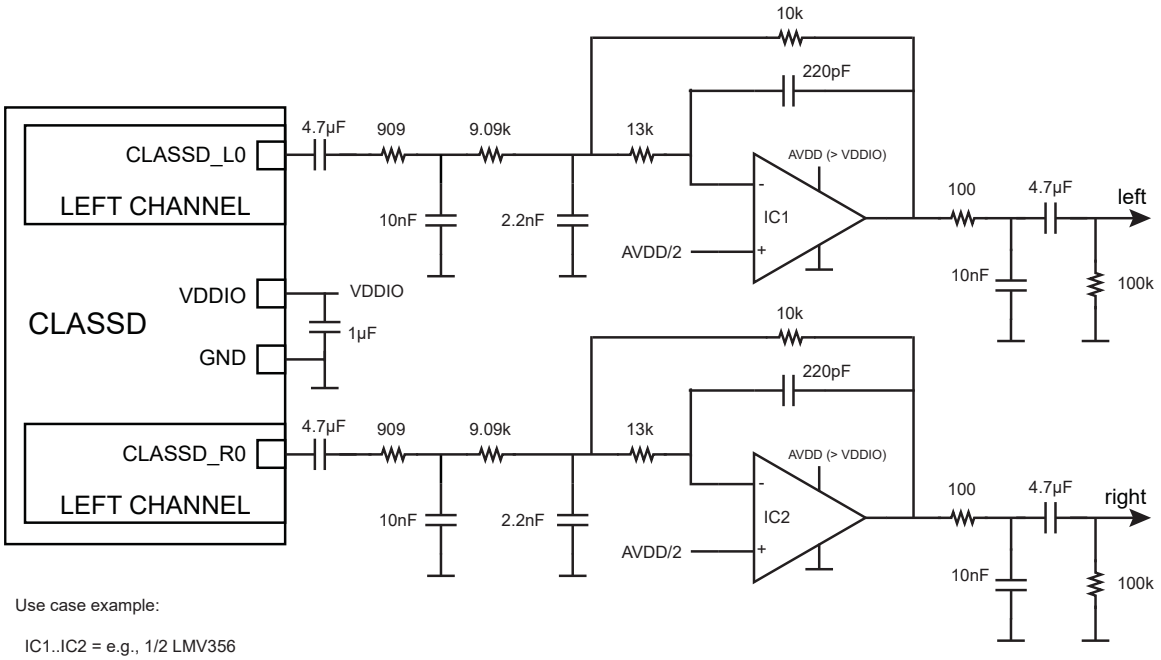


Figure 52.21. Use Case 4B: Stereo Audio DAC With Passive Low-Pass Filter and Single-ended Outputs

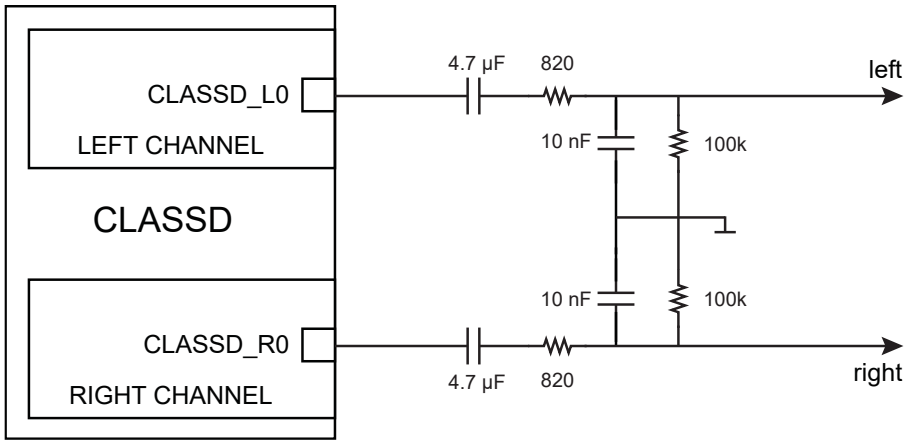
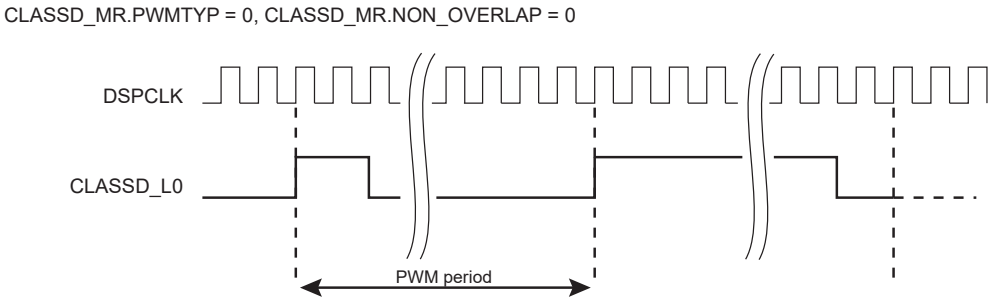


Figure 52.22. Use Cases 4A and 4B: Waveforms



In Use Case 4A, the CLASSD is used as an audio DAC with active low-pass filter. In this case, the single-ended outputs of the CLASSD are selected (PWMTYP = 0, trailing edge PWM) which leaves more I/Os to the application. A third-order 30 kHz low-pass Butterworth filter is shown in figure "Use Case 4A: Stereo Audio DAC With Active Low-Pass Filter and Single-ended Outputs". The AVDD/2 point can be fed at relatively high impedance as no current is drawn from this point (a simple resistive divider properly decoupled is acceptable). A reduced complexity schematic is presented in figure "Use Case 4B: Stereo Audio DAC With Passive Low-Pass Filter and Single-ended Outputs" above for less constrained applications.

52.6.7. Register Write Protection

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

The following registers can be write-protected:

- CLASSD Mode Register
- CLASSD Interpolator Mode Register

52.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CLASSD_CR	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x04	CLASSD_MR	31:24								
		23:16			NOVRVAL[1:0]					NON_OVERLAP
		15:8								PWMTYP
		7:0			RMUTE	REN			LMUTE	LEN
0x08	CLASSD_INTPMR	31:24		MONOMODE[1:0]		MONO	EQCFG[3:0]			
		23:16		FRAME[2:0]			SWAP	DEEMP		DSPCLKFREQ
		15:8		ATTR[6:0]						
		7:0		ATTL[6:0]						
0x0C	CLASSD_INTSR	31:24								
		23:16								
		15:8								
		7:0								CFGERR
0x10	CLASSD_THR	31:24	RDATA[15:8]							
		23:16	RDATA[7:0]							
		15:8	LDATA[15:8]							
		7:0	LDATA[7:0]							
0x14	CLASSD_IER	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x18	CLASSD_IDR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x1C	CLASSD_IMR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x20	CLASSD_ISR	31:24								
		23:16								
		15:8								
		7:0								DATRDY
0x24 ... 0xE3	Reserved									
0xE4	CLASSD_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0								WPEN

52.7.1. CLASSD Control Register

Name: CLASSD_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 CLASSD, simulating a hardware reset.

52.7.2. CLASSD Mode Register

Name: CLASSD_MR
Offset: 0x04
Reset: 0x00010022
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			NOVRVAL[1:0]					NON_OVERLAP
Access			R/W	R/W				R/W
Reset			0	0				1

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

Bit	7	6	5	4	3	2	1	0
			RMUTE	REN			LMUTE	LEN
Access			R/W	R/W			R/W	R/W
Reset			1	0			1	0

Bits 21:20 – NOVRVAL[1:0] Non-Overlapping Value
This field has no effect when NON_OVERLAP = 0.

Value	Name	Description
0	5NS	Non-overlapping time is 5 ns
1	10NS	Non-overlapping time is 10 ns
2	15NS	Non-overlapping time is 15 ns
3	20NS	Non-overlapping time is 20 ns

Bit 16 – NON_OVERLAP Non-Overlapping Enable

Value	Description
0	Non-overlapping circuit is disabled.
1	Non-overlapping circuit is enabled.

Bit 8 – PWMTYP PWM Modulation Type
0 (TRAILING_EDGE): The signal is single-ended.
If NON_OVERLAP is cleared, the signal is sent to CLASSD_L0 and CLASSD_R0 (see figure Use Case 4A or figure Use Case 4B).
If NON_OVERLAP is set, the signal is sent to CLASSD_L0/L1 and CLASSD_R0/R1 (see figure Use Case 2).
1 (UNIFORM): The signal is differential.
If NON_OVERLAP is cleared, the signal is sent to CLASSD_L0/L2 and CLASSD_R0/R2 (see figure Use Case 3A or figure Use Case 3B).

If NON_OVERLAP is set, the signal is sent to CLASSD_L0/L1/L2/L3 and CLASSD_R0/R1/R2/R3 (see figure Use Case 1).

Bit 5 – RMUTE Right Channel Mute

Value	Description
0	Right channel is unmuted.
1	Right channel is muted.

Bit 4 – REN Right Channel Enable

Value	Description
0	Right channel is disabled.
1	Right channel is enabled.

Bit 1 – LMUTE Left Channel Mute

Value	Description
0	Left channel is unmuted.
1	Left channel is muted.

Bit 0 – LEN Left Channel Enable

Value	Description
0	Left channel is disabled.
1	Left channel is enabled.

52.7.3. CLASSD Interpolator Mode Register

Name: CLASSD_INTPMR
Offset: 0x08
Reset: 0x00304E4E
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
		MONOMODE[1:0]		MONO	EQCFG[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	23	22	21	20	19	18	17	16
		FRAME[2:0]			SWAP	DEEMP		DSPCLKFREQ
Access		R/W	R/W	R/W	R/W	R/W		R/W
Reset		0	1	1	0	0		0

Bit	15	14	13	12	11	10	9	8
		ATTR[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		1	0	0	1	1	1	0

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

Bits 30:29 – MONOMODE[1:0] Mono Mode Selection
Defines which signal is sent to both channels when the MONO bit is set.

Value	Name	Description
0	MONOMIX	(left + right) / 2 is sent to both channels
1	MONOSAT	(left + right) is sent to both channels. If the sum is too high, the result is saturated.
2	MONOLEFT	THR[15:0] is sent to both the left and the right channels
3	MONORIGHT	THR[31:16] is sent to both the left and the right channels

Bit 28 – MONO Mono Signal
0 (DISABLED): The signal is sent stereo to the left and right channels.
1 (ENABLED): The same signal is sent to both the left and the right channels. The sent signal is defined by the MONOMODE field value.

Bits 27:24 – EQCFG[3:0] Equalization Selection
EQCFG field values 13–15 = flat response

Value	Name	Description
0	FLAT	Flat response
1	BBOOST12	Bass boost +12 dB
2	BBOOST6	Bass boost +6 dB
3	BCUT12	Bass cut -12 dB
4	BCUT6	Bass cut -6 dB
5	MBOOST3	Medium boost +3 dB
6	MBOOST8	Medium boost +8 dB

Value	Name	Description
7	MCUT3	Medium cut -3 dB
8	MCUT8	Medium cut -8 dB
9	TBOOST12	Treble boost +12 dB
10	TBOOST6	Treble boost +6 dB
11	TCUT12	Treble cut -12 dB
12	TCUT6	Treble cut -6 dB

Bits 22:20 – FRAME[2:0] CLASSD Incoming Data Sampling Frequency

Value	Name	Description
0	FRAME_8K	8 kHz
1	FRAME_16K	16 kHz
2	FRAME_32K	32 kHz
3	FRAME_48K	48 kHz
4	FRAME_96K	96 kHz
5	FRAME_22K	22.05 kHz
6	FRAME_44K	44.1 kHz
7	FRAME_88K	88.2 kHz

Bit 19 – SWAP Swap Left and Right Channels
0 (LEFT_ON_LSB): Left channel is on CLASSD_THR[15:0], right channel is on CLASSD_THR[31:16].
1 (RIGHT_ON_LSB): Right channel is on CLASSD_THR[15:0], left channel is on CLASSD_THR[31:16].

Bit 18 – DEEMP Enable De-emphasis Filter
0 (DISABLED): De-emphasis filter is disabled.
1 (ENABLED): De-emphasis filter is enabled.

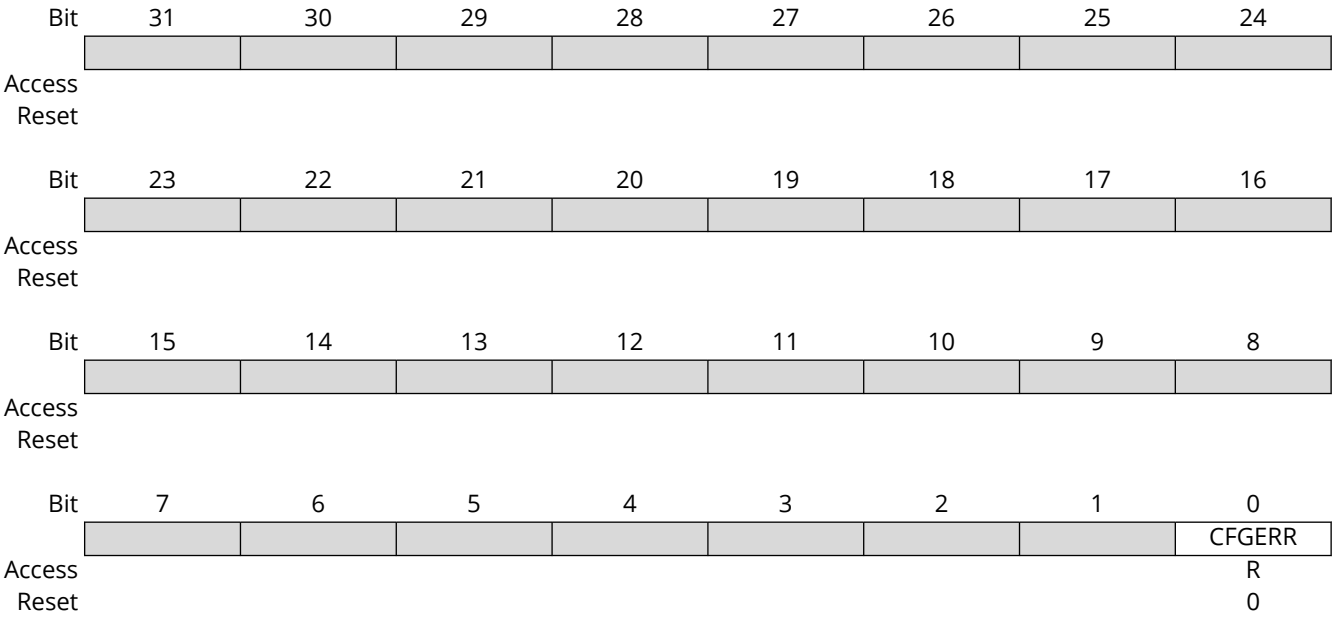
Bit 16 – DSPCLKFREQ DSP Clock Frequency
0 (12M288): DSP Clock (DSPCLK) is 12.288 MHz.
1 (11M2896): DSP Clock (DSPCLK) is 11.2896 MHz.

Bits 14:8 – ATTR[6:0] Right Channel Attenuation
Right channel attenuation is defined as follows:
– if ATTR ≤ 77 the attenuation is -ATTR dB
– else the right signal is muted

Bits 6:0 – ATTL[6:0] Left Channel Attenuation
Left channel attenuation is defined as follows:
– if ATTL ≤ 77 the attenuation is -ATTL dB
– else the left signal is muted

52.7.4. CLASSD Interpolator Status Register

Name: CLASSD_INTSR
Offset: 0x0C
Reset: 0x00000000
Property: Read-only



Bit 0 – CFGERR Configuration Error

Value	Description
0	The frame and clock configurations are correct.
1	The frame and clock configurations are incorrect (see Clock Configuration for information about allowed configurations).

52.7.5. CLASSD Transmit Holding Register

Name: CLASSD_THR
Offset: 0x10
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RDATA[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
	RDATA[7:0]							
Access	R/W	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
	LDATA[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
	LDATA[7:0]							
Access	R/W	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 – RDATA[15:0] Right Channel Data

Bits 15:0 – LDATA[15:0] Left Channel Data

52.7.6. CLASSD Interrupt Enable Register

Name: CLASSD_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
								DATRDY
Access								W
Reset								–

Bit 0 – DATRDY Data Ready

Value	Description
0	No effect.
1	Enables the interrupt when CLASSD is ready to receive new data to convert.

52.7.7. CLASSD Interrupt Disable Register

Name: CLASSD_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
Access								W
Reset								–

Bit 0 – DATRDY Data Ready

Value	Description
0	No effect.
1	Disables the interrupt when CLASSD is ready to receive new data to convert.

52.7.8. CLASSD Interrupt Mask Register

Name: CLASSD_IMR
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
								DATRDY
Access								R
Reset								0

Bit 0 – DATRDY Data Ready

Value	Description
0	The interrupt is disabled.
1	The interrupt is enabled.

52.7.9. CLASSD Interrupt Status Register

Name: CLASSD_ISR
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
								DATRDY
Access								R
Reset								0

Bit 0 – DATRDY Data Ready

Value	Description
0	CLASSD has not been ready to convert a value since the last read of CLASSD_ISR.
1	CLASSD has been ready to convert a value since the last read of CLASSD_ISR.

52.7.10. CLASSD Write Protection Mode Register

Name: CLASSD_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
0x434C44	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 write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x434C44 (“CLD” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x434C44 (“CLD” in ASCII).

53. Inter-IC Sound Multi-Channel Controller (I2SMCC)

53.1. Description

The Inter-IC Sound Controller (I2SMCC) provides a 5-wire, bidirectional, synchronous, digital audio link to external audio devices: I2SMCC_DIN, I2SMCC_DOUT, 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).

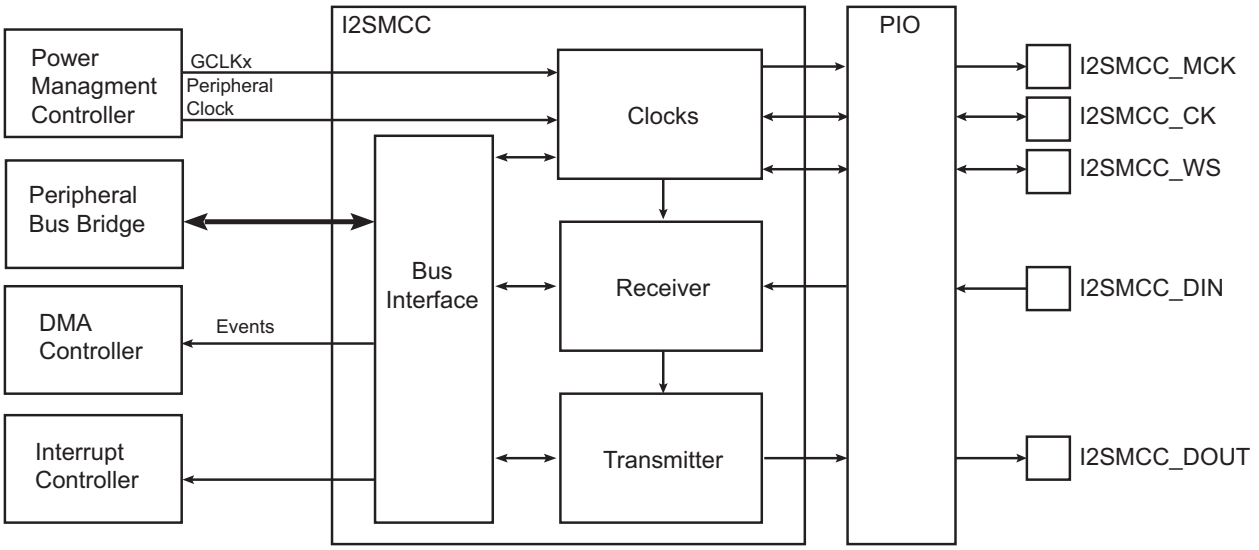
53.2. Embedded Characteristics

- 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
- 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

53.3. Block Diagram

Figure 53.1. I2SMCC Block Diagram



53.4. I/O Lines Description

Table 53.1. 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_DIN	Serial Data Input	Input
I2SMCC_DOUT	Serial Data Output	Output

53.5. Product Dependencies

To use the I2SMCC, other parts of the system must be configured correctly, as described below.

Note: In Device mode, the clock provided by the pad I2SMCC_CK must be free from glitch. It is recommended to enable hysteresis on the pad buffer driving these signals. This also applies to I2SMCC_WS.

53.5.1. 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.

53.5.2. 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.

53.5.3. 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.

53.5.4. Pad Hysteresis Control

In Client mode, the pad buffer driving I2SMCC_CK must be configured to manage the input voltage hysteresis.

53.5.5. DMA Controller

The I2SMCC interfaces to the DMA Controller. Using the I2SMCC DMA functionality requires the DMA Controller to be programmed first.

53.5.6. Interrupt Sources

The I2SMCC interrupt line is connected to the Interrupt Controller. Using the I2SMCC interrupt requires the Interrupt Controller to be programmed first.

53.6. Functional Description

53.6.1. 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.

53.6.2. Basic Operation

The receiver can be operated by reading the Receiver Holding register (I2SMCC_RHR), 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), 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.

RXLRDYx, RXRRDYx, TXLRDYx and TXRRDYx 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.

53.6.3. 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.

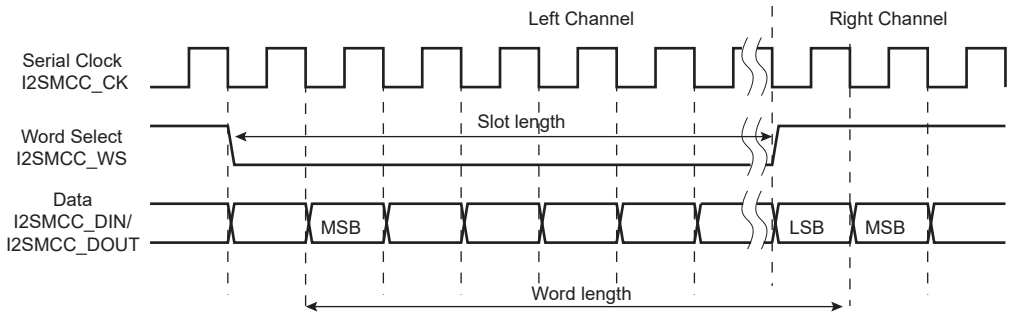
The mode is selected by writing I2SMCC_MRA.MODE. Since the MODE field changes the direction of the I2SMCC_WS and I2SMCC_CK pins, the I2SMCC_MRA must only be written when the I2SMCC is stopped in order to avoid unwanted glitches on the I2SMCC_WS and I2SMCC_CK pins.

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).

53.6.4. 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 53.2. 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 words managed in transmit and/or receive holding registers can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing I2SMCC_MRA.DATALENGTH. The length of the data transmitted or received on the I2S line (Slot length) depends on I2SMCC_MRA.DATALENGTH/IWS.

The slot length is defined in the following table.

Table 53.2. Slot Length (I²S format)

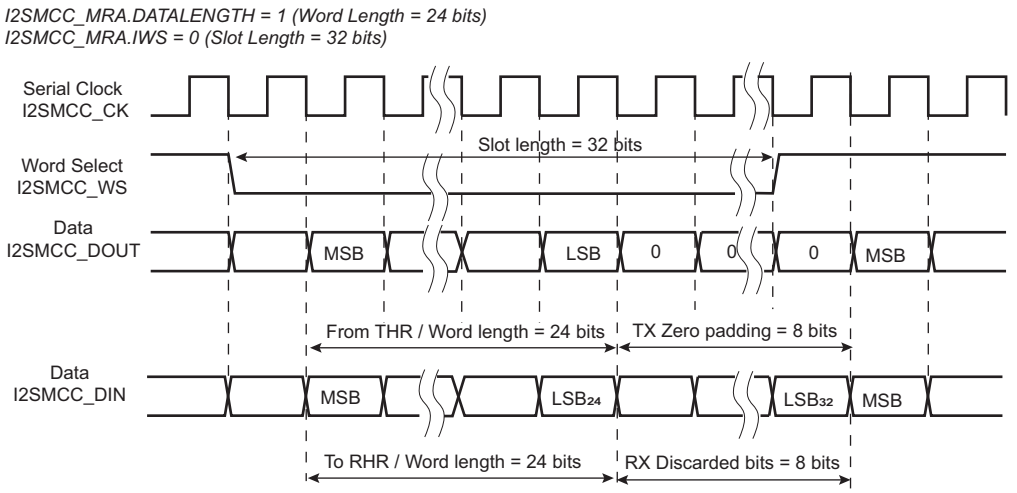
I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SMCC_MRA.IWS = 0 24 if I2SMCC_MRA.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	

Table 53.2. Slot Length (I^2S format) (continued)

I2SMCC_MRA.DATALENGTH	Word Length	Slot Length
6	8 bits	8
7	8 bits compact stereo	

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 (see examples in the following figure).

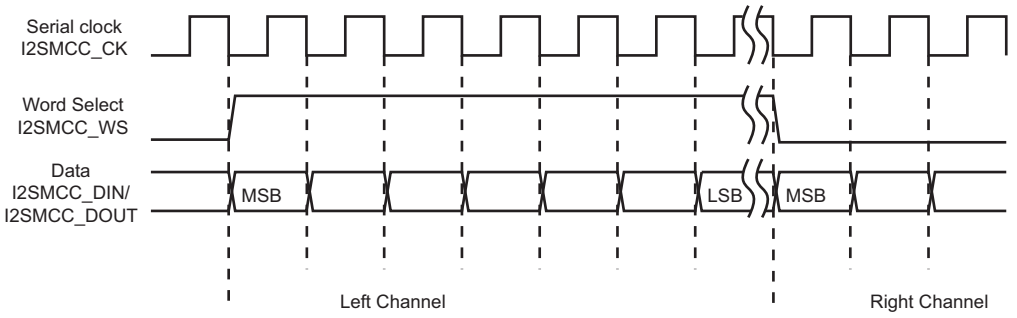
Figure 53.3. I^2S Transfer Format with TX Zero Padding and RX LSB Discarding



53.6.5. Left-Justified Reception and Transmission Sequence

As specified in the I^2S 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 53.4. 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, with a low level for the right channel and a high level for the left channel.

53.6.6. 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^2S 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 53.5. TDM Reception and Transmission Sequence

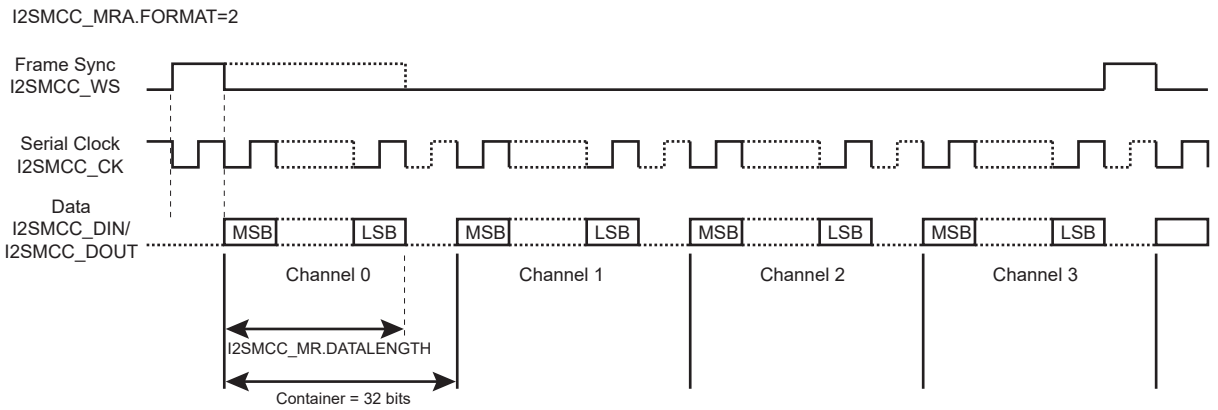
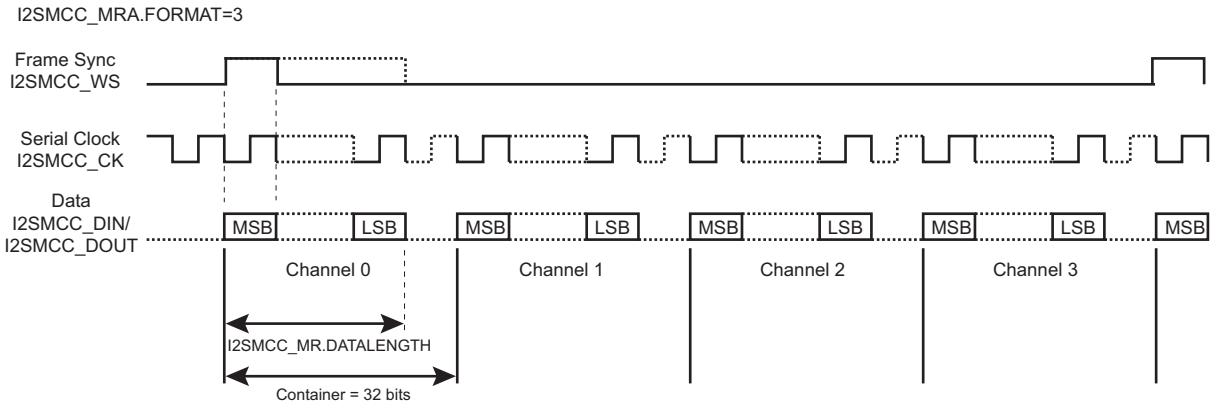


Figure 53.6. TDM Left-Justified Reception and Transmission Sequence



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 I2SMCC_MRA.TDMFS.

Note: In Client mode (I2SMCC_MRA.MODE = 0), the I2SMCC_MRA.TDMFS configuration must correspond to I2SMCC_WS high level period of the audio codec.

The number of channels is selected by writing I2SMCC_MRA.NBCHAN.

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 I2SMCC_MRA.DATALENGTH.

If the time slot allows for more data bits than programmed in I2SMCC_MRA.DATALENGTH, zeroes are appended to the transmitted data word or extra received bits are discarded.

53.6.7. Serial Clock and Word Select Generation

The generation of clocks in the I2SMCC is described in the following figure.

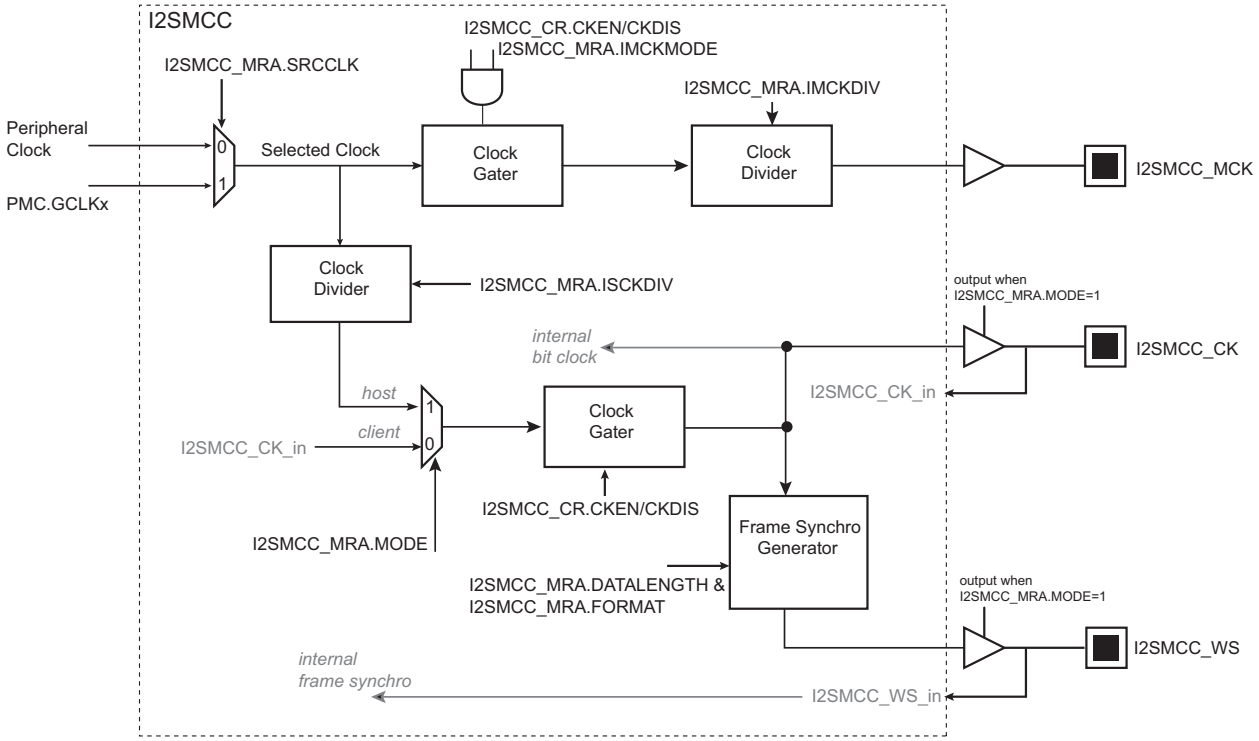
In Client mode, the word select clock is driven by an external host and the serial clock is driven by either an external host or the internal clock circuitry. I2SMCC_CK and I2SMCC_WS pins are inputs.

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 CK frequencies are configured independently using I2SMCC_MRA.IMCKDIV and I2SMCC_MRA.ISCKDIV, 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^2S format and as frame synchronization in TDM format, as described in I2S Reception and Transmission Sequence and TDM Reception and Transmission Sequence, respectively.

Figure 53.7. I2SMCC Clock Generation



53.6.8. 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, I2SMCC_ISRA.TXRRDYx are always read as '0' and the behavior of TXRUNFx 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 is unchanged.

53.6.9. Holding Registers

The I2SMCC user interface includes two common holding registers—the Receive Holding Register (I2SMCC_RHR) and the Transmit Holding Register (I2SMCC_THR). The common registers are used to access audio samples for all audio channels.

Each I²S or TDM channel has its own register. The register depth depending on the configuration is available in the following table.

Table 53.3. Registers Depth

I2SMCC_MRA.FORMAT	I2SMCC_MRA.NBCHAN	Register Depth
0 or 1 (I²S or LJ)	NA	4 words
2 or 3 (TDM or TDMLJ)	0 or 1	4 words
	2 or 3	2 words
	4, 5, 6 or 7	1 word

53.6.9.1. 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 bit RXLRDYx or RXRRDYx in I2SMCC_ISRA is set. Reading I2SMCC_RHR clears this bit. In I²S or Left-Justified mode, consecutive access to I2SMCC_RHR reads first the left channel then the right channel. In TDM or TDM Left-Justified mode, consecutive access to I2SMCC_RHR reads the TDM channels enabled.

When a receive overrun condition occurs, the corresponding bit RXLOVFx or RXROVFx in I2SMCC_ISRA is set. Reading I2SMCC_ISRA clears this bit.

When data is being received (i.e., stored in the internal shift register), it is stored in internal holding registers. As an example, when 2-wire mode is configured, up to two data for each wire can be stored because four internal holding registers are available.

If nothing is read from I2SMCC_RHR, or from I2MCC_RHLxR and I2SMCC_RHRxR, the overflow occurs if a new data becomes available for the left channel x or right channel x.

When a data can be written in I2SMCC_THR, the corresponding bit TXLRDYx bit or TXRRDYx in I2SMCC_ISRA is set. Writing to I2SMCC_THR clears this bit. In I²S or Left-Justified mode, consecutive access to I2SMCC_THR writes first the left channel then the right channel. 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 bit TXLUNFx or TXRUNFx 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.

53.6.10. 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 53.4. TX DMA Chunk Configurations

FORMAT	TXMONO	DATALENGTH	NBCHAN	I2SMCC_MRB.DMACHUNK Configuration	Maximum DMA Chunk Size Allowed
0 or 1 (I^2S or LJ)	0 (Stereo)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	No effect	0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	8-word chunk
	1 (Mono)			0	1-word chunk
				1	2-word chunk
				2	4-word chunk
				3	
	0 or 1 (Stereo or Mono)	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)	No effect	No effect	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	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 53.5. RX DMA Chunk Configurations

FORMAT	DATALENGTH	NBCHAN	I2SMCC_MRB.DMACHUNK Configuration	Maximum DMA Chunk Size Allowed
0 or 1 (I^2S or LJ)	0, 1, 2, 3, 4 or 6 (32, 24, 20, 18, 16 or 8 bits)	No effect	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)	No effect	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	

53.6.11. Loop-back Mode

For debug purposes, the I2SMCC can be configured to loop back the transmitter to the receiver. Writing a '1' to I2SMCC_MRA.RXLOOP 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.

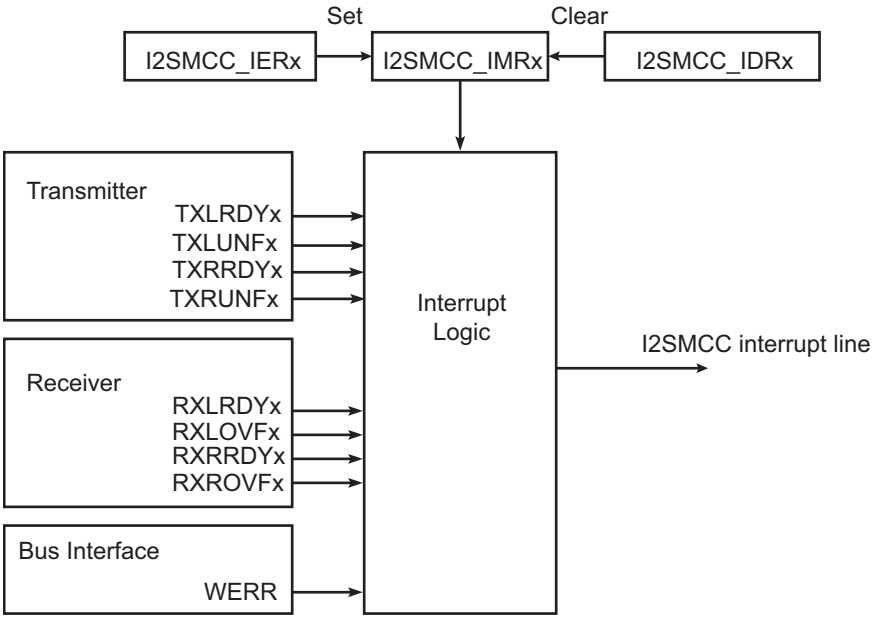
53.6.12. Interrupts

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)

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 53.8. Interrupt Block Diagram



53.6.13. 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

53.6.14. Functional Safety (Protection, Monitors and Reports)

53.6.14.1.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.

53.6.14.2.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.

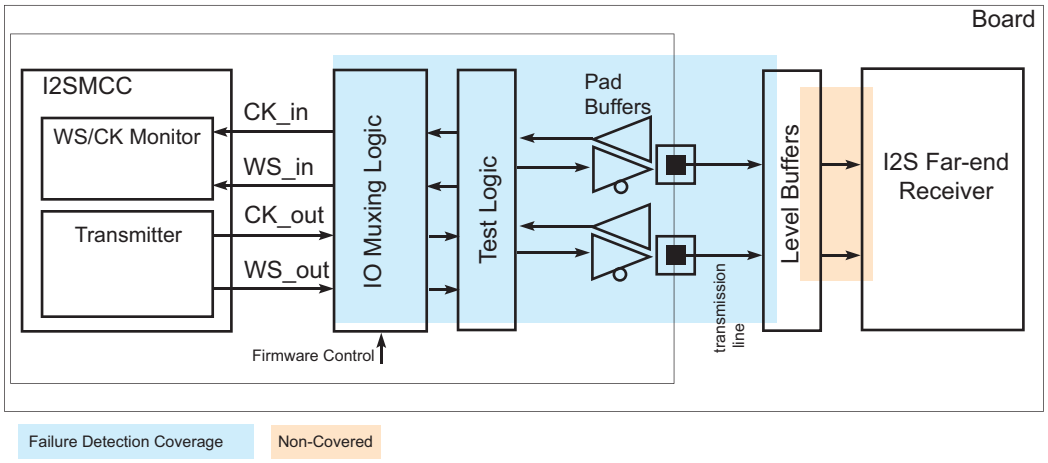
When an I2S format is configured (I2SMCC_MRA.FORMAT < 2), the I2SMCC_WS output pad value is oversampled by 2 and, as soon as 2 consecutive samples differ from the internal value generated by the I2SMCC, an error is reported. The rising edges of the I2SMCC_CK output are counted during a reference period 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 53.9. 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.

53.7. 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]		ZERO[1:0]		DATALENGTH[2:0]			MODE
0x08	I2SMCC_MRB	31:24								
		23:16								
		15:8							DMACHUNK[1:0]	
		7:0								
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								
		7:0								WERR
0x24	I2SMCC_IDRB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x28	I2SMCC_IMRB	31:24								
		23:16								
		15:8								
		7:0								WERR
0x2C	I2SMCC_ISRB	31:24								
		23:16								
		15:8								
		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]							
0x38 ... 0xE3	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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

53.7.1. I2SMCC Control Register

Name: I2SMCC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if WPCTEN 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. 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'. 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. 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’. I2SMCC_SR.RXEN is set when the receiver is activated.

53.7.2. I2SMCC Mode Register A

Name: I2SMCC_MRA
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFEN 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_DOUT 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]		ZERO[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 Slot Length (I^2S 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
In Client mode (I2SMCC_MRA.MODE = 0), the I2SMCC_MRA.TDMFS configuration must correspond to I2SMCC_WS high level period of the audio codec.

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 I2SMCC_CK 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 output 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 – ZERO[1:0] Must always be written to 0

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

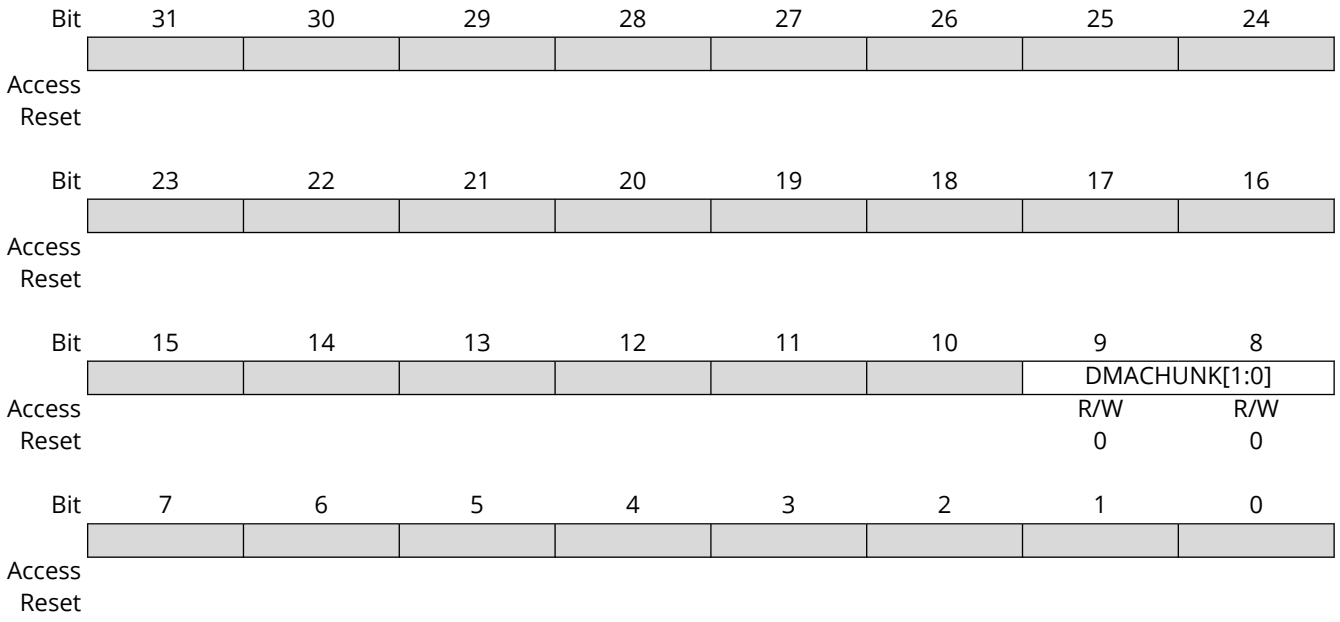
Value	Name	Description
0	SLAVE	Client mode. I2SMCC_CK and I2SMCC_WS pin inputs used as bit clock and word select/frame synchronization.
1	MASTER	Host mode. Bit clock and word select/frame synchronization generated by I2SMCC from Peripheral Clock or GCLK if I2SMCC_MCK/WS/CK rates must be independent of 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.

53.7.3. I2SMCC Mode Register B

Name: I2SMCC_MRB
Offset: 0x08
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPCFEN 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
Access								
Reset								

Bits 9:8 – DMACHUNK[1:0] DMA Chunk Size

Value	Name	Description
0	1_WORD	A DMA transfer request is issued when at least 1 word is empty in the FIFO.
1	2_WORDS	A DMA transfer request is issued when at least 2 words are empty in the FIFO.
2	4_WORDS	A DMA transfer request is issued when at least 4 words are empty in the FIFO. Limitations exist when operating in Mono or TDM. See TX DMA Chunk Configurations and RX DMA Chunk Configurations.
3	8_WORDS	A DMA transfer request is issued when at least 8 words are empty in the FIFO. Limitations exist when operating in Mono or TDM. See TX DMA Chunk Configurations and RX DMA Chunk Configurations.

53.7.4. 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.

53.7.5. I2SMCC Interrupt Enable Register A

Name: I2SMCC_IERA
Offset: 0x10
Reset: –
Property: Write-only

This register can only be written if WPITEN 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^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Enable

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Enable

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Enable

Bits 9, 11, 13, 15 – TXRUNFx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Enable

Bits 8, 10, 12, 14 – TXLUNFx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Enable

Bits 1, 3, 5, 7 – TXRRDYx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Enable

Bits 0, 2, 4, 6 – TXLRDYx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Enable

53.7.6. I2SMCC Interrupt Disable Register A

Name: I2SMCC_IDRA
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if WPITEN 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^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Disable

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Disable

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Disable

Bits 9, 11, 13, 15 – TXRUNFx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Disable

Bits 8, 10, 12, 14 – TXLUNFx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Disable

Bits 1, 3, 5, 7 – TXRRDYx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Disable

Bits 0, 2, 4, 6 – TXLRDYx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Disable

53.7.7. 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 (x=0 only) or TDM Channel [2x]+1 Overrun Interrupt Mask

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x (x=0 only) or TDM Channel 2x Overrun Interrupt Mask

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x (x=0 only) or TDM Channel 2x Ready Interrupt Mask

Bits 9, 11, 13, 15 – TXRUNFx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Underrun Interrupt Mask

Bits 8, 10, 12, 14 – TXLUNFx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Underrun Interrupt Mask

Bits 1, 3, 5, 7 – TXRRDYx I^2S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Interrupt Mask

Bits 0, 2, 4, 6 – TXLRDYx I^2S Transmit Left x (x=0 only) or TDM Channel 2x Ready Interrupt Mask

53.7.8. I2SMCC Interrupt Status Register A

Name: I2SMCC_ISRA
Offset: 0x1C
Reset: 0x00000003
Property: Read-only

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 (x=0 only) 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 I2SMCC_RHR.

Bits 24, 26, 28, 30 – RXLOVFx I^2S Receive Left x (x=0 only) 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 I2SMCC_RHR.

Bits 17, 19, 21, 23 – RXRRDYx I^2S Receive Right x (x=0 only) or TDM Channel [2x]+1 Ready Flag (Cleared by reading I2SMCC_RHR)

Value	Description
0	Cleared when a predefined number of read accesses are performed in 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 I2SMCC_RHR.

Bits 16, 18, 20, 22 – RXLRDYx I^2S Receive Left x (x=0 only) or TDM Channel 2x Ready Flag (Cleared by reading I2SMCC_RHR)

Value	Description
0	Cleared when a predefined number of read accesses is performed in I2SMCC_RHR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG/FORMAT and varies from 1 to 7.

Value	Description
1	Set when received data is available in I2SMCC_RHR.

Bits 9, 11, 13, 15 – TXRUNFx I²S Transmit Right x (x=0 only) 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 I2SMCC_THR.

Bits 8, 10, 12, 14 – TXLUNFx I²S Transmit Left x (x=0 only) 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 I2SMCC_THR.

Bits 1, 3, 5, 7 – TXRRDYx I²S Transmit Right x (x=0 only) or TDM Channel [2x]+1 Ready Flag (Cleared by writing I2SMCC_THR)

Value	Description
0	Cleared when a predefined number of write accesses is performed in I2SMCC_THR. The predefined number depends on the configuration of I2SMCC_MRA.WIRECFG/FORMAT and varies from 1 to 8.
1	Set when I2SMCC_THR is empty.

Bits 0, 2, 4, 6 – TXLRDYx I²S Transmit Left x (x=0 only) or TDM Channel 2x Ready Flag (Cleared by writing I2SMCC_THR)

Value	Description
0	Cleared when a predefined number of write accesses is performed in 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 is empty.

53.7.9. I2SMCC Interrupt Enable Register B

Name: I2SMCC_IERB
Offset: 0x20
Reset: –
Property: Write-only

This register can only be written if WPITEN is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bit 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
								WERR
Access								W
Reset								–

Bit 0 – WERR Write Error Interrupt Enable

53.7.10. I2SMCC Interrupt Disable Register B

Name: I2SMCC_IDRB
Offset: 0x24
Reset: –
Property: Write-only

This register can only be written if WPITEN is cleared in the Inter-IC Sound Write Protection Mode Register.

The following configuration values are valid for the listed bit 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
								WERR
Access								W
Reset								–

Bit 0 – WERR Write Error Interrupt Disable

53.7.11. I2SMCC Interrupt Mask Register B

Name: I2SMCC_IMRB
Offset: 0x28
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for the listed bit 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
Access								
Reset								

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

Bit 0 – WERR Write Error Interrupt Mask

53.7.12. I2SMCC Interrupt Status Register B

Name: I2SMCC_ISRB
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
								WERR
Access								R
Reset								0

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.

53.7.13. 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.

53.7.14. 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.

53.7.15. 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).

53.7.16. 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.

54. Synchronous Serial Controller (SSC)

54.1. 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:

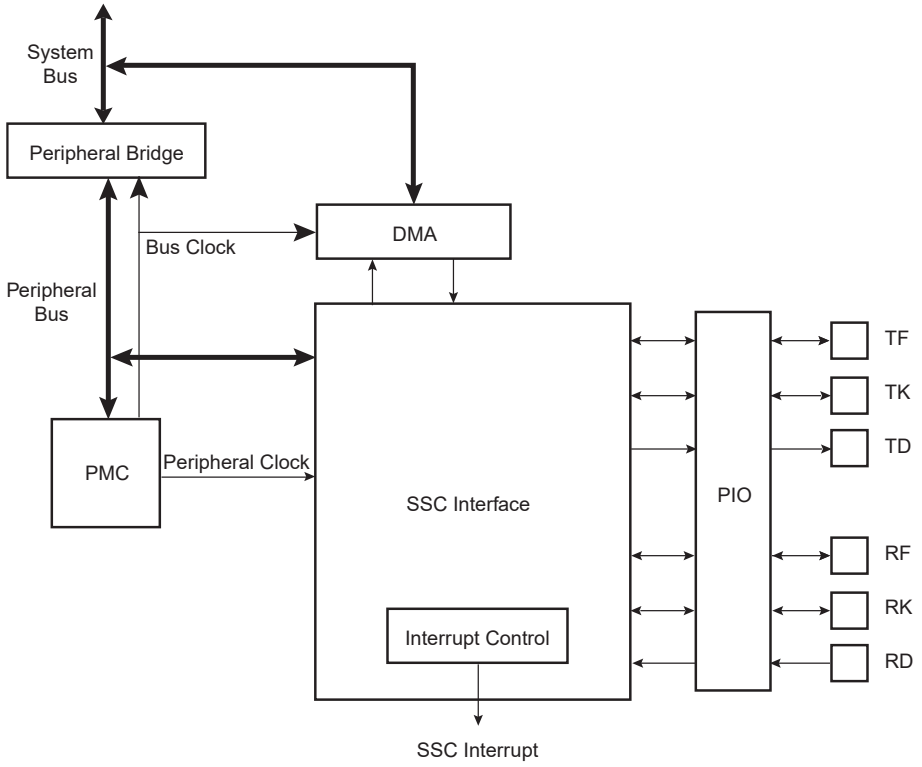
- Codecs in Host or Client mode
- DAC through dedicated serial interface, particularly I2S
- Magnetic card reader

54.2. 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
- Up to 16 Channels in TDM Mode

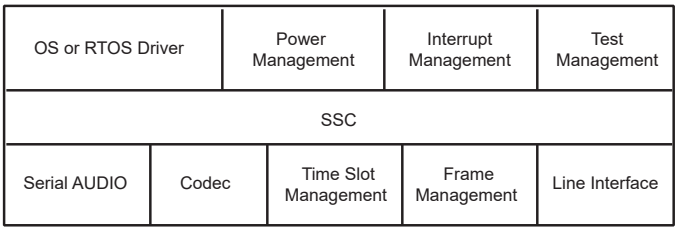
54.3. Block Diagram

Figure 54.1. SSC Block Diagram



54.4. Application Block Diagram

Figure 54.2. SSC Application Block Diagram



54.5. 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 54.3. Audio Application Block Diagram

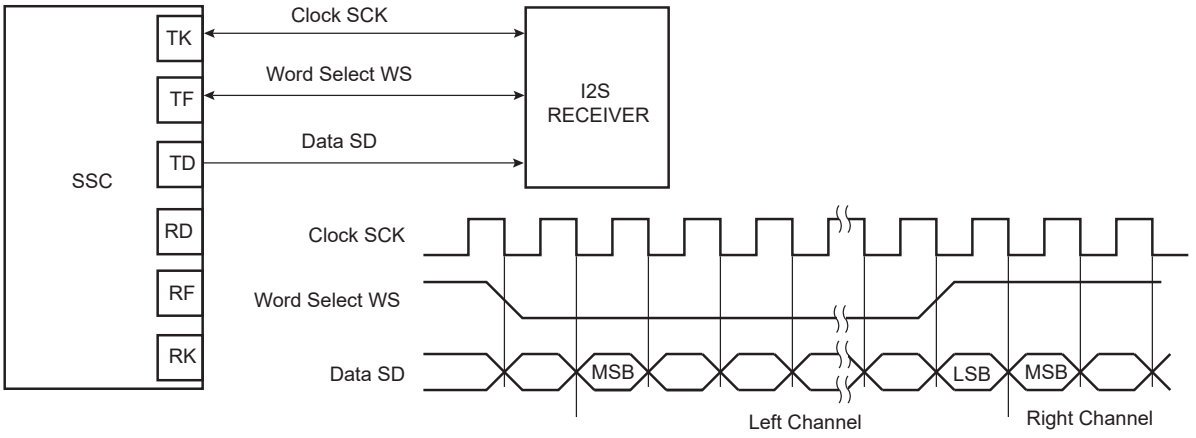


Figure 54.4. Codec Application Block Diagram

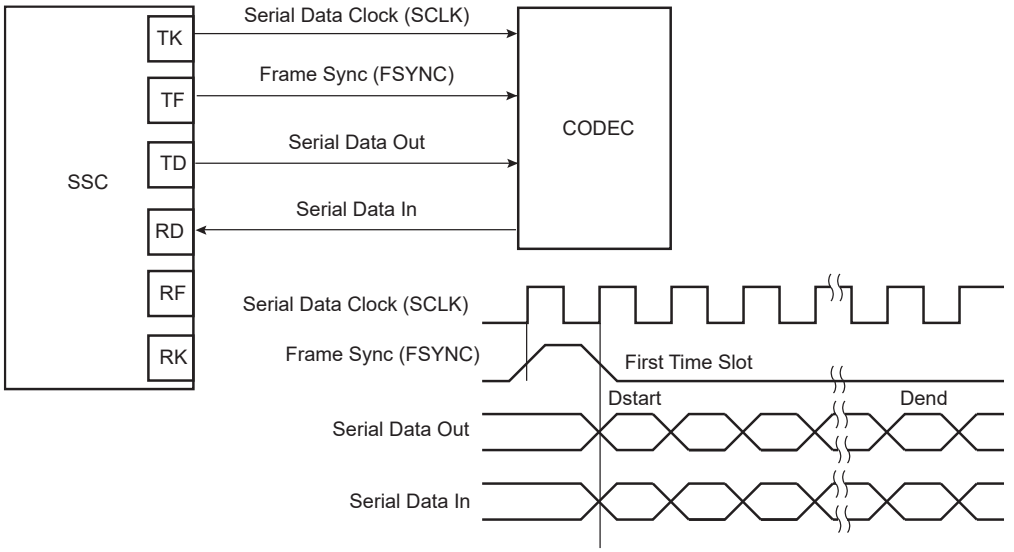
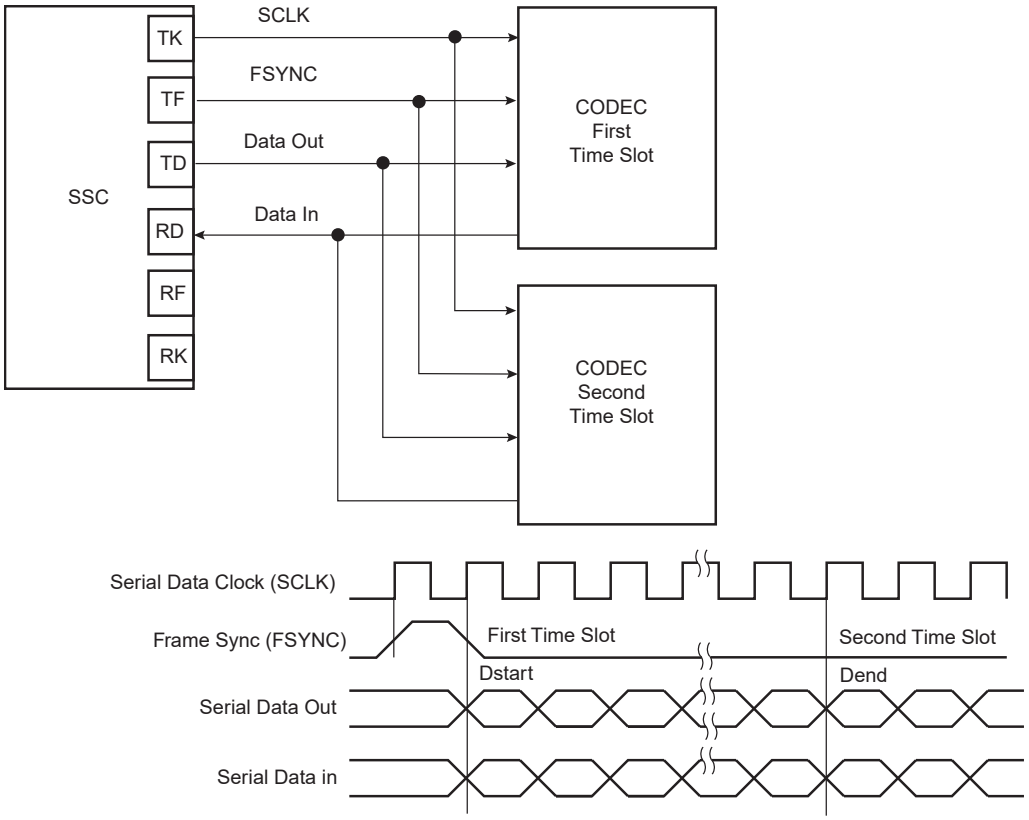


Figure 54.5. Time Slot Application Block Diagram



54.6. Pin Name List

Table 54.1. 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

54.7. Product Dependencies

54.7.1. 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.

54.7.2. 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.

54.7.3. 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.

54.7.4. Audio Sampling Rate Limitations

The maximum audio sampling rate that can be processed depends on the peripheral clock frequency and number of audio channels.

When operating in I2S mode (2 channels) the maximum rate in kilo samples per second (ksps) is the peripheral clock frequency given in kHz divided by 448 (i.e., 224 x 2).

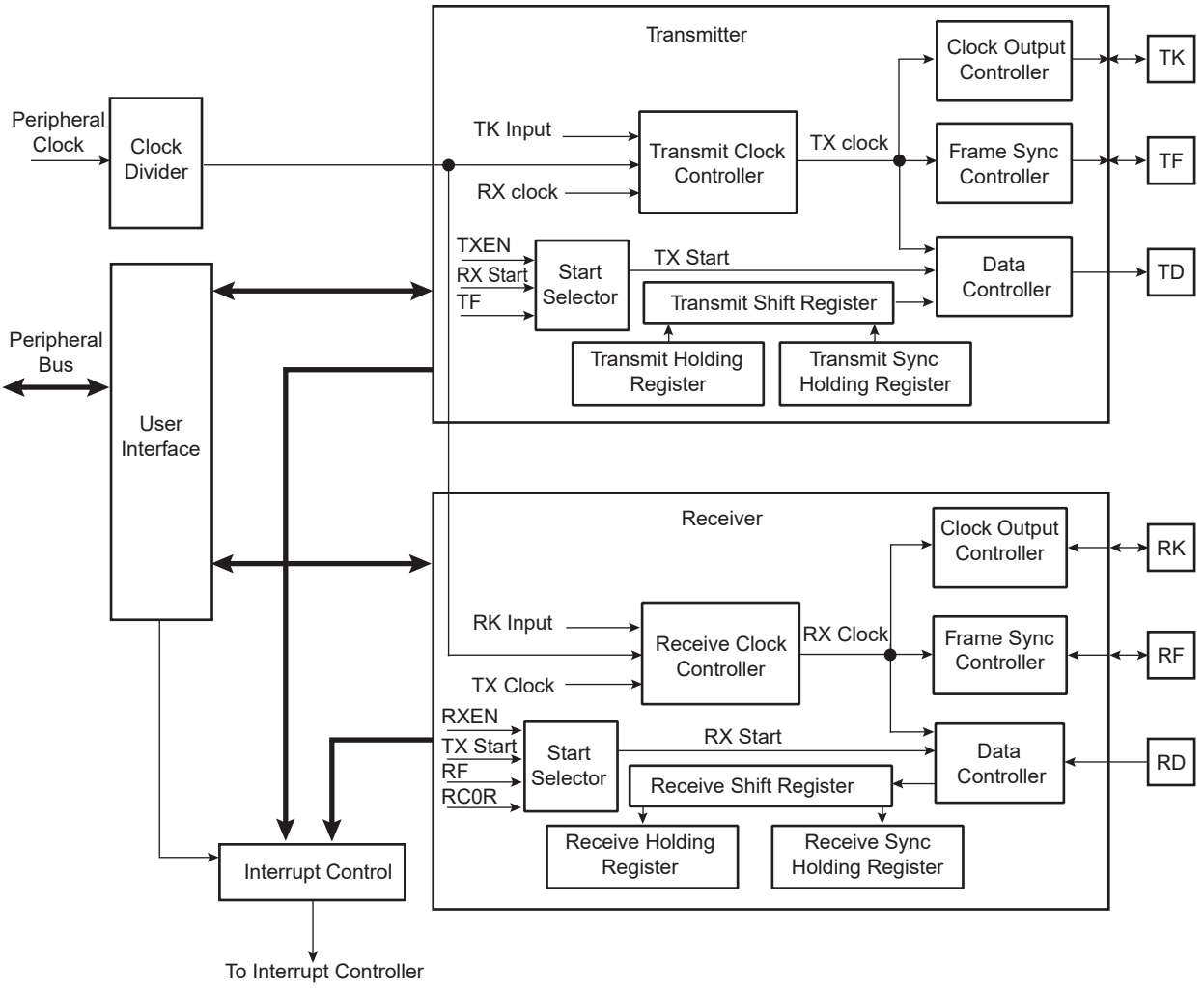
When operating in TDM mode (2 to 8 channels), the maximum rate in ksps is the peripheral clock frequency given in kHz divided by 224 x number of channels.

54.8. 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 54.6. SSC Functional Block Diagram



54.8.1. 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:

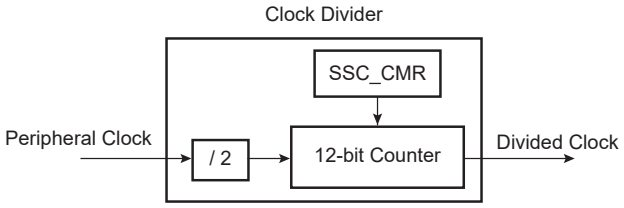
- an external clock received on the RK I/O pad
- the transmit clock
- the internal clock divider

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.

54.8.1.1.Clock Divider

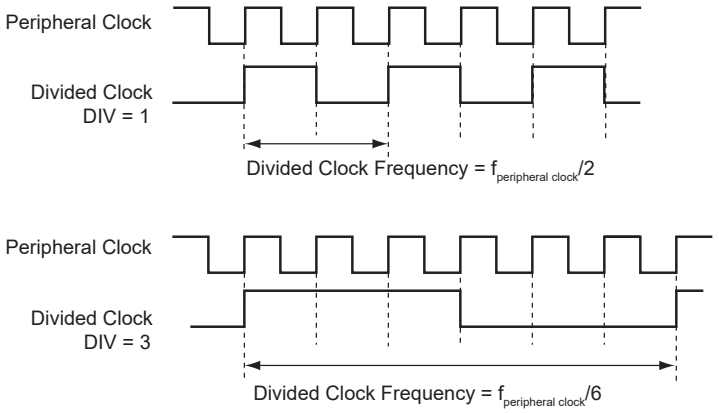
Figure 54.7. Divided Clock Block Diagram



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 54.8. Divided Clock Generation

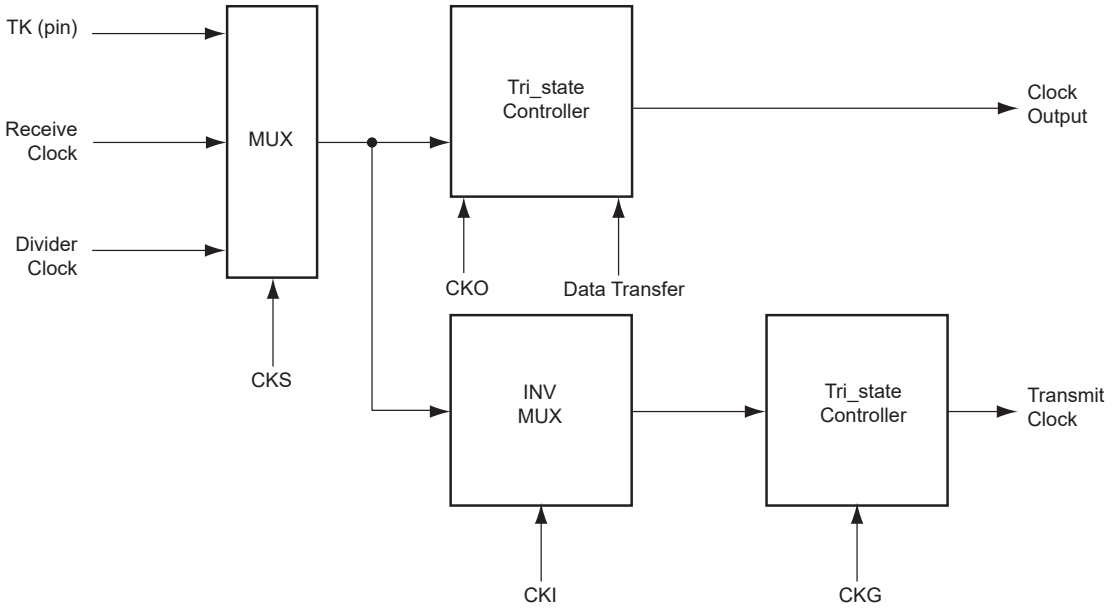


54.8.1.2.Transmit Clock Management

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 54.9. Transmit Clock Management

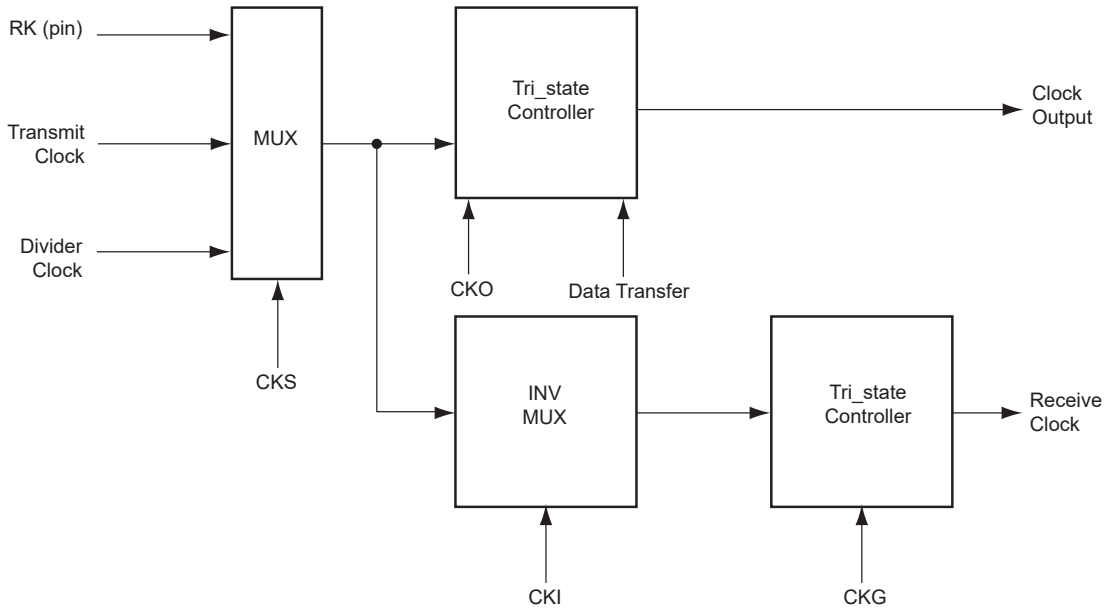


54.8.1.3. Receive Clock Management

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 54.10. Receive Clock Management



54.8.1.4.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.

54.8.2. Transmit Operations

A transmit frame is triggered by a start event and can be followed by synchronization data before data transmission.

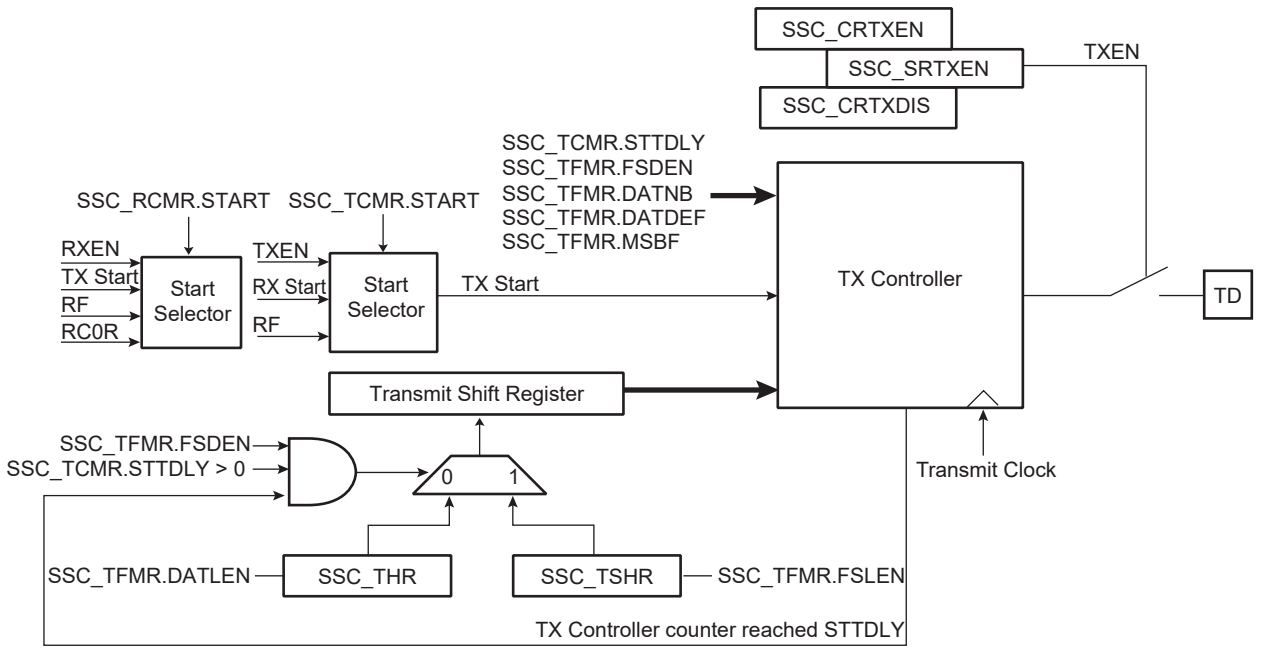
The start event is configured by setting the SSC_TCMR. See Start.

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 is written by the application to the Transmit Holding register (SSC_THR) then transferred to the transmit shift register according to the data format selected.

When both the SSC_THR and the transmit shift register are empty, the status flag TXEMPTY is set in the Status register (SSC_SR). When the Transmit Holding register is transferred in the transmit shift register, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the Transmit Holding register.

Figure 54.11. Transmit Block Diagram



54.8.3. Receive Operations

A receive frame is triggered by a start event and can be followed by synchronization data before data transmission.

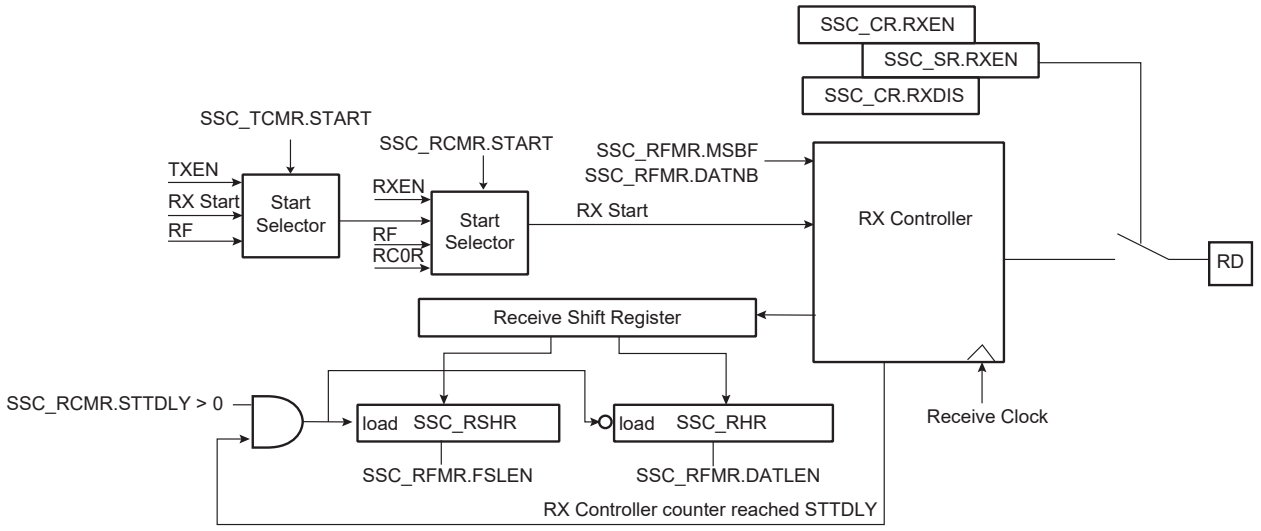
The start event is configured by setting the Receive Clock Mode Register (SSC_RCMR). See Start.

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 Holding register (SSC_RHR), the status flag RXRDY is set in the SSC_SR and the data can be read in the Receive Holding register. If another transfer occurs before read of the Receive Holding register, the status flag OVRUN is set in the SSC_SR and the receive shift register is transferred in the SSC_RHR. The old unread data is then lost.

Figure 54.12. Receive Block Diagram



54.8.4. 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 a word is written in SSC_THR 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 54.13. Transmit Start Mode

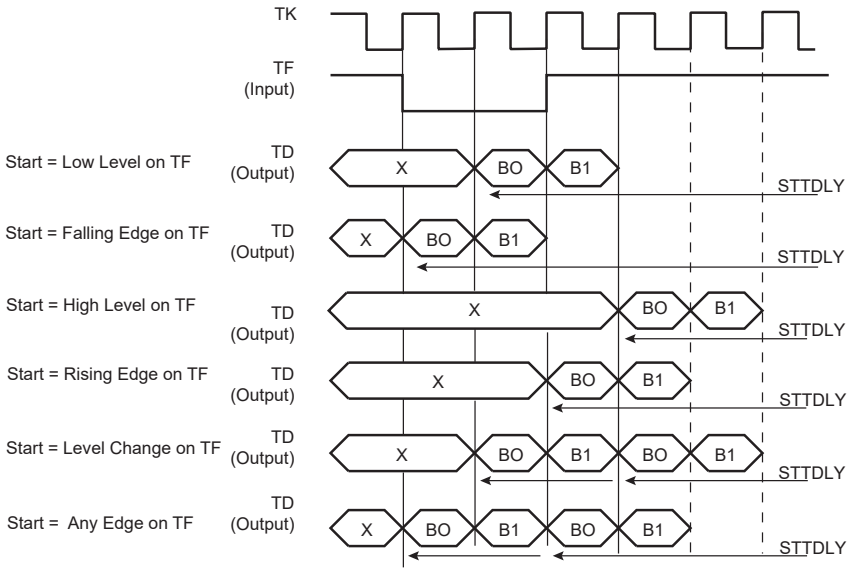
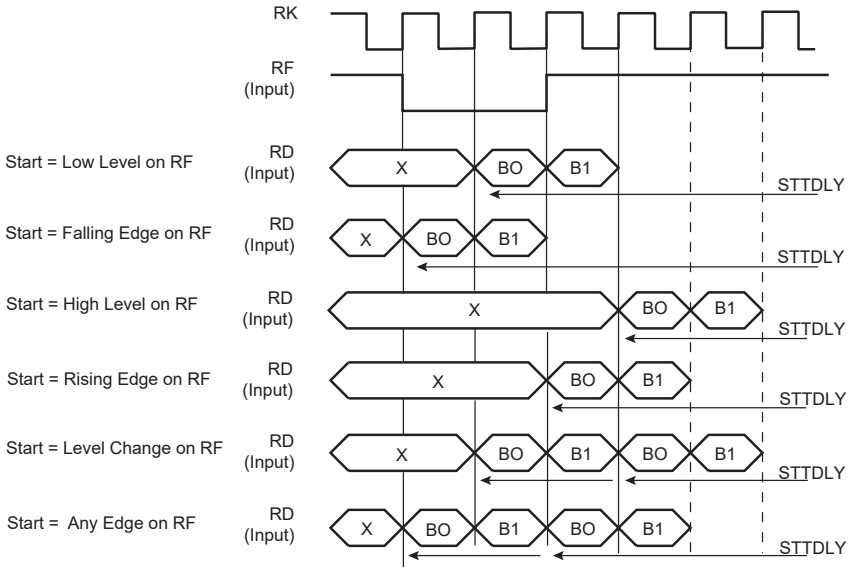


Figure 54.14. Receive Pulse/Edge Start Modes



54.8.5. 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.

54.8.5.1. 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.

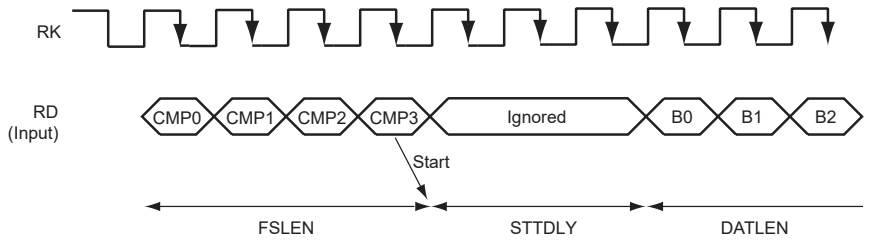
The Transmit Frame Sync Operation is performed by the transmitter only if the bit Frame Sync Data Enable (FSDEN) in SSC_TFMR is set. If the Frame Sync length is equal to or lower than the delay between the start event and the current data transmission, the normal transmission has priority and the data contained in the Transmit Sync Holding Register is transferred in the Transmit Register, then shifted out.

54.8.5.2. 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).

54.8.6. Receive Compare Modes

Figure 54.15. Receive Compare Modes



54.8.6.1. 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.

54.8.7. 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)

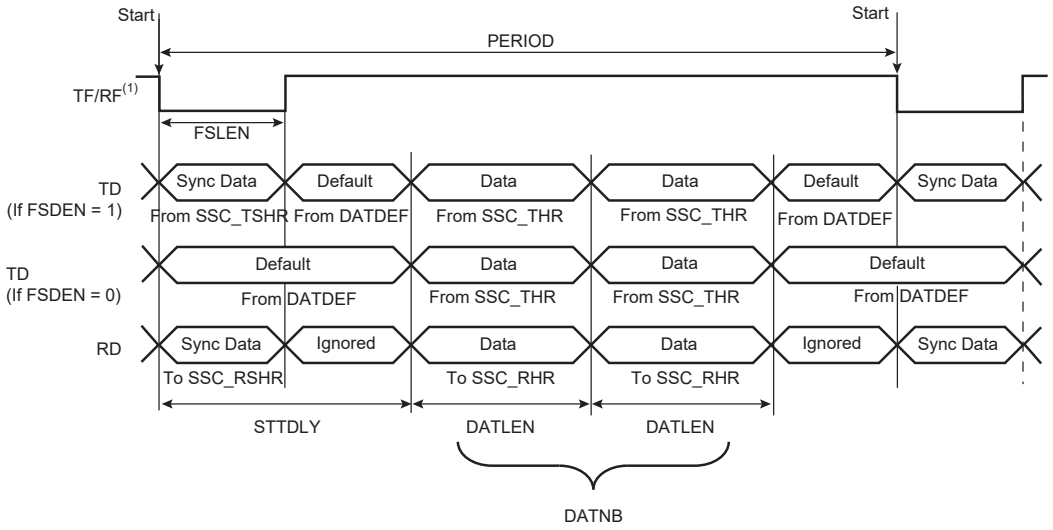
- Number of data to be transferred for each start event (DATNB)
- Length of synchronization transferred for each start event (FSLEN)
- Bit sense: most or least significant bit first (MSBF)

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 54.2. 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 54.16. 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_THR is loaded twice. The FSDEN value has no effect on the transmission. SyncData cannot be output in Continuous mode.

Figure 54.17. Transmit Frame Format in Continuous Mode (STTDLY = 0)

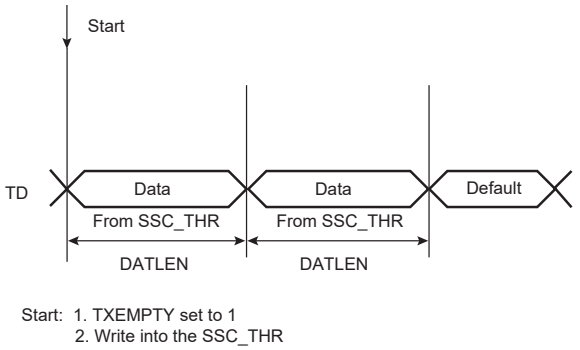
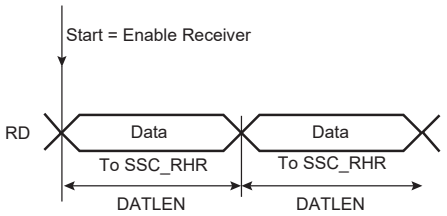


Figure 54.18. Receive Frame Format in Continuous Mode (STTDLY = 0)



54.8.8. Loop Mode

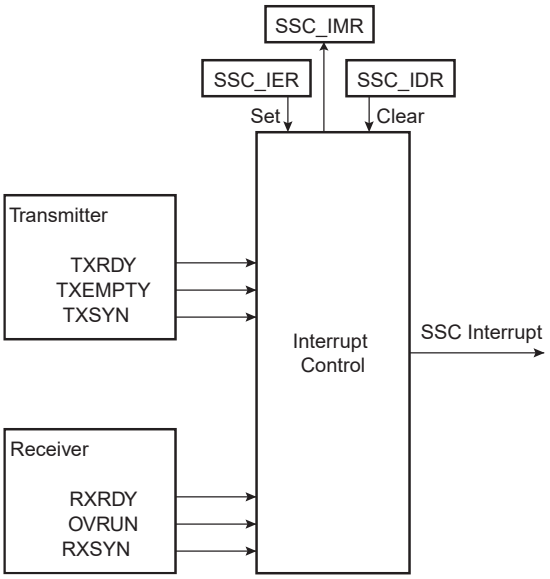
The receiver can be programmed to receive transmissions from the transmitter. This is done by setting the Loop Mode (LOOP) bit in the SSC_RFMR. In this case, RD is connected to TD, RF is connected to TF and RK is connected to TK.

54.8.9. 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 54.19. Interrupt Block Diagram



54.8.10. 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

54.9. 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_RHR	31:24	RDAT[31:24]							
		23:16	RDAT[23:16]							
		15:8	RDAT[15:8]							
		7:0	RDAT[7:0]							
0x24	SSC_THR	31:24	TDAT[31:24]							
		23:16	TDAT[23:16]							
		15:8	TDAT[15:8]							
		7:0	TDAT[7:0]							
0x28 ... 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]							
0x3C	SSC_RC1R	31:24								
		23:16								
		15:8	CP1[15:8]							
		7:0	CP1[7:0]							
0x40	SSC_SR	31:24								
		23:16							RXEN	TXEN
		15:8					RXSYN	TXSYN	CP1	CP0
		7:0			OVRUN	RXRDY			TXEMPTY	TXRDY

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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

54.9.1. 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.

Value	Description
1	Enables Receive if RXDIS is not set.

54.9.2. 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 $f_{peripheral\ clock}/2 \times 4095 = f_{peripheral\ clock}/8190$.

54.9.3. 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

Value	Name	Description
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

54.9.4. 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 bit.

54.9.5. 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: STTDLY must be set carefully. If STTDLY is too short in respect to TAG (Transmit Sync Data) transmission, data is transmitted instead of the end of TAG.

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
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

54.9.6. 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 the description of FSLEN.

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 bit.

54.9.7. SSC Receive Holding Register

Name: SSC_RHR
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.

54.9.8. SSC Transmit Holding Register

Name: SSC_THR
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.

54.9.9. 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

54.9.10. 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

54.9.11. 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

54.9.12. 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

54.9.13. SSC Status Register

Name: SSC_SR
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
							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

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	No Rx Sync has 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	No Tx Sync has 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	No compare 1 has occurred since the last read of the Status register.

Value	Description
1	A compare 1 has occurred since the last read of the Status register.

Bit 8 – CP0 Compare 0

Value	Description
0	No compare 0 has 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.

54.9.14. 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.

Value	Description
1	Enables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Enable

Value	Description
0	No effect.
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.

54.9.15. 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 Disable

Value	Description
0	No effect.
1	Disables the Rx Sync Interrupt.

Bit 10 – TXSYN Tx Sync Interrupt Disable

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.

Value	Description
1	Disables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Disable

Value	Description
0	No effect.
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.

54.9.16. 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.

Value	Description
1	The Receive Overrun Interrupt is enabled.

Bit 4 – RXRDY Receive Ready Interrupt Mask

Value	Description
0	The Receive Ready Interrupt is disabled.
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.

54.9.17. SSC Write Protection Mode Register

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

See Register Write Protection for the list of registers that can be 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
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

Value	Description
0	Disables the write protection of the configuration registers if WPKEY corresponds to 0x535343 (“SSC” in ASCII).
1	Enables the write protection of the configuration registers if WPKEY corresponds to 0x535343 (“SSC” in ASCII).

54.9.18. 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.

Security and Cryptography Subsystem

55. Overview

55.1. Components

- Advanced Encryption Standard (AES) engine
- Triple Data Encryption Standard (TDES) engine
- Secure Hash Algorithm (SHA)
- True Random Number Generator (TRNG)
- One-Time-Programming Memory Controller (OTPC)
- One Physically Unclonable Function (PUF)

Data is moved into and out of the different blocks using a Direct Memory Access (DMA) engine.

55.1.1. Cryptography Subsystem Keybus

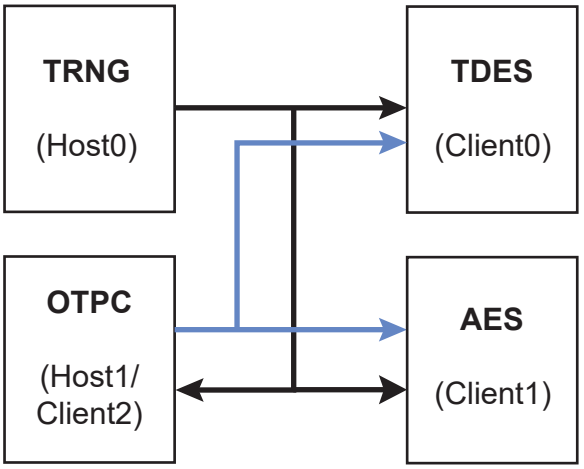
The keybus is a private bus transferring keys from hosts (TRNG, OTPC) to clients (AES, TDES, OTPC) with no possibility for the processor or software to read the keys.

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 and TDES, the PKRS bit must be set in the MR/EMR register.

At host level, the client destination and the key length must be defined before starting the transfer.

The device features a keybus system with two hosts and three clients, connected as shown in the following figure.

Figure 55.1. Cryptography Keybus



55.2. Product Dependencies

55.2.1. Clocks

All clocks are controlled by the PMC, which is part of the system controller.

55.2.2. Interrupts

Refer to the table Peripheral Identifiers.

55.2.3. Reset

Cryptography peripherals are connected to the processor and peripherals reset line.

55.2.4. I/Os

None.

55.3. Special Functions in SFR

The PUF is controlled with several registers:

- SFR_PUFCTL manages power and enable/disable of the peripheral.
- SFR_PUFDIS manages the allowed functions.
- SFR_PUFRUCR0, SFR_PUFWOCR0, SFR_PUFRUCR1 and SFR_PUFWOCR1 manages the key generation allowed for the consumers (ROM code, bootstrap, various applications).

56. Advanced Encryption Standard (AES)

56.1. 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.

56.2. 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

56.3. Product Dependencies

56.3.1. 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.

56.3.2. Interrupt Sources

The AES interface has an interrupt line connected to the Interrupt Controller.

Handling the AES interrupt requires programming the Interrupt Controller before configuring the AES.

56.4. 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.

56.4.1. 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

56.4.2. 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.

56.4.3. 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.

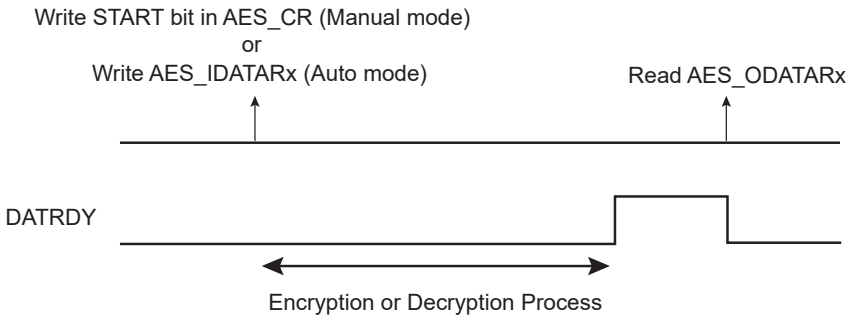
This data are only available in AES_ODATARx.

56.4.3.1.Manual and Auto Modes

56.4.3.1.1.If AES_MR.LOD = 0

The DATRDY flag is cleared when at least one AES_ODATARx is read (see the following figure).

Figure 56.1. Manual and Auto Modes with AES_MR.LOD = 0



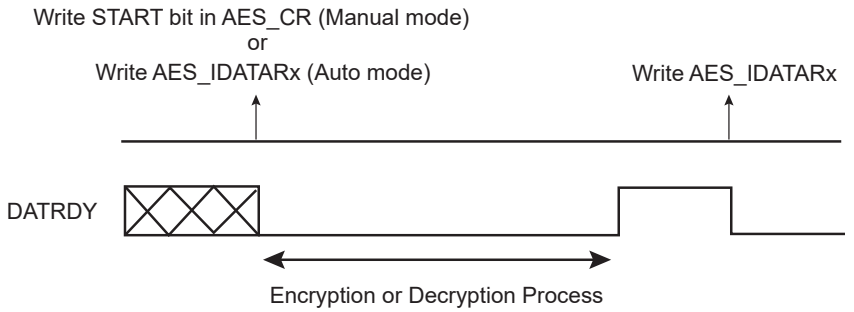
If the user does not want to read AES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user cannot know the end of the following encryptions/decryptions.

56.4.3.1.2.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 56.2. Manual and Auto Modes with AES_MR.LOD = 1



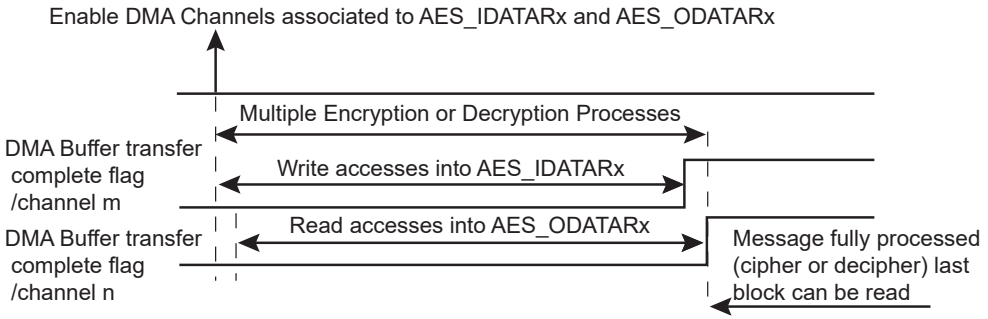
56.4.3.2.DMA Mode

56.4.3.2.1.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 56.3. DMA Transfer with AES_MR.LOD = 0



56.4.3.2.2.If AES_MR.LOD = 1

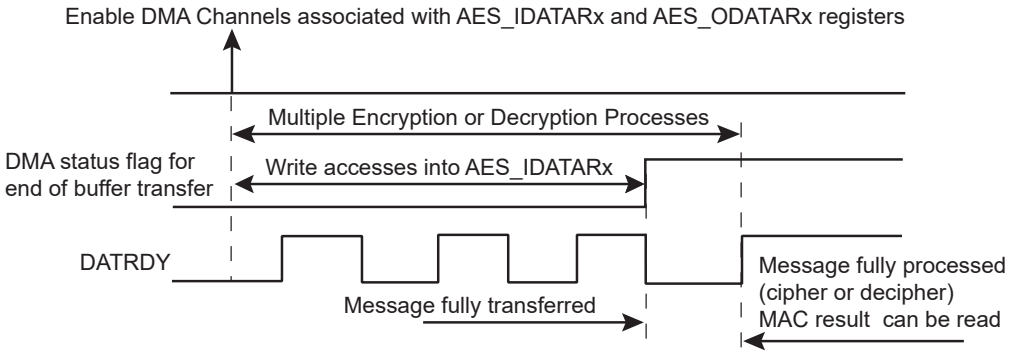
This mode is optimized to process AES CBC-MAC operating mode.

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 56.4. DMA Transfer with AES_MR.LOD = 1



The following table summarizes the different cases.

Table 56.1. 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.

56.4.4. Galois/Counter Mode (GCM)

56.4.4.1. 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.

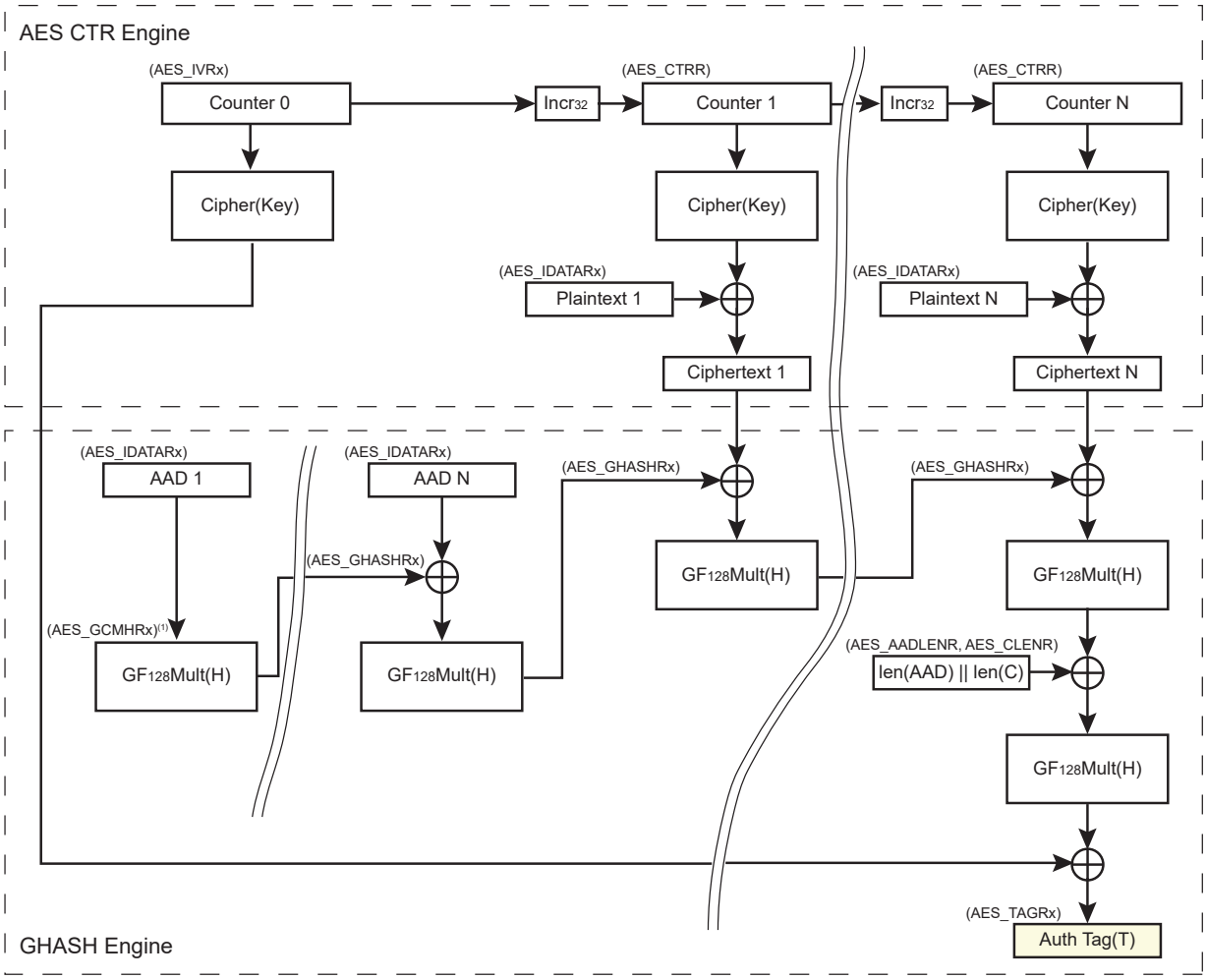
The GHASH engine processes data packets after the AES operation. GCM assures the confidentiality of data through the AES Counter mode of operation for encryption. Authenticity of the confidential data is assured through the GHASH engine. GCM can also provide assurance of data that is not encrypted. Refer 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 56.5. GCM Block Diagram



Note: 1. Optional

56.4.4.2.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

56.4.4.3.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).

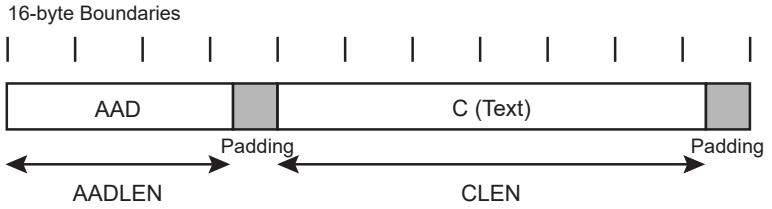
The Tag generation can be done either automatically, after the end of *AAD*/C processing if AES_MR.GTAGEN is set, or manually using AES_GHASHRx.GHASH (see subsections Processing a Complete Message with Tag Generation and Manual GCM Tag Generation for details).

56.4.4.3.1.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 56.6. 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.

56.4.4.3.2.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.

56.4.4.3.3.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.

56.4.4.3.4.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.

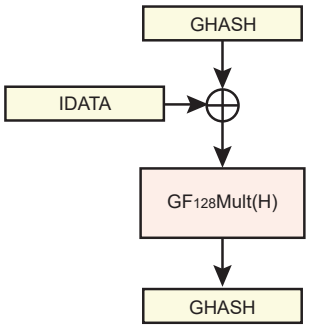
Processing *T* = GCTRK(J_0, S):

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).

56.4.4.3.5.Processing a Message with only AAD (GHASHH)

Figure 56.7. 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).

56.4.4.3.6.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.

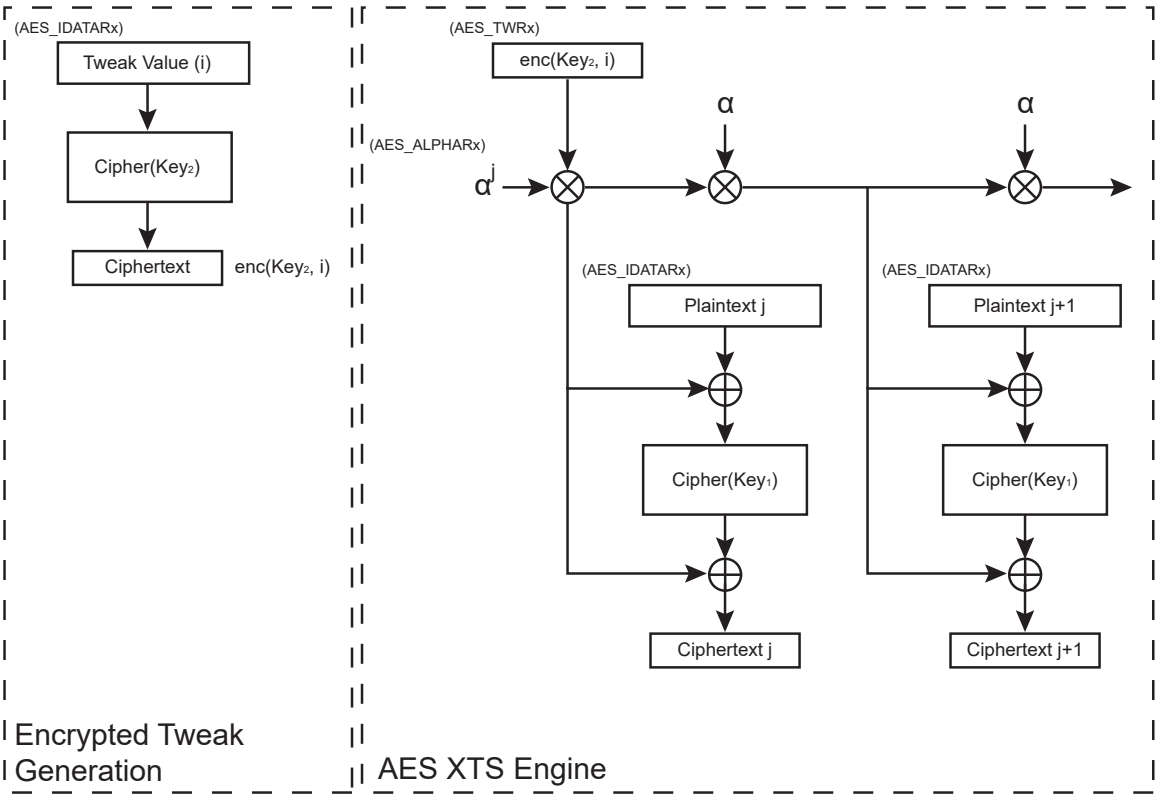
Note: AES_GHASHRx.GHASH can be initialized with a value C between step 3 and step 4 to run a ((A XOR C) x B) GF_{128} multiplication.

56.4.5. 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 56.8. XTS Block Diagram



56.4.5.1.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.

56.4.5.1.1.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.

56.4.5.1.2.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.

56.4.6. 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.

56.4.7. Temporary Secured Storage for Keys

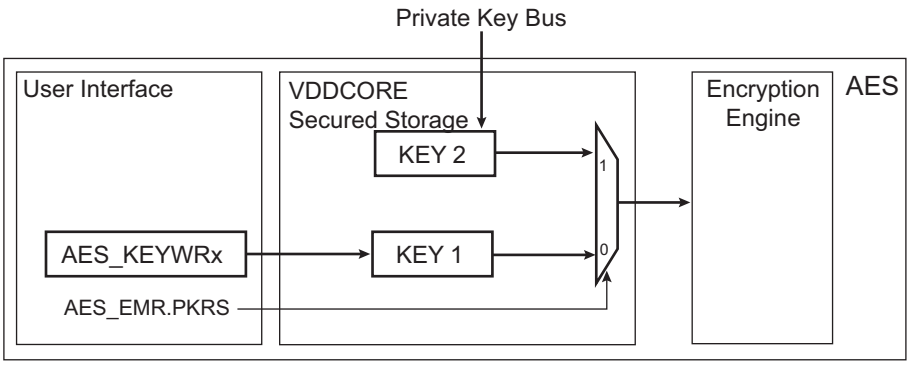
The AES provides secure storage for up to 2x 256-bit keys. The storage is available while VDDCORE voltage is supplied.

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 56.9. Temporary Secured Storage for Keys



56.4.8. Start Modes

AES_MR.SMOD allows selection of the encryption (or decryption) Start mode.

56.4.8.1.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 56.2. 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.

56.4.8.2.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.

56.4.8.3.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 56.3. 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

56.4.9. 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.

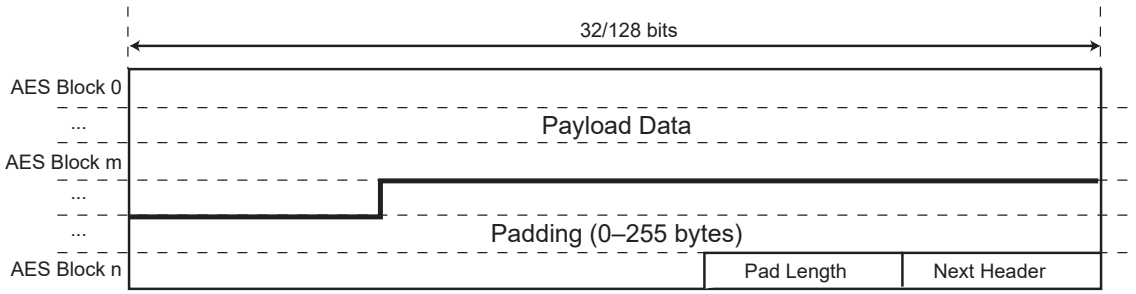
The auto padding feature only supports CBC and CTR modes.

Note: When automatic padding is enabled and AES_MR.SMOD=2, AES_MR.DUALBUFF must be cleared.

56.4.9.1. 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 56.10. 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).

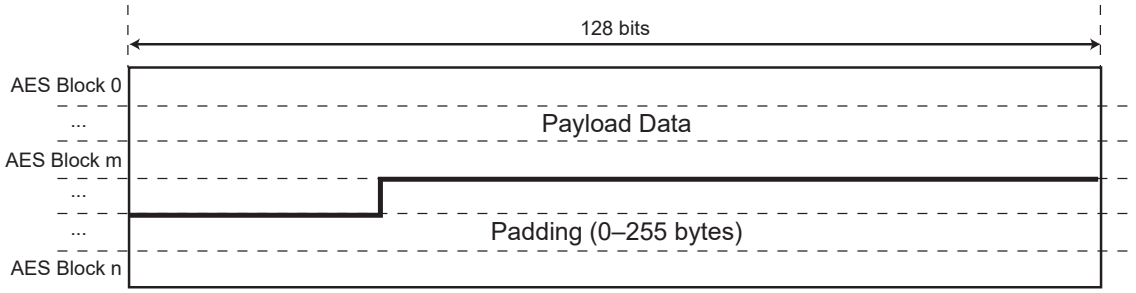
To process an IPSec message using auto-padding, the sequence is as follows:

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.

56.4.9.2.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 56.11. 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.

56.4.9.3. 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.

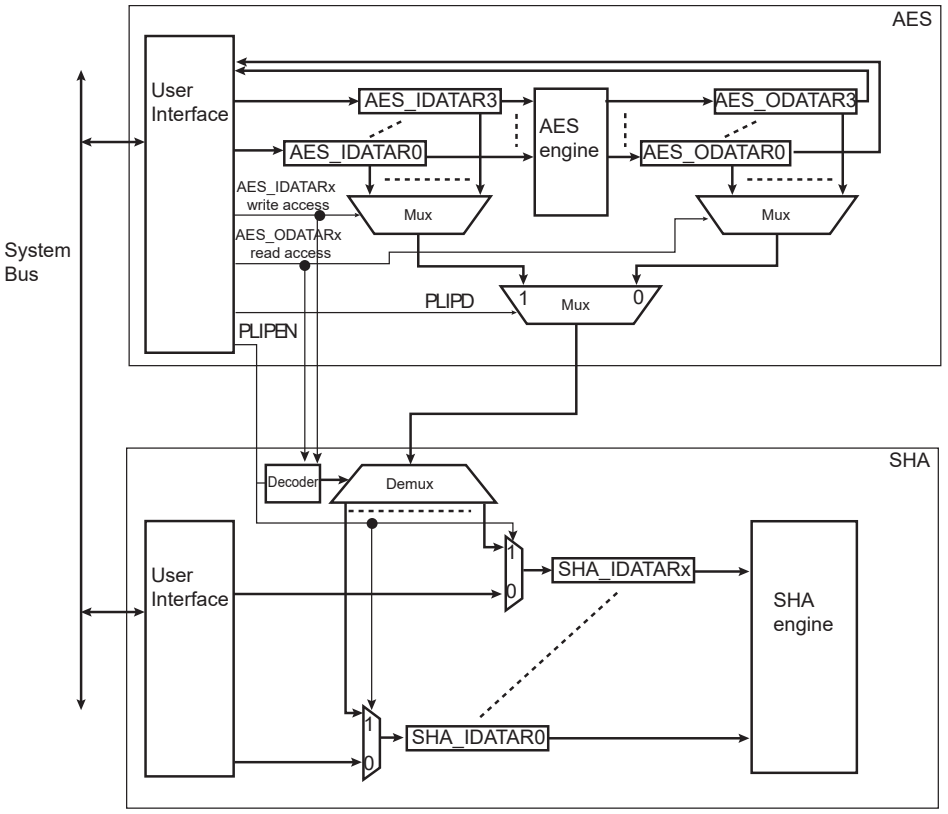
In IPSec/SSL standard message length including padding must be a multiple of the AES block size when CBC mode is used and multiple of 32-bit if CTR mode is used.

56.4.10. 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 56.12. Secure Protocol Layers Improved Performances Block Diagram



56.4.10.1.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.

56.4.10.2.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.

56.4.10.3.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.

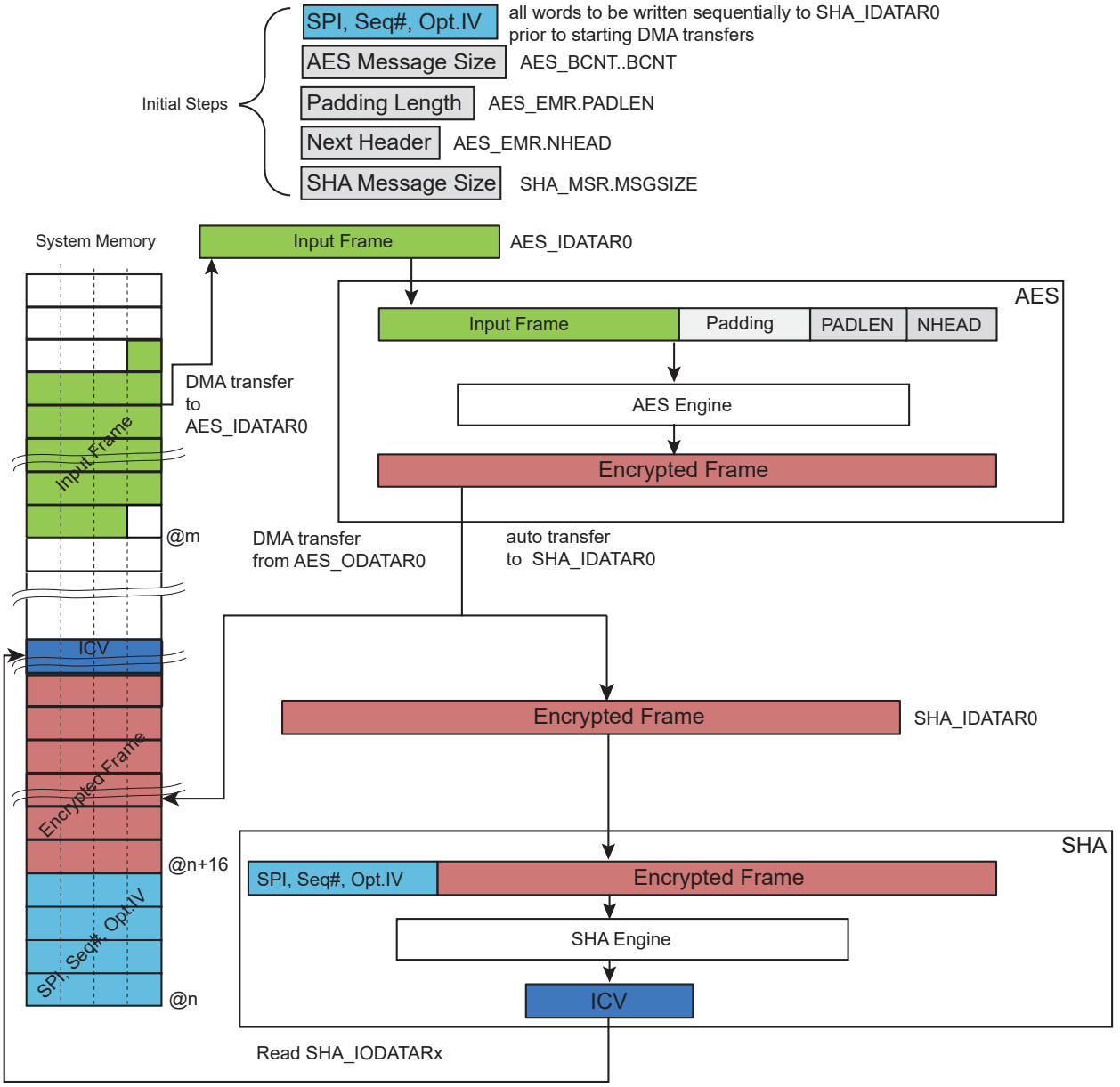
When the AES module is configured to improve the performance of the secure protocol layers (AES_EMR.PLIPEN = 1), the data transfers are simplified, limiting the bandwidth requirements on the system bus.

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.

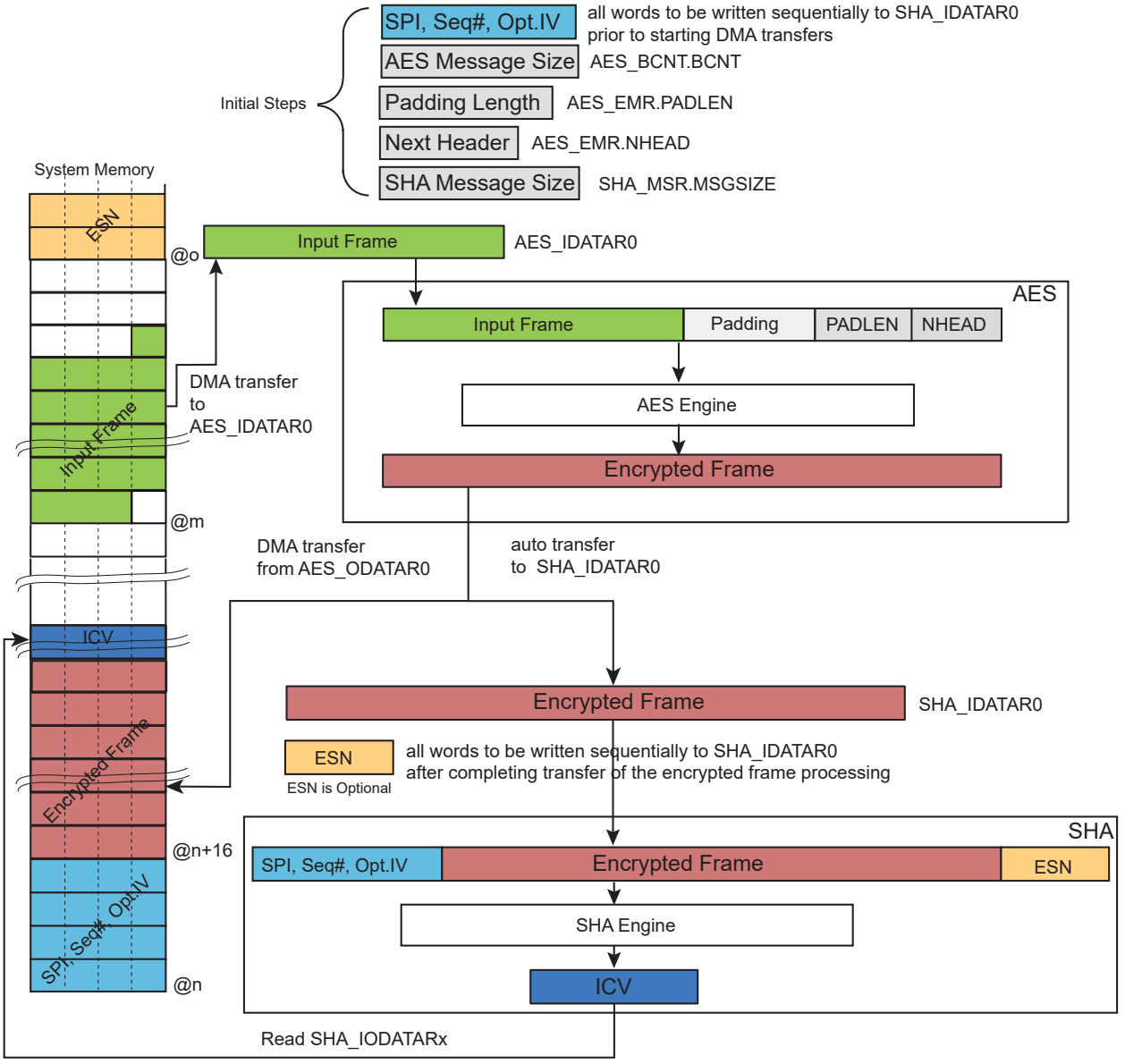
See the following figures.

Figure 56.13. 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 56.14. 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.

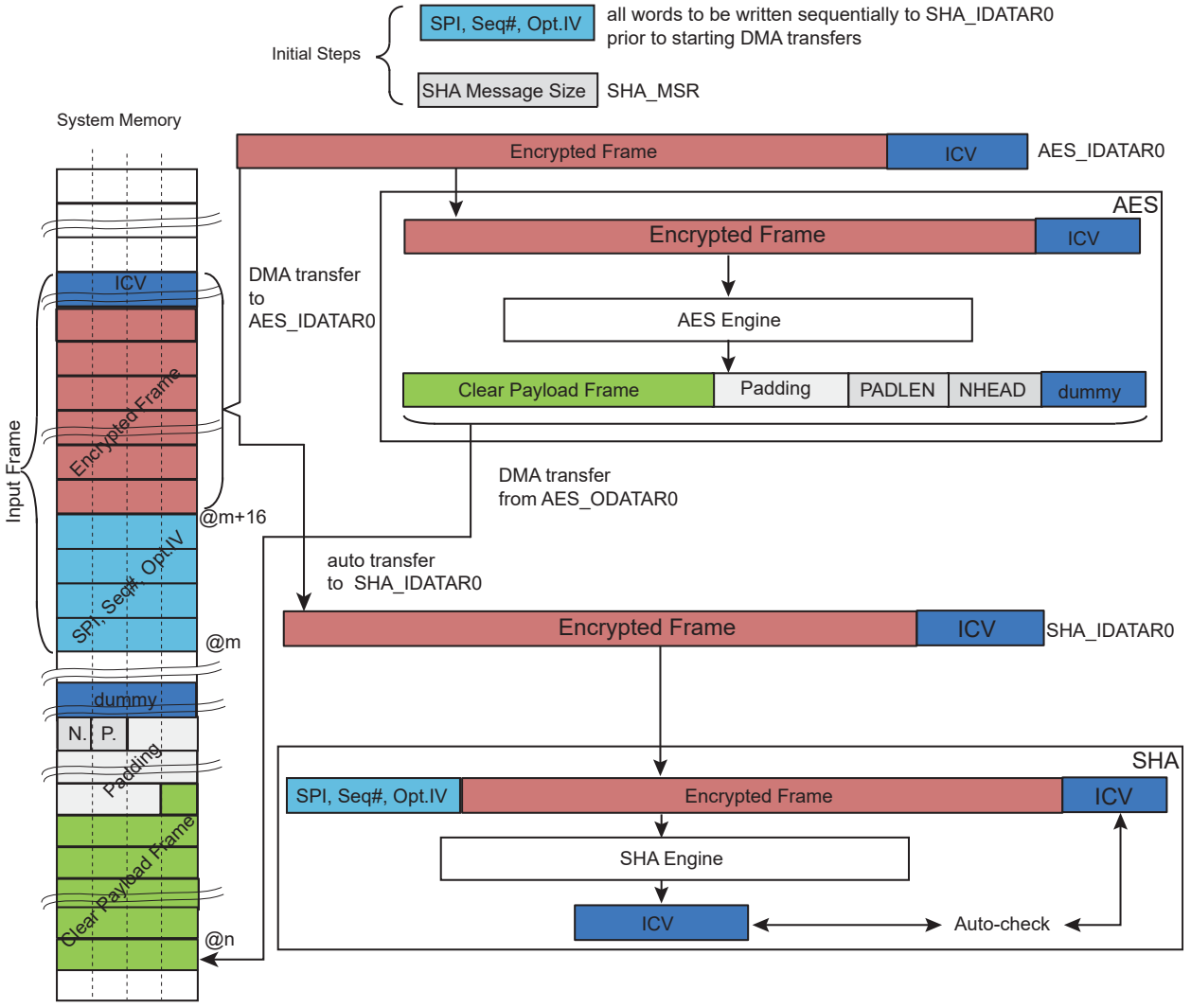
- A first DMA transfer descriptor must be configured to load the received encrypted frame from the system memory to AES_IDATARx for decryption. The start address of the first transfer descriptor must be defined after the SPI, SEQ#, and optional IV (see the following figure).
- A second DMA descriptor 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.

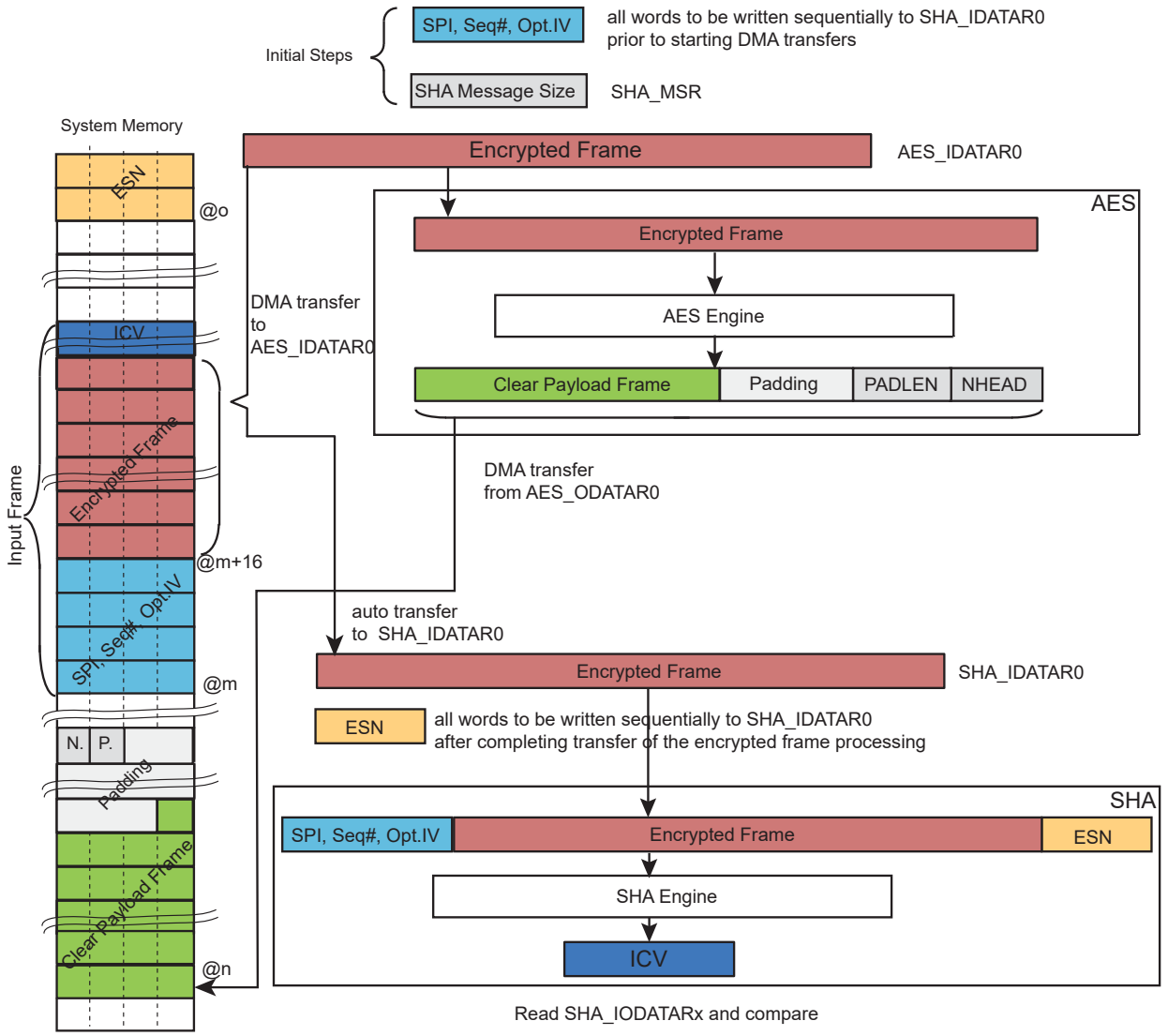
Figure 56.15. Decryption of an ESP IP Sec Frame without ESN



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 56.16. Decryption of an ESP IPSec Frame with ESN



56.4.11. Security Features

56.4.11.1.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.

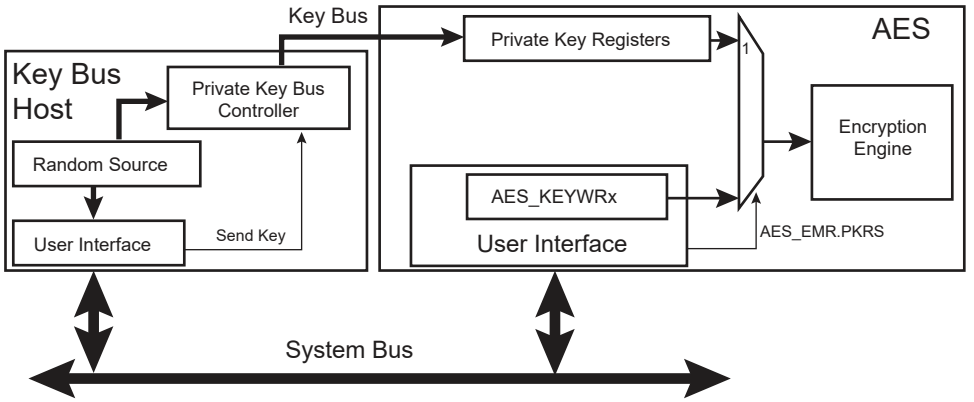
To select the Private Key internal registers as the source of the AES key, AES_EMR.PKRS must be written to '1'.

When AES_EMR.PKRS is modified, it is mandatory to load the corresponding key value even if the key has been previously written with same value.

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.

Figure 56.17. Key Selection



56.4.11.2.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.

56.4.11.3.Clearing Key on Tamper Event

On a tamper detection event on WKUP[8:1] 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)”.

56.4.11.4.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

56.4.11.5.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.

56.5. 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								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]							
0x3C	AES_KEYWR7	31:24	KEYW[31:24]							
		23:16	KEYW[23:16]							
		15:8	KEYW[15:8]							
		7:0	KEYW[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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									
0xB0	AES_EMR	31:24	BPE							
		23:16	NHEAD[7:0]							
		15:8	PADLEN[7:0]							
		7:0	PKRS		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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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

56.5.1. 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
								START
Access								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 0 – START Start Processing

Value	Description
0	No effect.
1	Starts manual encryption/decryption process.

56.5.2. AES Mode Register

Name: AES_MR
Offset: 0x04
Reset: 0x00080000
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.

56.5.3. 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

56.5.4. 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

56.5.5. 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

56.5.6. 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.
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.

56.5.7. 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.

56.5.8. 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.

56.5.9. 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.

56.5.10. 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

56.5.11. 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.

56.5.12. 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.

56.5.13. 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.

56.5.14. 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.

56.5.15. 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.

56.5.16. 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.

56.5.17. 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		PLIPD	PLIPEN			APM	APEN
Access	R/W		R/W	R/W			R/W	R/W
Reset	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 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.

56.5.18. 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.

56.5.19. 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.

56.5.20. 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.

56.5.21. 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.

Value	Name	Description
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.
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).

56.5.22. 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.

57. Secure Hash Algorithm (SHA)

57.1. 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.

57.2. Embedded Characteristics

- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256, SHA384, SHA512)
- Supports Hash-based Message Authentication Code (HMAC) Algorithm (HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512)
- 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

57.3. Product Dependencies

57.3.1. 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.

57.3.2. Interrupt Sources

The SHA interface has an interrupt line connected to the Interrupt Controller.

Handling the SHA interrupt requires programming the Interrupt Controller before configuring the SHA.

57.4. 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.

57.4.1. SHA Algorithm

The SHA can process SHA1, SHA224, SHA256, SHA384, SHA512 by configuring the ALGO field in the Mode register (SHA_MR).

57.4.2. 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) || (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.

57.4.3. Processing Period

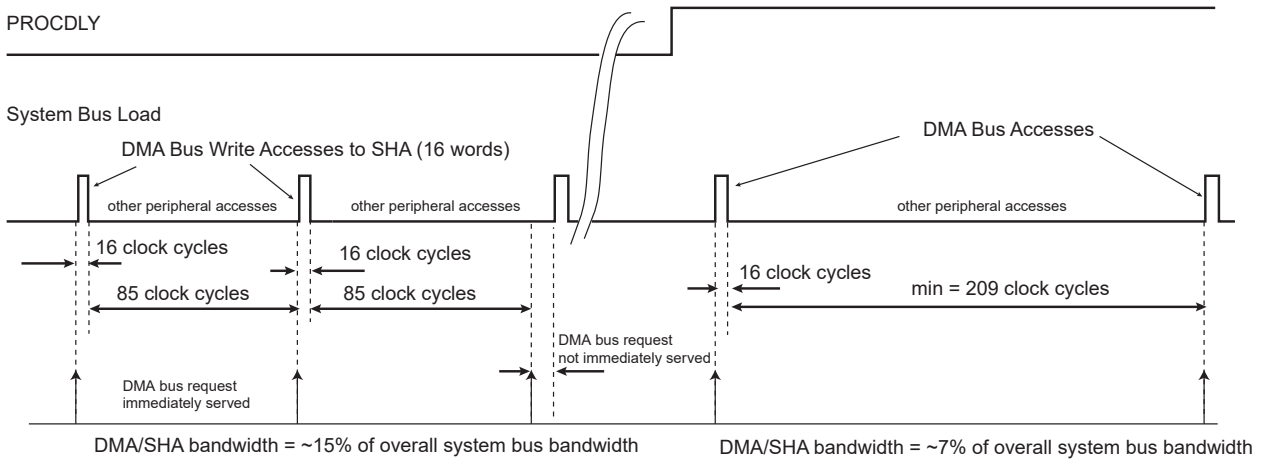
When SHA is enabled and DMA is used to write the messages, the inherent processing period may result, depending on the application, in a significant bandwidth usage at system bus level. In some applications, it may be important to keep as much bandwidth as possible for the other peripherals (e.g. CPU, 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 57.1. Bandwidth Usage in SHA-1 Mode



57.4.4. 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.

57.4.5. 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 57.1. 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)

Table 57.1. Configuration Values of Internal Registers (continued)

Register	SHA Modes (ALGO < 8)			HMAC Modes (ALGO > 7)
	CHECK = 0	CHECK = 1	CHECK = 2	
IR1	User Initial Hash	Expected Hash Result	User Initial Hash	hash($K_0 \oplus$ opad)

To calculate the initial HMAC values, follow this sequence:

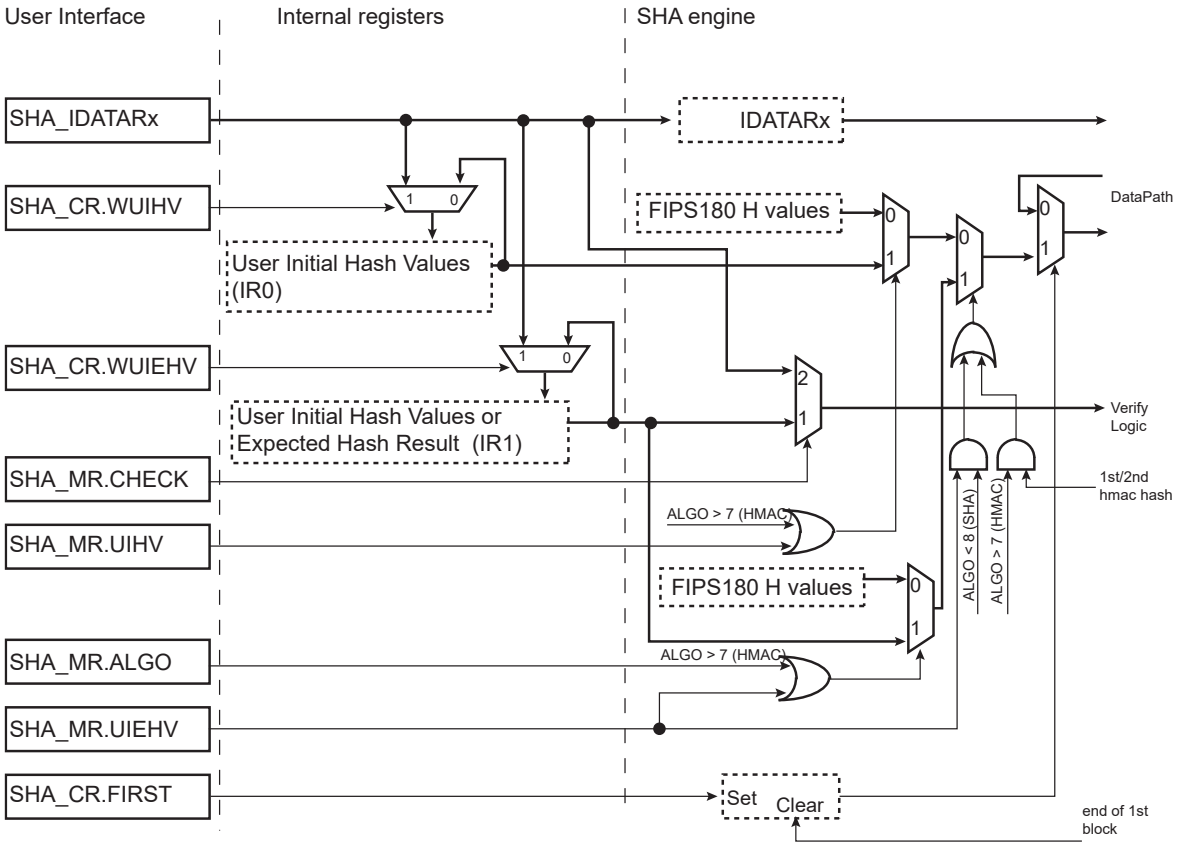
1. Calculate K_0.
2. Calculate $K_0 \oplus$ ipad and $K_0 \oplus$ opad.
3. Perform a hash of the result of $K_0 \oplus$ ipad and $K_0 \oplus$ opad (auto-padding must be disabled for that type of hash).
4. Write h($K_0 \oplus$ ipad) and h($K_0 \oplus$ opad) in IR0 and IR1 respectively.

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
   - For expected hash result:
     - SHA_IDATAR0 to SHA_IDATAR4 for SHA1
     - SHA_IDATAR0 to SHA_IDATAR6 for SHA224
     - SHA_IDATAR0 to SHA_IDATAR7 for SHA256
     - 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 57.2. User Initial Hash Value and Expected Hash Internal Register Access



57.4.6. 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.

57.4.7. 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).

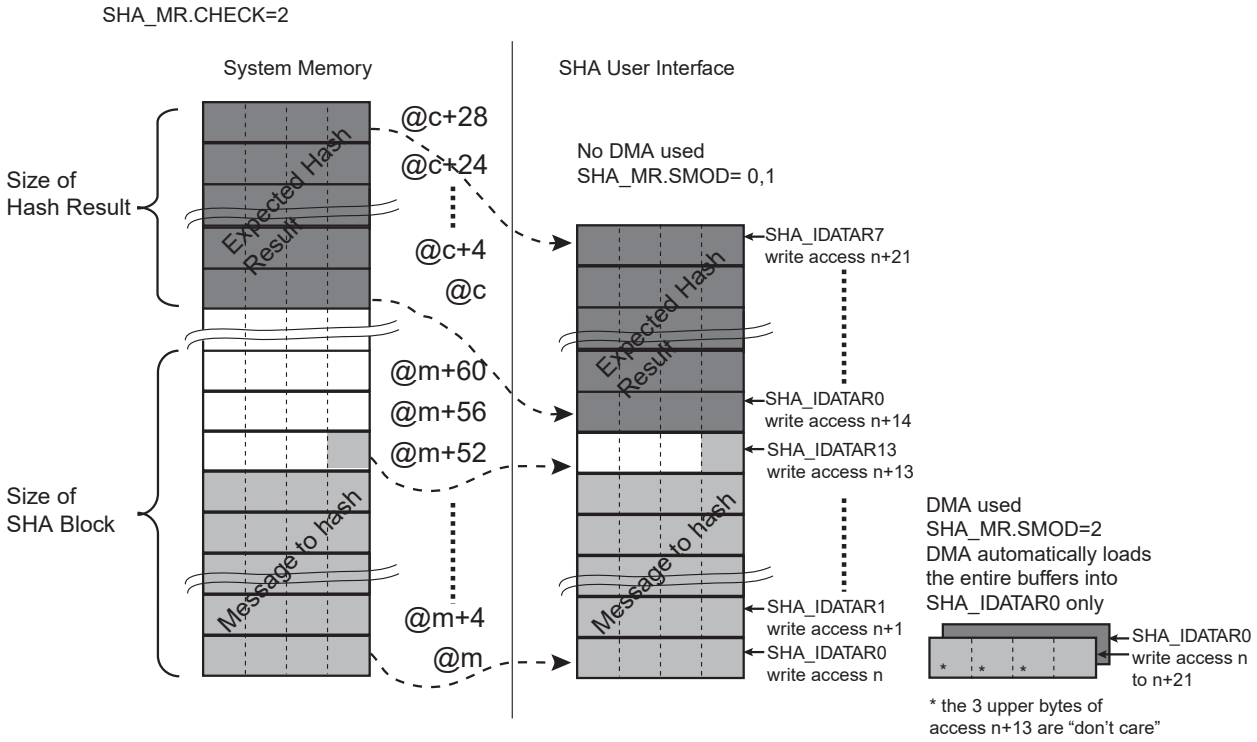
Automatic check requires the automatic padding feature to be enabled (MSGSIZE and BYTCNT fields must be greater than 0).

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 57.3. 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.

57.4.8. 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.

57.4.9. Start Modes

SHA_MR.SMOD is used to select the Hash Processing Start mode.

57.4.9.1. 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.

57.4.9.2. 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.

57.4.9.3. 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 57.4. Interrupts Processing with DMA

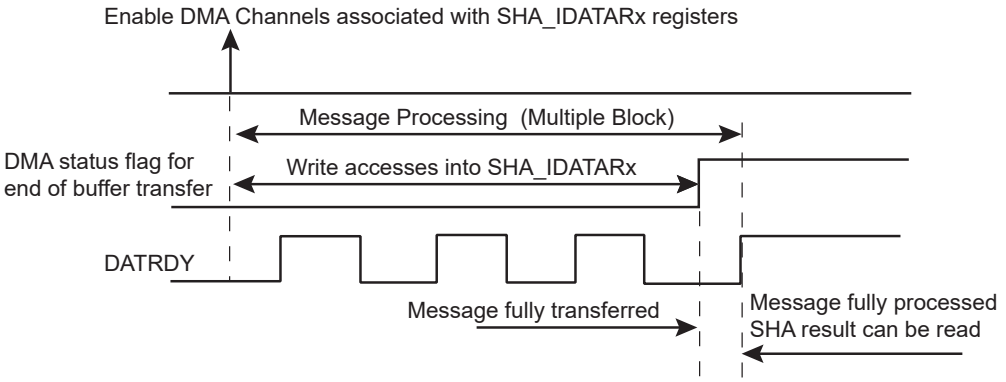
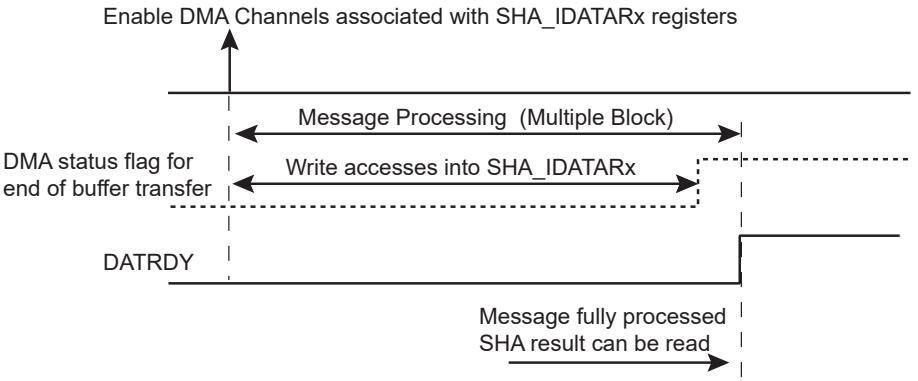


Figure 57.5. Interrupts Processing with DMA and Automatic Padding



57.4.9.4.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:

0x61626380000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000018

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

57.4.10. Security Features

57.4.10.1. 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.

57.4.10.2.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

57.4.10.3.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.

57.5. 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]							
...										
0xBC	SHA_IODATAR15	31:24	IODATA[31:24]							
		23:16	IODATA[23:16]							
		15:8	IODATA[15:8]							
		7:0	IODATA[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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

57.5.1. 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.

57.5.2. 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
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	Reserved	–
14	Reserved	–

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)

57.5.3. 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

57.5.4. 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

57.5.5. 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

57.5.6. 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.

57.5.7. 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.

57.5.8. 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.

57.5.9. 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.

57.5.10. 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.

57.5.11. 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).

57.5.12. 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.

Value	Description
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.

58. Triple Data Encryption Standard (TDES)

58.1. 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.

58.2. 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

58.3. Product Dependencies

58.3.1. 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.

58.3.2. Interrupt Sources

The TDES interface has an interrupt line connected to the Interrupt Controller. In order to handle interrupts, the Interrupt Controller must be programmed before configuring the TDES.

58.4. 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 58.1. 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.

58.4.1. 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).

58.4.2. Temporary Secured Storage for Keys

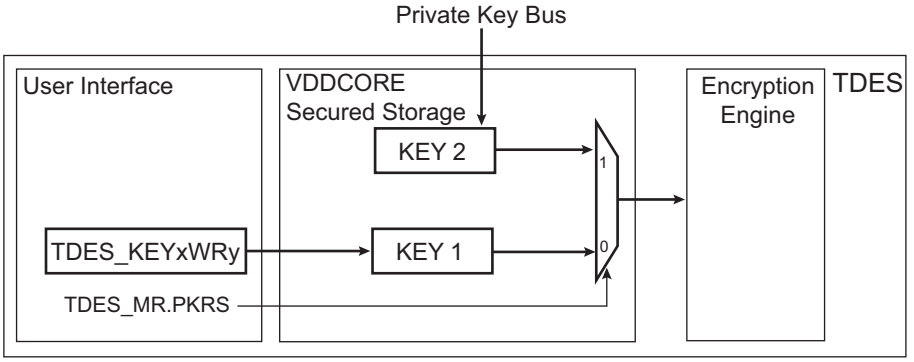
The TDES provides secure storage for two sets of three 64-bit keys. The storage is available while VDDCORE voltage is supplied.

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 58.1. Temporary Secured Storage for Keys



58.4.3. Start Modes

TDES_MR.SMOD selects the Encryption (or Decryption) start mode.

58.4.3.1. 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 58.2. 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

58.4.3.2. 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.

58.4.3.3. 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 58.3. 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

Table 58.3. DMA Data Transfer Type for the Different Operating Modes (continued)

Operating Mode	Chunk Size	Destination/Source Data Transfer Type
CFB 64-bit	1	Word
CFB 32-bit	1	Word
CFB 16-bit	1	Half-word
CFB 8-bit	1	Byte

58.4.4. 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 Table 58.4).

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.

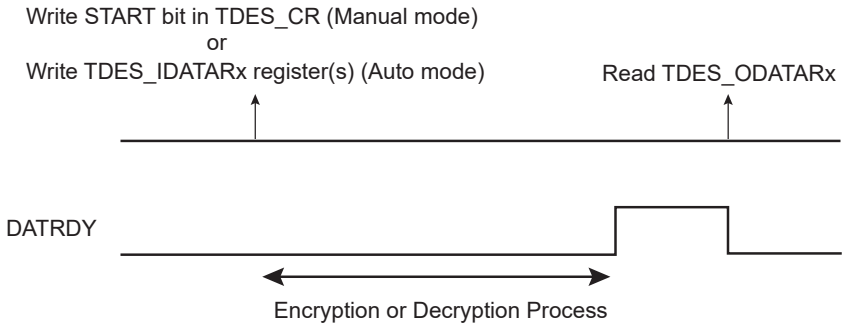
Therefore, there is no need to define a read buffer in DMA mode.

58.4.4.1.Manual and Auto Modes

58.4.4.1.1.TDES_MR.LOD = 0

The DATRDY flag is cleared when at least one TDES_ODATARx is read. See the figure below.

Figure 58.2. Manual and Auto Modes with LOD = 0

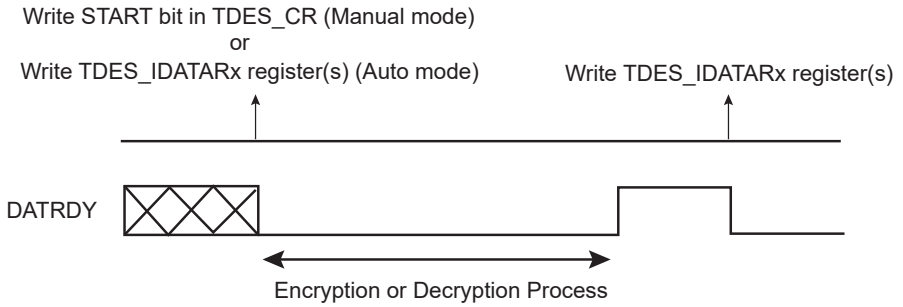


If the user does not want to read TDES_ODATARx between each encryption/decryption, the DATRDY flag will not be cleared. If the DATRDY flag is not cleared, the user will not be informed of the end of the encryptions/decryptions that follow.

58.4.4.1.2.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 58.3. Manual and Auto Modes with LOD = 1



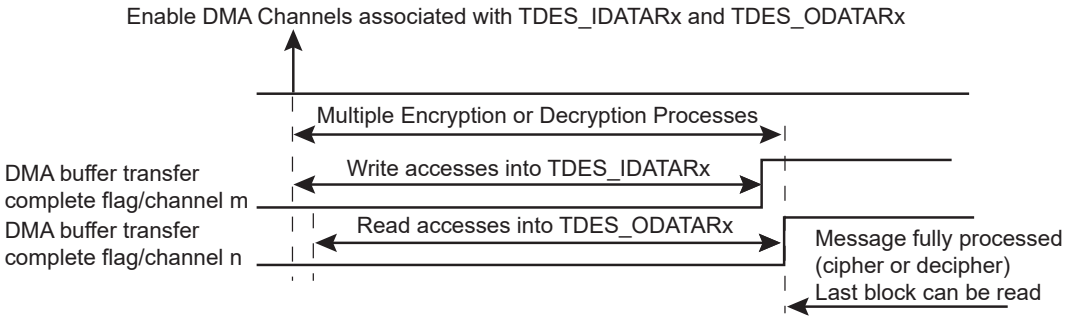
58.4.4.2.DMA Mode

58.4.4.2.1.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 58.4. DMA Transfer with LOD = 0



58.4.4.2.2.TDES_MR.LOD = 1

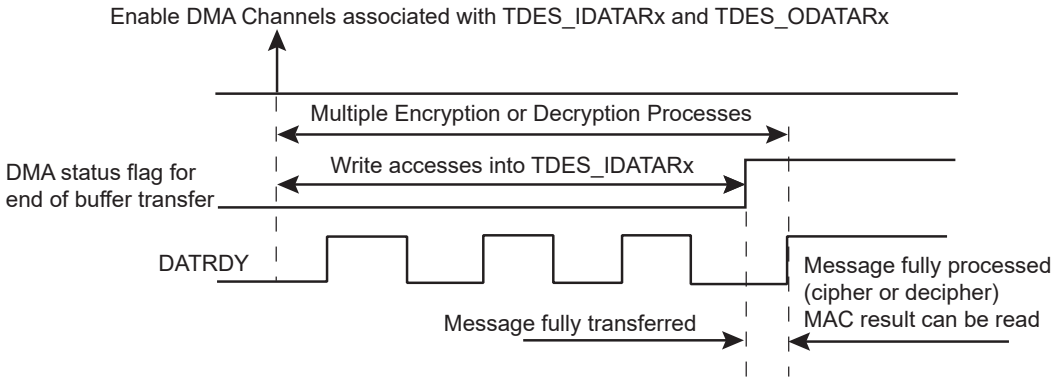
This mode is optimized to process the TDES CBC-MAC operating mode.

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 58.5. DMA Transfer with LOD = 1



The table below summarizes the different cases.

Table 58.4. 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.

58.4.5. Security Features

58.4.5.1. 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.

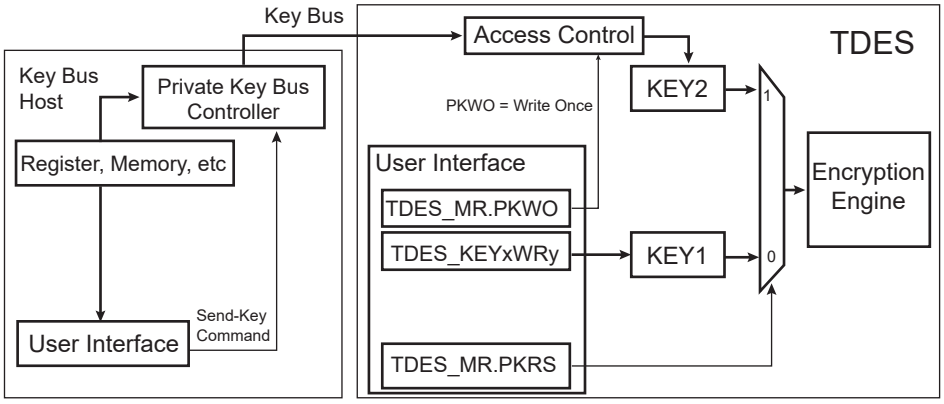
To select the Private Key internal registers as the source of the TDES key, TDES_MR.PKRS must be written to '1'.

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 58.6. 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.

58.4.5.2. 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.

58.4.5.3. Clearing Key on Tamper Event

On a tamper detection event on WKUP[8:1] 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).

58.4.5.4. 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

58.4.5.5. 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). For example, 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.

58.5. 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									
0x40	TDES_IDATAR0	31:24	IDATA[31:24]							
		23:16	IDATA[23:16]							
		15:8	IDATA[15:8]							
		7:0	IDATA[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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

58.5.1. 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.

58.5.2. TDES Mode Register

Name: TDES_MR
Offset: 0x04
Reset: 0x00000002
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	1	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 bits
1	SIZE_32BIT	32 bits
2	SIZE_16BIT	16 bits
3	SIZE_8BIT	8 bits

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.
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.

58.5.3. 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

58.5.4. 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

58.5.5. 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

58.5.6. 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.

Value	Description
1	Encryption or decryption process is completed.

58.5.7. 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.

58.5.8. 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.

58.5.9. 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.

58.5.10. 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.

58.5.11. 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.

58.5.12. 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.

58.5.13. 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.

58.5.14. 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).

58.5.15. 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.

59. Random Number Generator (TRNG)

59.1. Description

The 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.

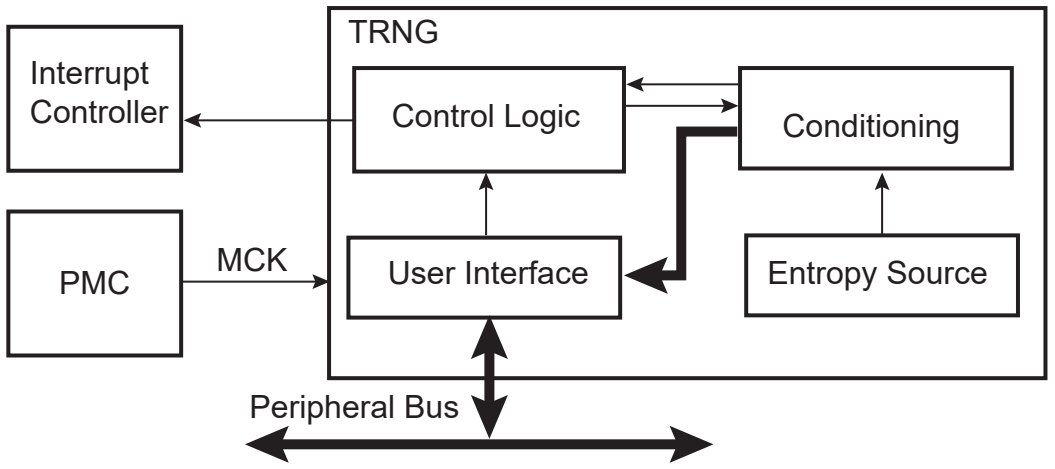
The TRNG is fully designed with digital cells, and under the specified operating conditions, external factors such as temperature, humidity, etc. affect TRNG ageing in the same manner as all other digital peripherals (CPU core, bus matrix, etc.) of the product.

59.2. 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

59.3. Block Diagram

Figure 59.1. TRNG Block Diagram



59.4. Product Dependencies

59.4.1. 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.

59.4.2. 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.

59.5. 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 (KSLAVE) and length of the key to be generated (KLENGTH).

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.

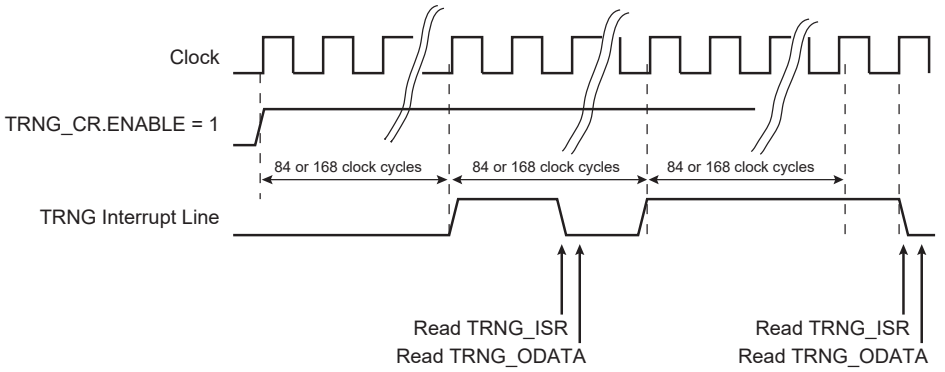
By writing a ‘1’ to the HALFR bit in the Mode register (TRNG_MR), the random values are provided every 168 cycles instead of every 84 cycles. HALFR must be written to ‘1’ when the TRNG peripheral clock frequency is above 100 MHz.

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 59.2. TRNG Data Generation Sequence



Key Bus Operating Mode

After a write to KSLAVE 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 59.3. TRNG Private Key Bus

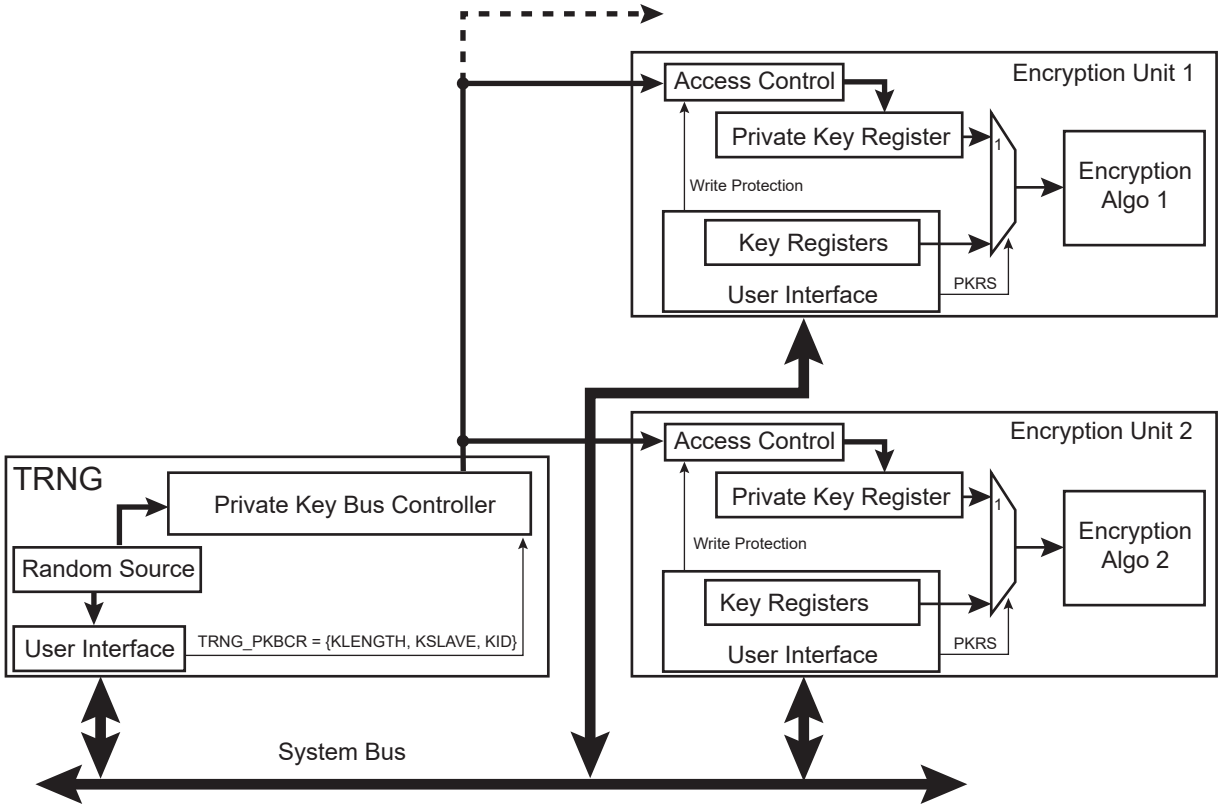
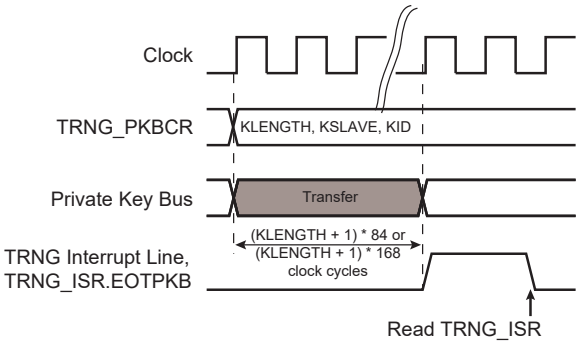


Figure 59.4. TRNG Private Key Bus Transfer



59.5.1. 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.

59.5.2. 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.

59.5.3. Enhanced Conditioning by Difference Checking

By default, the current data available when TRNG_ISR.DATRDY=1 is not compared against the previous value provided together with the previous assertion of the TRNG_ISR.DATRDY flag. There is no formal guarantee that two consecutive data differ. If a difference is required, by default the software must post-process the data.

When TRNG_MR.DIFF=1, any new internal data sample (every 84 or 168 MCK clock cycles) is first checked against the previous data and the TRNG_ISR.DATRDY flag is set only if a difference exists. When TRNG_MR.DIFF=1, the maximum streaming rate can be slightly slower compared to TRNG_MR.UDIFF=0.

59.5.4. 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_MR

The following registers can be write-protected when WPITEN is set:

- TRNG_IER

- TRNG_IDR

The following registers can be write-protected when WPCREN is set:

- TRNG_CR
- TRNG_PKBCR

59.5.5. 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). For example, 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.

59.6. 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	DIFF							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

59.6.1. 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.

59.6.2. 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
	DIFF							HALFR
Access	R/W							R/W
Reset	0							0

Bit 7 – DIFF Minimum Hamming Distance

Value	Name	Description
0	DISABLED	Delivers a new random sample without condition with the previous sample (unless HD=1).
1	ENABLED	Delivers a new random sample only if it differs from the previous delivered sample (unless HD=1).

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).

59.6.3. 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	OTPC_ID	OTPC
3	Reserved_ID	Reserved

Bit 0 – KID Key ID (Must always be written to 0)

59.6.4. 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

59.6.5. 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

59.6.6. 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

59.6.7. 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.

59.6.8. 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.

59.6.9. 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).

59.6.10. 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).
5	LOCK_ERR	A write access to TRNG_WPMR has been attempted when one of the write protection bits is already locked, its corresponding lock control bit is set and the corresponding write protection bit is cleared, which looks like an unlock tentative (Warning).

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.
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.

60. Physical Unclonable Functions (PUF)

60.1. Description

Physical Unclonable Functions (PUF) use the behavior of standard embedded static RAM (SRAM) to differentiate chips from each other. They are virtually impossible to duplicate, clone or predict. This makes them very suitable for applications such as secure key generation and storage, device authentication, flexible key provisioning and chip asset management.

The start-up values of uninitialized SRAM provide randomness. These values form a unique chip fingerprint, called the SRAM PUF response.

An SRAM PUF response is a noisy fingerprint, and turning it into a high-quality and secure key vault is done with the PUF controller. The PUF controller reliably reconstructs the same cryptographic key under all environmental circumstances.

The PUF generates an activation code during the enroll phase that allows further operations.

The activation code, in combination with the SRAM start-up behavior, is used to reconstruct, on demand, in real time, an intrinsic PUF key which is never stored. When the key is no longer needed, it can be removed from memory. When it is needed later it can be reconstructed.

The intrinsic PUF key can be used as a root key for key derivation and wrapping. A key protected by the PUF controller is integrity protected and can be retrieved only on the same device while it will be meaningless on other devices.

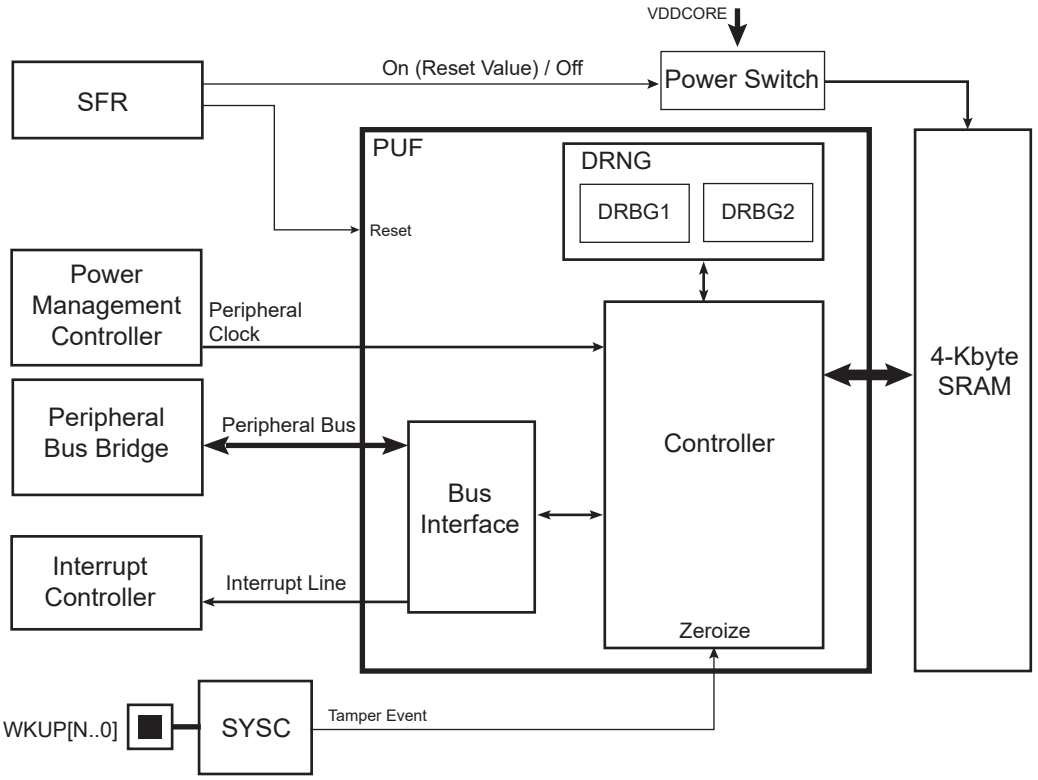
The benefits include device-unique high-quality keys, no secrets when power is off, and no root key programming.

60.2. Embedded Characteristics

- 256-bit Key Entropy
- On-chip Enrollment
- Key Derivation and Wrapping
- Deterministic Random Number Generator (DRNG) According to NIST SP800-90A/B
- PUF Diagnostics
- Built-In Self-Test (BIST)
- Functional Safety Monitors and Reports
  - Abnormal software access reports (undefined address, writing a read-only register, reading a write-only register)

60.3. Block Diagram

Figure 60.1. PUF Block Diagram



60.4. Product Dependencies

60.4.1. SRAM Power Supply

The power supply of the PUF-dedicated SRAM can be enabled or disabled in the Special Function registers. Refer to the section “Special Function Registers (SFR)” for more details. The PUF SRAM must be powered prior to using the PUF controller. After a power-up sequence, the PUF SRAM is powered.

60.4.2. PUF Reset

The reset of the PUF is controlled by a configuration bit in the SFR. The PUF is active after a power-up sequence and the reset deassertion.

60.4.3. Peripheral clock

The PUF peripheral clock must be enabled in the PMC prior to using the PUF.

60.4.4. Interrupt Sources

The PUF has an interrupt line connected to the interrupt controller. In order to handle interrupts, the interrupt controller must be programmed before configuring the PUF controller.

60.4.5. Special Function Register (SFR)

Refer to the section “Special Function Registers (SFR)” for all configurations related to PUF.

60.5. Functional Description

60.5.1. PUF Introduction

The PUF controller provides the following functionalities:

- Enrollment (usually only needed once)
- Reconstruction (Start or Reconstruct)
- Key derivation
- Wrapping and unwrapping of user-defined keys
- Wrapping and unwrapping of random intrinsic keys
- Random generation
- Zeroization from software or hardware external event
- Test functionality

60.5.2. PUF Operations

During start-up, the PUF controller first tests the PUF-dedicated SRAM for defects. If defects are found, the PUF SRAM is disqualified, and the PUF controller does not allow any operations. This prevents security breaches due to a malfunctioning PUF.

Enrollment is done to obtain the device's intrinsic PUF key, and as a result creates helper data (the activation code). This code must be stored in a non-volatile memory area. From then on, the device's activation code can be provided to reconstruct the intrinsic PUF key.

When the intrinsic PUF key is available (which is the case after enrollment and after successful reconstruction), key operations can be performed.

Table 60.1. List of Operations

Operation Name	Runtime (in clock cycles)
Initialization	36800 with memory test
	26400 without memory test
	14550 for a warm reset
Enroll	33800
Start (12.5% PUF noise)	53400
Start (worst case PUF noise)	55650
Reconstruct (12.5% PUF noise)	40450
Reconstruct (worst case PUF noise)	42700
Stop	1600
Get Key (256 bits)	1200
Wrap Generated Random (256 bits)	6000
Wrap (256 bits)	4450
Unwrap (256 bits)	2850
Generate Random (256 bits)	1650
Reseed (no external entropy)	4600
Reseed (external entropy)	5800
Test Memory	12800
Test PUF	2400
BIST	57100
Zeroize	1600

When all required key operations have been completed, a Stop command removes all key material from the PUF controller. When more key operations need to be performed later, this can be enabled by a new reconstruction.

With PUF DRNG, bit strings can be generated for use in other parts of the system (for example, as IV, session key, nonce, etc.).

A Built-In Self-Test (BIST) is available to test digital logic in the field. It clears and then tests all digital logic. The result is provided via a register. After BIST has completed, operations can be resumed as if the PUF controller had just been reset.

60.5.3. Security Strength Versus Key Length

The PUF controller security strength is 256 bits, which reflects the strength of its internal secrets and hence determines the highest level of cryptographic security that any of PUF controller cryptographic operations can offer. Depending on the operation and the provided parameters, the effective security strength can be lower, but never higher than 256 bits.

60.5.4. PUF Controller States

The PUF controller provides several operations which are triggered by software commands or hardware events. For several reasons (security, etc.), not all operations are available at all times. The following figure illustrates the operation capabilities.

Figure 60.2. Operation Diagram

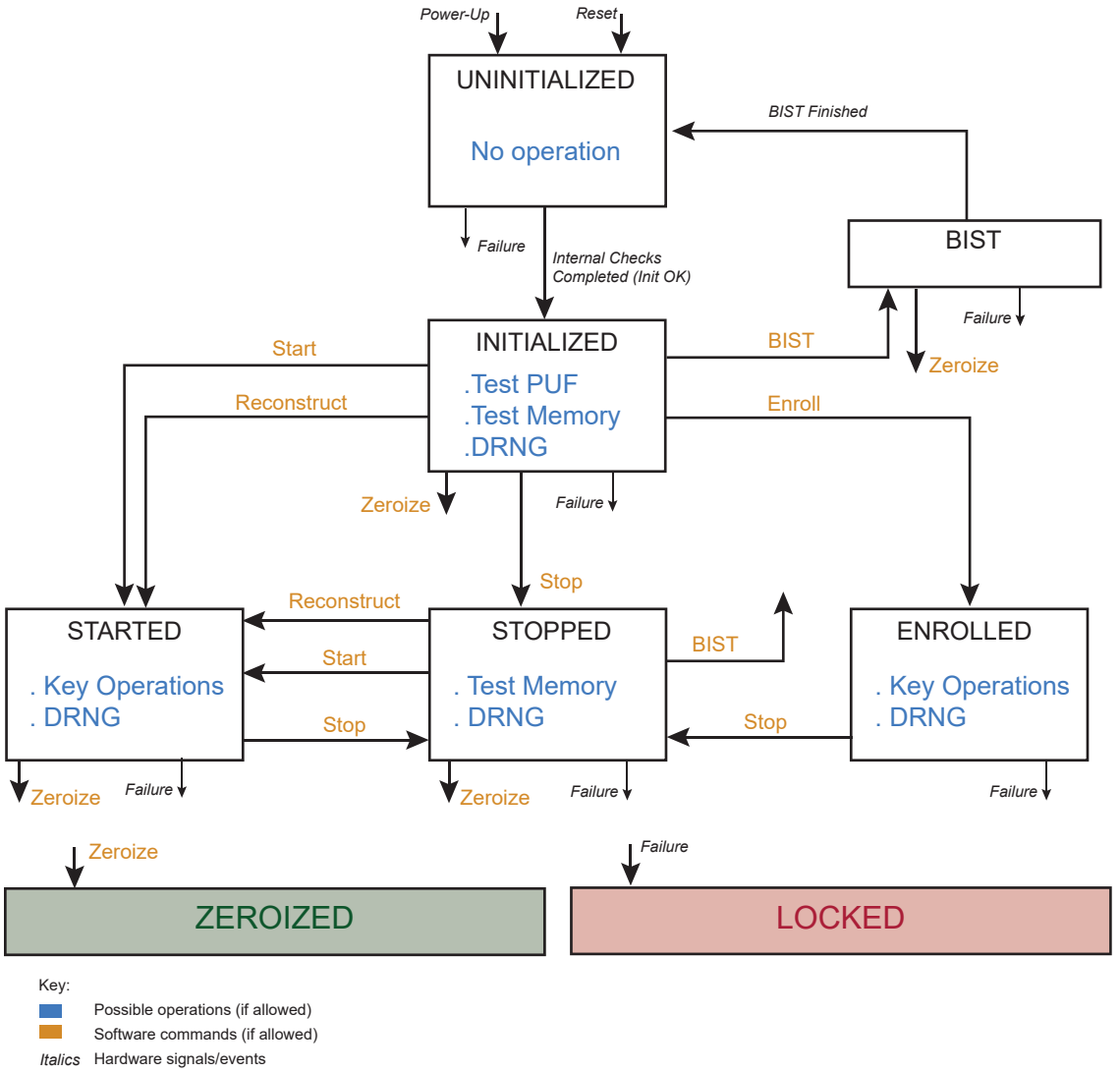


Figure 60.3. Operation Diagram

After power-up and reset is released (refer to the section “Special Function Registers (SFR)” for details on reset deassertion), the PUF controller begins in the Uninitialized state, and runs its initialization sequence (see Initialization Operation). This is indicated by the flag BUSY=1 in the Status register (PUF_SR).

Reset takes precedence over all PUF controller functionalities. As long as the reset is held active, the PUF controller stays in the Uninitialized state, and commands or hardware events have no effect.

When initialization finishes successfully, the PUF controller moves to the Initialized state. It moves to the Locked state on failure.

In the Initialized state, several operations can be performed: BIST, Test PUF, Test SRAM, Generate Random, Reseed, Enroll, Start, Reconstruct, Stop.

Note: An operation can only be performed when it is not disabled (See PUF Hardware Settings Register). This is valid in all states. PUF Hardware Settings Register is controlled by the SFR.

After a successful Enroll operation (see Enroll Operation), the PUF controller is in the Enrolled state and can perform a Generate Random operation, a Reseed, key operations or a Stop operation (when no further actions is required at that moment).

After a successful Start (see Start Operation) or Reconstruct (see Reconstruct Operation), the PUF controller is in the Started state. In this state, a Generate Random operation, a Reseed, key operations or a Stop operation can be done.

A Stop operation (see Stop Operation) brings the PUF controller to the Stopped state. In this state, no sensitive data is present in the controller and the following operations can be performed: BIST, Test Memory, Generate Random, Reseed, Start, Reconstruct.

When in Started or Enrolled state, key operations can be performed (see Get Key Operation to Unwrap Operation). After such an operation is complete, the PUF controller returns to the state it was in before the operation.

In an Initialized, Enrolled, Started or Stopped state, random data can be generated with the Generate Random command (see Generate Random Operation). The random data is output via the Data Output register (PUF_DOR) in the user interface.

With the Reseed operation (see Reseed Operation), the DRNG is seeded with new entropy and its reseed counter is reset. This can be done at any moment that another operation can be started. The entropy can be provided via the Data Input register (PUF_DIR) in the user interface.

In Initialized or Stopped state, the PUF SRAM can be tested with the Test Memory operation (see Test Memory Operation). Details on the memory test are provided in PUF SRAM Test.

With the Test PUF operation (see Test PUF Operation), diagnostic information about the PUF quality is collected and presented in the Score register (PUF_PSR). This operation is intended for production test purposes. It can only be executed once per reset or power cycle. Details on the diagnostics are provided in PUF Diagnostics.

The Zeroize command (see Test PUF Operation) erases all critical security parameters and prevents the PUF controller from executing any more commands by entering the Zeroized state. The only way to leave this state consists in power-cycling the device, which puts the PUF controller in Uninitialized state and starts initialization. This command can be run via the Control register (PUF_CR).

If an operation is unsuccessful, the PUF_SR.ERROR flag is set. In this case, the PUF controller returns to the state it was in when the command was issued.

If a failure (unrecoverable error) occurs during any of the above-mentioned operations (including Initialization and Zeroize), the PUF controller goes to the Locked state. In this state, no commands can be executed except Zeroize. After a reset, the PUF controller attempts to initialize.

Errors and failures are detailed in PUF Error Handling. The Locked and Zeroized states are indicated by the OK, ERROR and ZEROIZED flags in PUF_SR. See the following table.

Table 60.2. Locked and Zeroized State Details

PUF_SR Flag	Locked State	Zeroized State
OK	0	1
ERROR	1	0
ZEROIZED	1	1

A BIST operation (see PUF Built-In Self-Test (BIST)) can be performed when the PUF controller is in Initialized or Stopped state.

During BIST, no other operations can be performed. After BIST has finished, the PUF controller enters the Uninitialized state and begins initialization. It behaves the same way as for a reset, except that the BIST result is reported in the Test register (PUF_TEST) with the BISTOK and BISTERR flags.

The following table provides the list of operations with their inputs and outputs. Input, Result and Output columns are defined as follows:

- Input: data (via PUF_DIR) or settings (via other registers)
- Result: indicates whether the result of the operation is provided via PUF_SR and the Operation Result register (PUF_ORR)
- Output: data (via PUF_DOR) or information (via other registers)

Table 60.3. Overview of Operations

Operation	Input	Result	Output
Initialization	None	Yes	None
Enroll	None	Yes	Activation code via PUF_DOR. PUF Score via PUF_PSR
Start	Activation code (2x) via PUF_DIR	Yes	PUF score via PUF_PSR
Reconstruct	Activation code via PUF_DIR	Yes	PUF score via PUF_PSR
Stop	None	Yes	None
Get Key	Key destination context for keys via PUF_DIR	Yes	Keys via PUF_DOR
Generate Random	Context for random via PUF_DIR	Yes	Random data via PUF_DOR
Reseed	External random Entropy via PUF_DIR	Yes	Random data via PUF_DOR
Test Memory	None	Yes	None
Test PUF	None	Yes	PUF score via PUF_PSR
Zeroize	None	Yes	None
BIST	None	None	Bist results via PUF_TEST

60.5.5. PUF Controller Operations

60.5.5.1. Introduction

Programming 1 to a command bit in PUF_CR starts the corresponding operation. Each operation follows the same protocol.

When a command bit is activated and the corresponding bit in the Allow register (PUF_AR) is set to 1, the operation is started and the flags PUF_SR.BUSY=1, PUF_SR.OK=0 and PUF_AR.ALLOW_*=0.

The command bit in PUF_CR is automatically cleared when the command is accepted.

After the operation is finished, the PUF controller sets the following status:

- PUF_SR.BUSY=0, while the OK and ERROR bits show the result of the operation.
- PUF_AR.ALLOW_x bits indicate which commands are allowed next.

When the result is successful, PUF_SR.OK=1 and PUF_SR.ERROR=0 and the PUF controller moves to the next state. If the result is not successful (OK=0 and ERROR =1), the software determines what to do next.

With result codes provided via PUF_ORR.RCODE, the PUF controller shows additional information about the cause of an error. Details about PUF controller errors, result codes and error handling are described in PUF Error Handling.

The PUF_SR.REJECTED flag indicates that a past command was not accepted. It is intended as a signal to software that something went wrong and specific actions are required. The PUF controller sets REJECTED=1 in the following cases:

- More than one command bit is 1 at a time.
- A new command is issued while the current command is still running.
- An unallowed command is issued.
- A command with incorrect or unallowed parameters is issued.

The PUF_SR.REJECTED flag remains at 1 until it is cleared by writing PUF_SR.REJECTED=1. While PUF_SR.REJECTED=1, it is still possible to run commands, and when these commands are valid, the PUF controller accepts them and starts the corresponding operation.

Note: After a system reset or power-up, the PUF controller is initialized; it is busy and no commands are allowed. Any command issued during initialization is rejected.

The Zeroize command differs from the standard protocol: it can be issued at any time and will never be rejected, whether an operation (or initialization) is already running or not.

60.5.5.2. Initialization Operation

Initialization brings the PUF controller to a consistent state after each reset and power cycle.

That operation includes a memory test and initialization of the DRNG. A Known Answer Test and an Entropy Evaluation are performed according to the NIST SP 800-90 recommendations.

After a reset or a power cycle, the PUF controller runs a memory test on the PUF SRAM. The memory test automatically resumes when interrupted by a reset and, once completed, is not repeated until the next power cycle.

60.5.5.3. Enroll Operation

During the Enroll operation, the SRAM start-up values are read, an intrinsic PUF key is generated and the corresponding activation code is generated. The activation code must be stored in a non-volatile memory area for future use.

During enrollment, the PUF controller verifies that the PUF SRAM contains qualitative start-up data. If that is not the case, enrollment is terminated and the PUF controller responds with an error indication. The (partially) provided activation code must then be discarded.

After successful completion of the Enroll operation, the diagnostic information about the PUF quality is provided in PUF_PSR.

Note: The Enroll operation collects the same diagnostic information as the Test PUF operation.

The activation code is not sensitive (it does not contain any information on the corresponding intrinsic PUF key) and is unique for every device and every enrollment. It can be stored in non-secure memory area.

The activation code and the intrinsic PUF key are linked to a device, and cannot be used on another device. Every time an enrollment is performed, a new intrinsic PUF key is generated together with its activation code.

The intrinsic PUF key always stays inside the PUF controller and is never output to the rest of the system.

60.5.5.4. Start Operation

During the Start operation, the SRAM start-up values and the activation code that was generated during enroll are used to reconstruct the intrinsic PUF key. When the activation code of another device is used, reconstruction fails, and the PUF controller responds with an error indication.

After successful completion of the Start operation, diagnostic information about the PUF quality is provided in PUF_PSR (see PUF Diagnostics for details).

Note: The PUF scores returned after a Start (or Reconstruct) operation and Enroll (or Test PUF) operation can differ, because these operations measure PUF quality in different ways.

Start operates in two phases (Reconstruction and Anti-aging), each requiring the activation code. Therefore the activation code must be sent twice during a Start operation.

60.5.5.5. Reconstruct Operation

The Reconstruct operation is the same as the Start operation, except that Reconstruct does not include the Anti-aging phase. This operation can be used instead of a Start operation in the following cases only:

- The PUF SRAM is powered off most of the time.
- The time from power-on or reset to key operations is time-critical.

In other cases, the Start operation must be used.

Important: If the Reconstruct operation is used and the device's PUF SRAM remains powered on for a long time, Stop and Start operations must be launched as soon as possible after the time-critical part of the application has been completed. This applies the Anti-aging procedure and improves the long-term reliability of the PUF controller.

During the Reconstruct operation, the SRAM start-up values and the activation code that was generated during enroll are used to reconstruct the intrinsic PUF key. When the activation code of another device is used, the reconstruction will fail, and the PUF controller will respond with an error indication.

After successful completion of the Reconstruct operation, diagnostic information about the PUF quality is provided in PUF_PSR (see PUF Diagnostics for details).

Note: The PUF scores returned after a Reconstruct (or Start) operation and Enroll (or Test PUF) operation can differ because these operations measure PUF quality in different ways.

60.5.5.6. Stop Operation

The Stop operation removes sensitive data from the PUF controller and the PUF SRAM, and sets the PUF controller to the Stopped state. After this command has finished, key operations can no longer be performed. This can be used when no key operations are expected for an extended period of time (e.g. when the system goes into Sleep mode), and ensures that the intrinsic PUF key is only present in the system when needed, which improves security.

If key operations need to be performed again after a Stop operation, then a Start or Reconstruct operation needs to be completed first, as explained in PUF Controller States.

60.5.5.7. Get Key Operation

The Get Key operation derives a key from the intrinsic PUF key and the provided context. The context also defines the length of the derived key. The key is provided to the defined destination interface. Depending on the state of the PUF controller, the allowed destinations may be limited as described in Context Specification for Key Operations. When an invalid destination is defined (all bits 0, or more than one bit 1), the command is rejected.

Bits 3:0 of the user_context_0 word of the context are passed to the Key Index register (PUF_KEY_INDEX). When the key is output to PUF_DOR, the value can be used to transfer the key to a specific process or function.

60.5.5.8. Wrap Generated Random Operation

The Wrap Generated Random operation wraps a random key into a key code (KC). The length, scope and user context of the random key are defined with a context as described in Context Specification for Key Operations. Key codes start with the provided context.

A key code is not sensitive (it does not reveal information on the corresponding key) and is linked to the activation code that was used during the last Enroll, Start or Reconstruct command.

60.5.5.9.Wrap Operation

The Wrap operation wraps a user-defined key into a key code. The length, scope and user context of the key are defined with a context as described in Context Specification for Key Operations. Key codes start with the provided context.

A key code is not sensitive (it does not reveal information on the corresponding key) and is linked to the activation code that was used during the last Enroll, Start or Reconstruct command.

60.5.5.10.Unwrap Operation

The Unwrap operation unwraps the key from a previously created key code. Depending on the state of PUF and the key scope that is included in the key code, the allowed destinations may be limited as described in Context Specification for Key Operations. When an invalid destination is defined (all bits 0, or more than one bit 1), the command is rejected.

Bits 3:0 of the user_context_0 word of the context, as provided during the Wrap or Wrap Generated Random operation, are passed to PUF_KEY_INDEX.

60.5.5.11.Generate Random Operation

The Generate Random operation outputs the requested amount of random data as specified in a provided context. The context is defined in Context Specification for Generate Random Operation.

60.5.5.12.Reseed Operation

With the Reseed operation, DRBG2 is reseeded with new entropy, and its reseed counter is reset.

New entropy always comes from DRBG1. In addition, external random entropy can be provided via PUF_DIR. The Data Source register (PUF_SRC) must be configured with 0x1.

The following table shows the number of bytes that must be provided per entropy source.

Table 60.4. Number of Bytes per Entropy Source for Reseed Operation

Entropy Source	Number of Bits	Number of Bytes
DRBG1	384	48
Entropy via PUF_DIR	1024	128

60.5.5.13.Test Memory Operation

With the Test Memory operation, the PUF SRAM is tested for defects. If the test fails, an error is indicated and PUF enters the Locked state.

60.5.5.14.Test PUF Operation

With the Test PUF operation, diagnostics about the PUF quality are collected and presented in a PUF score. When the operation is successful, the result is read in PUF_PSR. Details about PUF diagnostics can be found in PUF Diagnostics.

Note: The Test PUF operation collects the same diagnostic information as the Enroll operation.

60.5.5.15.Zeroize Operation

The Zeroize operation removes all sensitive data from the PUF controller and the PUF SRAM. This disables all commands until the next power cycle. The Zeroize command can be used when a security breach is detected. When the Zeroize operation is successful, the PUF controller goes to the Zeroized state. If the operation fails, the PUF controller enters the Locked state.

As part of the Zeroize operation, the DRBG1 and DRBG2 are uninstantiated.

Zeroize operation is performed by writing PUF_CR.ZEROIZE=1.

Zeroize takes precedence over all other activities of the PUF controller, except when BIST is running. This means it can also be started during initialization, or while another operation is active. Zeroize cancels the current operation.

Note: When Zeroize occurs during an operation, any ongoing data transfers via PUF_DIR or PUF_DOR are interrupted. Software must ensure that these events can be handled.

60.5.6. PUF Data Formats

60.5.6.1. Activation Code Format

The activation code includes a 64-bit header carrying a unique value checked by the PUF controller, a payload and a 32-bit checksum checked for transfer errors.

The overall length of the activation code is 996 bytes.

The header (big-endian format) is 0x0001_0307_2005_0500.

Note: The checksum can be used by software to check that the activation code is stored safely in the non-volatile memory area.

60.5.6.2. Key Code Format

The key code includes a 128-bit header carrying a unique value and the context provided during the wrap or Wrap Generated Random operation, a payload and a 32-bit checksum checked for transfer errors.

The header (big-endian format) is 0x37 followed by word 0 to 3 of the provided context (See Context Specification for Key Operations).

The length of the key code for the accepted key length is provided in the following table.

Table 60.5. Key Code Length Versus Key Length

Key Length (bits)	Key Code Length (bytes)
64	60
128	68
192	76
256	84
320	92
384	100
448	124
512	132
576	140
640	148
704	156
768	164
832	188
896	196
960	204
1024	212
2048	388
3072	548
4096	724

60.5.6.3. Context Specification for Key Operations

The context input for Key operations is used to specify the properties of a key, and enables the explicit differentiation of keys that have distinct purposes. Under no circumstance must the same context be used on the same device for keys with different properties or purposes.

For Get Key operations, the combination of the intrinsic PUF key and the context uniquely defines the key that is output. If the same context is used, the same key is produced.

For Wrap and Wrap Generated Random operations, the combination of the intrinsic PUF key and context uniquely determines which keys are used to protect the wrapped data. Each key code is still unique, even when the same data and context are used.

Table 60.6. Fields in the Context for Key Operations

Word Index	Bit Range	Value	Description
0	[31:24]	0	Reserved. Must always be 0.
	[23:16]	0x10	Context for key operations
	[15:13]	0	Reserved. Must always be 0.
	[12:0]	N x 64 (with N= 1..16), 2048, 3072, 4096	Length of the key in bits. The key length is not necessarily the same as the security strength.
1	[31:16]	0	Must always be 0.
	[15:10]	0	Reserved. Must always be 0.
	[9]	0	Reserved. Must always be 0.
	[8]	0, 1	Defines the allowed key destinations, when PUF is in the Started state: 0: Key cannot be available via PUF_DOR 1: Key can be available via PUF_DOR
	[7:2]	0	Reserved. Must always be 0.
	[1]	0	Reserved. Must always be 0.
	[0]	0, 1	Defines the allowed key destinations, when PUF is in the Enrolled state: 1: Key is available via PUF_DOR
2	[31:0]	Any	User context for key derivation; available bits are restricted by the value read in PUF_HW_RUC0.
3	[31:0]	Any	User context for key derivation; available bits are restricted by the value read in PUF_HW_RUC1.

60.5.6.4. Context Specification for Generate Random Operation

The context for Generate Random operations is used for internal random generation. The following table defines the fields in the context for Generate Random.

Table 60.7. Fields in the Context for Generate Random Operations

Word Index	Bit Range	Value	Description
0	[31:24]	0	Reserved. Must always be 0.
	[23:16]	0x10	Context for Generate Random operations
	[15:13]	0	Reserved. Must always be 0.
	[12:0]	N x 64 (with N= 1..16), 2048, 3072, 4096	Length of the requested data in bits

60.5.7. PUF Operation Restrictions

The PUF Hardware Settings Register provides information on current restrictions applying to operations.

60.5.8. PUF Error Handling

60.5.8.1. Introduction

The PUF controller generates errors and failures. Errors are events that indicate that an operation was not successful, but have no immediate impact on the secure operation of the PUF controller.

Failures are events that prevent the PUF controller from ensuring a secure continuation of operations, and put the PUF controller in Locked state.

60.5.8.2. Failures

If a failure (e.g. PUF SRAM access error) is detected, either during Initialization, Zeroization or other operations, the PUF controller reports a failure result code in the PUF Operation Result Register (PUF_ORR.RCODE > 0xFA) and enters the Locked state. When the PUF controller is in Locked state, no operations other than Zeroize can be performed. The software must determine if a Zeroize operation is required.

The following table shows the possible failure result codes for each operation type.

Table 60.8. Possible Failure Result Codes per Operation

Operation / Result Code	Description	Possible Action
All Operations		
All / FAILURE_SRAM	PUF SRAM access failed.	Try to repower the PUF or the product.
Initialization		
Initialization / FAILURE_DRBG_HEALTH	One or more of the DRNG health tests failed (hardware is out of order, or has intermittent failure).	Try to repower the PUF or the product.
Initialization / FAILURE_SETTINGS	An illegal combination of values is read in PUF_HW_SETTINGS.	None
Initialization / FAILURE_DRBG1_ENTROPY	The entropy test for DRBG 1 failed.	Try to repower the PUF or the product.
Initialization / FAILURE_DRBG2_ENTROPY	Activation of DRBG 2 failed (DRBG 1 cannot provide entropy).	Try to repower the PUF or the product.
Reseed		
Reseed / FAILURE_DRBG2_ENTROPY	Reseed of DRBG 2 failed (DRBG 1 cannot provide entropy).	Try to repower the PUF or the product.

60.5.8.3. Errors

Upon detecting an error during an operation, the PUF controller cancels the command, presents an ERROR result code (PUF_ORR.RCODE > 0 and < 0xFA) and goes back to the state it was in before the command was issued. After an error, new commands can be executed by the PUF controller.

If an error occurs, the software determines what next steps apply:

- The occurred error is acceptable—The error condition can be repaired and a retry can be done. For security reasons the number of acceptable (sequential) errors must be limited. If more than this number of errors occur, it can be considered an attack and the last error cannot be accepted.
- The occurred error is not acceptable—Depending on the type and severity of the error, multiple actions can be initiated by software, for example:
  - Send a Stop command to PUF. This removes sensitive data from PUF and software can then perform other actions (e.g. ensure that the correct activation code is available), after which a new Start or Reconstruct command can be run.
  - Zeroize PUF, send Zeroize commands to other peripherals that have such command, and zeroize parts of system memory and registers that contain sensitive data.
  - Switch off the product/device, or the parts of the device that contain sensitive data.

– Reset the product/device and check if the error remains.

The following table shows the possible error result codes for each operation type.

Table 60.9. Possible Error Result Codes per Operation

Operations / Result Code	Description	Possible Action
Enroll		
Enroll / ERR_PUF_QUALITY	PUF-dedicated SRAM quality verification fails.	Try to repower the PUF or the product.
Start		
Start / ERR_PRODUCT	The provided activation code (AC) is invalid.	Provide an AC that was created with this product/device.
Start / ERR_TRANSFER	The provided AC is corrupted.	• Verify the AC checksum. • Try sending the AC again (it might be a transient error). • Check that the AC buffer (if used) is not overwritten by another process.
Start / ERR_AUTH	Authentication of the provided AC failed.	• Verify with the AC header that the AC was created for this device. • Check that the AC buffer (if used) is not overwritten by another process. • Try to repower the PUF or the product/device.
Start / ERR_PRODUCT_PH2	The AC in the second phase is invalid.	Provide an AC that was created with this product/device.
Start / ERR_TRANSFER_PH2	The AC in the second phase is corrupted.	• Verify the AC checksum. • Try sending the AC again (it might be a transient error). • Check that the AC buffer (if used) is not overwritten by another process.
Start / ERR_AUTH_PH2	Authentication of the provided AC failed in the second phase.	Check that the AC buffer (if used) is not overwritten by another process.
Reconstruct		
Reconstruct / ERR_PRODUCT	The provided activation code (AC) is invalid.	Provide an AC that was created with this product/device.
Reconstruct / ERR_TRANSFER	The provided AC is corrupted.	• Verify the AC checksum. • Try sending the AC again (it might be a transient error). • Check that the AC buffer (if used) is not overwritten by another process.
Reconstruct / ERR_AUTH	Authentication of the provided AC failed.	• Verify with the AC header that the AC was created for this device. • Check that the AC buffer (if used) is not overwritten by another process. • Try to repower the PUF or the product/device.
Get Key		
Get Key / ERR_CONTEXT	An incorrect or unsupported context is provided.	Provide a valid context.
Get Key / ERR_DESTINATION	A key destination that was set is not allowed by the key scope in the provided context and the current PUF.	Provide a valid context.

Table 60.9. Possible Error Result Codes per Operation (continued)

Operations / Result Code	Description	Possible Action
Wrap Generated Random, Wrap		
Wrap Generated Random, Wrap / ERR_CONTEXT	An incorrect or unsupported context is provided.	Provide a valid context.
Unwrap		
Unwrap / ERR_PRODUCT	The provided key code (KC) is invalid.	Provide a KC with a valid context.
Unwrap / ERR_CONTEXT	The context in the key code (KC) header is incorrect.	Provide a KC with a valid context.
Unwrap / ERR_DESTINATION	A key destination that was set is not allowed by the key scope in the KC header and the current PUF state.	Provide a valid destination.
Unwrap / ERR_TRANSFER	The provided KC is corrupted.	• Verify the KC checksum. • Try sending the KC again (it might be a transient error). • Check that the KC buffer (if used) is not overwritten by another process.
Unwrap / ERR_AUTH	Authentication of the provided KC failed.	• Verify with the KC header that the AC was created for this device. • Ensure that the KC was created with the same AC as was used during the last Start or Reconstruct operation (this must be managed by the system software). • Check that the KC buffer (if used) is not overwritten by another process.
Generate Random		
Generate Random / ERR_CONTEXT	An incorrect or unsupported context is provided.	Provide a valid context.

60.5.9. PUF Diagnostics

To determine the PUF quality, the PUF controller performs a statistical analysis on various aspects of the PUF, like randomness and noise. This is done during the Test PUF, Enroll, Start and Reconstruct operations, and results in a PUF score with a value from 0 to 7 that can be read in PUF Score Register after the operation successfully completes. A lower PUF score indicates a better PUF quality.

If the operation is not successful (PUF_SR.ERROR=1), the PUF score is not presented, and the register contains 0xF instead.

The PUF scores returned after Enroll (or Test PUF) give an indication of the PUF quality, while the scores after Start (or Reconstruct) provide an indication of the PUF noise level. Therefore the scores can be different.

When a high PUF score is unexpected, the system level software must determine what actions, if any, are required.

60.5.10. PUF Built-in Tests

60.5.10.1. PUF SRAM Test

The PUF SRAM is tested using a basic non-destructive memory test that retains the SRAM PUF response. The test verifies that the SRAM can be read and written, and that the address and data buses work correctly.

The test is performed in the following cases:

- After every power cycle
- During the Test Memory operation

60.5.10.2.PUF Quality Test

The PUF quality test is done with the Test PUF operation (see Test PUF Operation). See PUF Diagnostics for details on PUF diagnostics.

60.5.10.3.PUF Built-In Self-Test (BIST)

The PUF controller BIST is controlled by the PUF_TEST register.

The BIST is started by writing PUF_TEST.BISTEN=1. Once BIST is started, it cannot be interrupted by other commands (including Zeroize), and it runs until it is finished. The only way to stop it sooner is a PUF reset (refer to the section “Special Function Registers (SFR)”).

The BIST can only be started when PUF_TEST.BISTALLOW=1, which is the case when the PUF controller is in Initialized or Stopped state.

When BISTEN=1 while BISTALLOW=0, the BIST does not start until the BIST is allowed.

After PUF reset, the PUF_TEST.BISTOK and BISTERR flags are both cleared.

When BISTEN=1, the PUF_TEST.BISTACTIVE and BISTRUN flags go to 1 and the BISTOK and BISTERR flags go to 0. When the BIST is finished, BISTACTIVE goes to 0, and both BISTOK and BISTERR go to 1. This proves that the result signals are not stuck at 1 or 0, and that the result is reliable.

The BIST result is only shown after BISTEN=0. Either BISTOK or BISTERR goes to 0, while the other signal stays at 1, depending on the BIST result. When both signals remain at 1, it is also an indication that an error has occurred.

After the BIST is finished and the result is shown, the PUF controller resets, sets PUF_TEST.BISTRUN=0, PUF_SR.BUSY=1, runs the Initialization operation and enters Initialized state. The BIST result remains available.

In case of a BIST error, software must determine what is the next applicable action (e.g. retry BIST to check if the error was transient, zeroize, etc.).

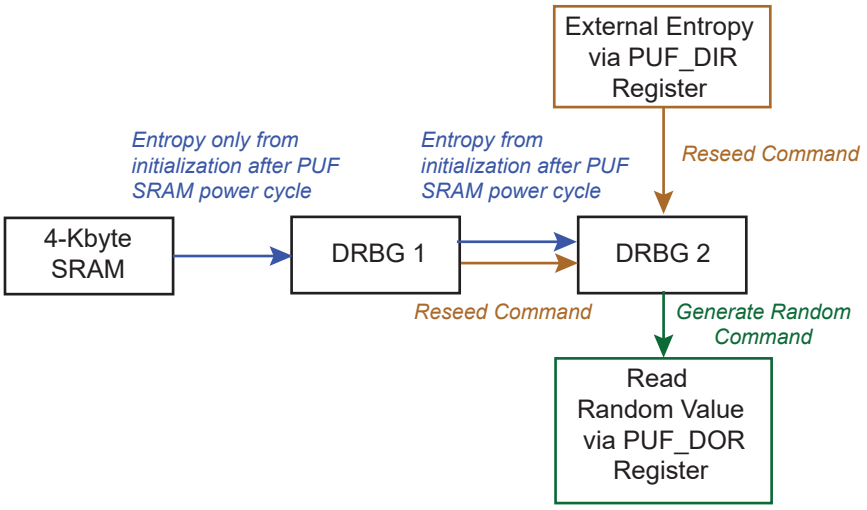
When the BIST operation is performed, a time-out procedure must be included in the calling software. This prevents infinite waiting times in case a PUF controller hardware error prevents the BIST from finishing. At the moment the software time-out occurs, the PUF controller must be reset to stop BIST.

60.5.11. PUF Deterministic Random Number Generator (DRNG)

60.5.11.1.Introduction

The deterministic random number generator is constructed with two deterministic random bit generators (DRBG), as shown in the following figure.

Figure 60.4. DRNG Structure



DRBG 1 uses the entropy from the PUF SRAM. This is done once per power cycle. During initialization (after every power cycle and after every reset), DRBG 2 uses entropy from DRBG 1.

During Reseed operation, DRBG 2 also uses entropy from DRBG 1 and in addition can accept entropy from external random entropy sources (1024 bits per source) via PUF_DIR. The Reseed operation can be started at any time that DRBG operations can be performed.

As defined by the NIST SP 800-90 recommendations, the maximum number of random requests between reseeds is 2^{48}. This is tracked internally with an internal reseed counter. Each DRBG has its own counter. For normal operations, DRBG 2 is used to generate random. DRBG 1 is only used during initialization.

The reseed counters track the number of executed internal (DRBG) Generate operations. Requests for random by means of the Generate Random operation can require multiple internal Generate operations.

Therefore, the reseed counter may be increased multiple times for each Generate Random operation.

Additionally, internal processes might also consume random data (e.g. to mask sensitive data or clear buffers) and thus increase the reseed counter.

The PUF controller guarantees that a started operation can always be completed without reaching the reseed counter limit. To this end, at least 256 ticks must be available on the reseed counter for any normal operation to be allowed.

If less than 256 ticks are available, i.e. the counter value is larger than 2^{48} - 257, then the flag PUF_ISR.RESEEDR=1. Only the Reseed and Zeroize operations can be performed once the counter reaches this limit and, depending on the PUF controller state, also BIST operations. Alternatively, the PUF controller or the entire device can also be reset or power-cycled to trigger a reinitialization.

Because BIST, reset and power-cycling transition the PUF controller to the Uninitialized state, DRBG 2 is reinitialized with a different value (using entropy from DRBG 1) and no reseed is required anymore. However, if external random entropy (PUF_DIR) is desired, a Reseed operation can be started after initialization has finished.

To make sure that no reseed will be required at an inconvenient moment, the PUF controller provides a warning when the internal reseed counter approaches the maximum value by 2^{32} or less.

This is indicated with 1 on the PUF_ISR.RESEEDW flag. Although about 4 * 10^9 random requests can still be made, the Reseed command must be activated at the first convenient moment.

Table 60.10. Reseed Counter Limits

Parameter	Value	Equivalent in Years at 1 Random Request per Second
Absolute maximum number of internal random requests	2^{48} - 2	8.9 * 10^6
Maximum value of reseed counter to begin an operation	2^{48} - 257	8.9 * 10^6
Warning level for number of random requests	2^{48} - 2^{32}	8.9 * 10^6 - 136

The limit to the number of requests for DRBG 1 is also 2^{48} - 257. In the unlikely event that its reseed counter reaches this value, the PUF controller enters the Locked state. This is persistent over resets. Only the Zeroize operation can be performed, or the PUF controller or the device can be power-cycled.

60.5.11.2.DRBG Mechanism Functions

The DRBG mechanism uses the HMAC_DRBG algorithms and are available for random generation as soon as the PUF controller is out of the Uninitialized or BIST state.

Seed material for DRBG 1:

- Entropy input (2048 bytes) from PUF SRAM, first compressed to 512 bits
- Nonce input: 8192 bits (1024 bytes) from PUF SRAM, first compressed to 256 bits
- Personalization string: 256 bits, product-specific value

Seed material for DRBG 2:

- Entropy input: 256 bits (32 bytes) from DRBG 1
- Nonce input: 128 bits (16 bytes) from DRBG 1
- Personalization string: 256 bits, product-specific value

The inputs of the DRBG2 reseed mechanism (DRNG1 cannot be reseeded) are:

- 384 bits (48 bytes) from DRBG 1
- up to 1024 bits (128 bytes) via PUF_DIR (when enabled)

Zeroize operation disables the DRNGs.

60.6. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PUF_CR	31:24	TESTPUF	TESTMEM						
		23:16								RESEED
		15:8	GENRAND						WRAP	WGENRAND
		7:0	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	ZEROIZE
0x04	PUF_ORR	31:24	LOPNUM[7:0]							
		23:16								
		15:8	RESEEDR	RESEEDW						
		7:0	RCODE[7:0]							
0x08	PUF_SR	31:24		RESEEDR	RESEEDW					
		23:16								
		15:8								
		7:0		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
0x0C	PUF_AR	31:24	TESTPUF	TESTMEM						
		23:16								RESEED
		15:8	GENRAND						WRAP	WGENRAND
		7:0	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	
0x10	PUF_IER	31:24								
		23:16								
		15:8								
		7:0								INTEN
0x14	PUF_IMR	31:24		RESEEDR	RESEEDW					
		23:16								
		15:8								
		7:0		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
0x18	PUF_ISR	31:24		RESEEDR	RESEEDW					
		23:16								
		15:8								
		7:0		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
0x1C ... 0x1F	Reserved									
0x20	PUF_DATA_DEST	31:24								
		23:16								
		15:8								
		7:0								DO
0x24	PUF_DATA_SRC	31:24								
		23:16								
		15:8								
		7:0								DI
0x28	PUF_KEY_INDEX	31:24								
		23:16								
		15:8								
		7:0					KI[3:0]			
0x2C ... 0x9F	Reserved									
0xA0	PUF_DIR	31:24	DI[31:24]							
		23:16	DI[23:16]							
		15:8	DI[15:8]							
		7:0	DI[7:0]							
0xA4 ... 0xA7	Reserved									
0xA8	PUF_DOR	31:24	DO[31:24]							
		23:16	DO[23:16]							
		15:8	DO[15:8]							
		7:0	DO[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xAC ... 0xBF	Reserved									
0xC0	PUF_MISC	31:24								
		23:16								
		15:8								
		7:0								DEND
0xC4 ... 0xCF	Reserved									
0xD0	PUF_IF_SR	31:24								
		23:16								
		15:8								
		7:0								BUSERR
0xD4 ... 0xD7	Reserved									
0xD8	PUF_TEST	31:24	BISTALLOW							
		23:16								
		15:8								
		7:0	BISTERR	BISTOK	BISTACTIVE	BISTRUN				BISTEN
0xDC	PUF_PSR	31:24								
		23:16								
		15:8								
		7:0					SCORE[3:0]			
0xE0	PUF_HW_RUC0	31:24	RUC31	RUC30	RUC29	RUC28	RUC27	RUC26	RUC25	RUC24
		23:16	RUC23	RUC22	RUC21	RUC20	RUC19	RUC18	RUC17	RUC16
		15:8	RUC15	RUC14	RUC13	RUC12	RUC11	RUC10	RUC9	RUC8
		7:0	RUC7	RUC6	RUC5	RUC4	RUC3	RUC2	RUC1	RUC0
0xE4	PUF_HW_RUC1	31:24	RUC31	RUC30	RUC29	RUC28	RUC27	RUC26	RUC25	RUC24
		23:16	RUC23	RUC22	RUC21	RUC20	RUC19	RUC18	RUC17	RUC16
		15:8	RUC15	RUC14	RUC13	RUC12	RUC11	RUC10	RUC9	RUC8
		7:0	RUC7	RUC6	RUC5	RUC4	RUC3	RUC2	RUC1	RUC0
0xE8 ... 0xEF	Reserved									
0xF0	PUF_HW_SETTINGS	31:24	TESTPUF	TESTMEM	MEMTEST		EXTSVIADIR		LABTESTSEL	LABTEST
		23:16								RESEED
		15:8	GENRAND						WRAP	WGENRAND
		7:0	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	

60.6.1. PUF Control Register

Name: PUF_CR
Offset: 0x000
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding operation.

Note: Only one command bit must be activated at a time, except ZEROIZE. Writing ZEROIZE=1 takes precedence over all other commands.

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

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

Bit	15	14	13	12	11	10	9	8
	GENRAND						WRAP	WGENRAND
Access	R/W						R/W	R/W
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	ZEROIZE
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 – TESTPUF Test PUF Operation

Bit 30 – TESTMEM Test Memory Operation

Bit 16 – RESEED Reseed Operation

Bit 15 – GENRAND Generate Random Operation

Bit 9 – WRAP Wrap Operation

Bit 8 – WGENRAND Wrap Generated Random Operation

Bit 7 – UNWRAP Unwrap Operation

Bit 6 – GETKEY Get Key Operation

Bit 5 – STOP Stop Operation

Bit 3 – RECO Reconstruct Operation

Bit 2 – START Start Operation

Bit 1 – ENROLL Enroll Operation

Bit 0 – ZEROIZE Zeroize Operation

60.6.2. PUF Operation Result Register

Name: PUF_ORR
Offset: 0x004
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	LOPNUM[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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RESEEDR	RESEEDW						
Access	R	R						
Reset	0	0						

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

Bits 31:24 – LOPNUM[7:0] Last Operation Number

Value	Name	Description
0x00	IID_PUF_LO_PROGRESS	An operation is in progress.
0x01	IID_PUF_LO_ENROLL	The last operation was ENROLL.
0x02	IID_PUF_LO_START	The last operation was START.
0x03	IID_PUF_LO_RECONSTRUCT	The last operation was RECONSTRUCT.
0x4	Reserved	Reserved
0x05	IID_PUF_LO_STOP	The last operation was STOP.
0x06	IID_PUF_LO_GET_KEY	The last operation was GET KEY.
0x07	IID_PUF_LO_UNWRAP	The last operation was UNWRAP.
0x08	IID_PUF_LO_WRAP_GEN_RND	The last operation was WRAP GENERATED RANDOM.
0x09	IID_PUF_LO_WRAP	The last operation was WRAP.
0x0A–0x0E	Reserved	Reserved
0x0F	IID_PUF_LO_GEN_RND	The last operation was GENERATED RANDOM.
0x10	IID_PUF_LO_RESEED	The last operation was RESEED.
0x11–0x1D	Reserved	Reserved
0x1E	IID_PUF_LO_TEST_MEMORY	The last operation was TESTMEMORY.
0x1F	IID_PUF_LO_TEST_PUF	The last operation was TESTPUF.
0x20	IID_PUF_LO_INITIALIZATION	The last operation was INITIALIZATION.
0x21–0x2E	Reserved	Reserved
0x2F	IID_PUF_LO_ZEROIZE	The last operation was ZEROIZE.

Bit 15 – RESEEDR Reseed Required

Value	Name	Description
0	NO_ACTION	No action required.

Value	Name	Description
1	REQUIRED_ACTION	The maximum number of DRBG2 requests has been done. A reseed must be performed before other operations can be done. This bit is the same as in PUF_SR.

Bit 14 – RESEEDW Reseed Warning

Value	Name	Description
0	NO_ACTION	No action required.
1	RECOMMENDED_ACTION	The DRBG reseed counter is close to its limit, it is recommended to reseed at the first suitable moment. This bit is the same as in PUF_SR.

Bits 7:0 – RCODE[7:0] Last Operation Result Code

Value	Name	Description
0x00	IID_PUF_OK	The last operation was successful or an operation is in progress.
0xF0	IID_PUF_ERR_PRODUCT	The provided activation code is not correct.
0xF1	IID_PUF_ERR_PRODUCT_PH2	The activation code in the second phase is not correct.
0xF2	IID_PUF_ERR_TRANSFER	The provided activation code is corrupted.
0xF3	IID_PUF_ERR_TRANSFER_PH2	The activation code in the second phase is corrupted.
0xF4	IID_PUF_ERR_AUTH	Authentication of the provided activation code failed.
0xF5	IID_PUF_ERR_AUTH_PH2	Authentication of the provided activation code failed in the second phase.
0xF6	IID_PUF_ERR_PUF_QUALITY	Dedicated Embedded Static RAM PUF quality verification fails.
0xF7	IID_PUF_ERR_CONTEXT	An incorrect or unsupported context is provided.
0xF8	IID_PUF_ERR_DESTINATION	A data destination that was set is not allowed according to other settings and the current PUF state.
0xF9-0xFA	Reserved	Reserved
0xFB	IID_PUF_FAILURE_DRBG2_ENTROPY	DRBG2 Reseed or Instantiation failed (DRBG1 cannot provide entropy).
0xFC	IID_PUF_FAILURE_DRBG1_ENTROPY	The entropy test for DRBG1 failed.
0xFD	IID_PUF_FAILURE_DRBG_HEALTH	One or more of the DRBG health tests failed.
0xFE	IID_PUF_FAILURE_SETTINGS	Initialization cannot finish due to illegal values in PUF_HW_SETTINGS.
0xFF	IID_PUF_FAILURE_SRAM	PUF SRAM access has failed.

60.6.3. PUF Status Register

Name: PUF_SR
Offset: 0x008
Reset: 0x00000001
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		RESEEDR	RESEEDW					
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	1

Bit 30 – RESEEDR Reseed Required

Value	Name	Description
0	NO_ACTION	No action required.
1	REQUIRED_ACTION	The maximum number of DRBG2 requests has been done. A reseed must be performed before other operations can be done. This bit is the same as in PUF_ORR.

Bit 29 – RESEEDW Reseed Warning

Value	Name	Description
0	NO_ACTION	No action required.
1	RECOMMENDED_ACTION	The reseed counter of the DRBG is close to its limit, it is recommended to reseed at the first suitable moment. This bit is the same as in PUF_ORR.

Bit 6 – DOREQ Data Out Request Status

Value	Description
0	No data-out transfer is in progress.
1	A data-out transfer is in progress following a request via PUF_DOR.

Bit 5 – DIREQ Data In Request Status

Value	Description
0	No data-in transfer is in progress.
1	A data-in transfer is in progress following a request via PUF_DIR.

Bit 4 – REJECTED Last Command Rejection Status (cleared by writing a 1)

Value	Description
0	No rejection event occurred since the last clearing access.
1	A rejection event occurred when the last command has been launched. Cleared by writing a 1.

Bit 3 – ZEROIZED PUF Zeroization in Progress

Value	Description
0	No zeroization is in progress.
1	A zeroization action is in progress or the PUF is in Locked state.

Bit 2 – ERROR Last Operation Failed

Value	Description
0	The last operation completed successfully.
1	The last operation failed.

Bit 1 – OK Last Operation Successfully Achieved

Value	Description
0	ERROR bit must be checked.
1	The last operation completed successfully.

Bit 0 – BUSY Operation in Progress

Value	Description
0	No operation is in progress. A new operation can be initiated.
1	An operation is in progress. No new operation must be launched.

60.6.4. PUF Allow Register

Name: PUF_AR
Offset: 0x00C
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding action is not allowed.

1: The corresponding action is allowed.

Bit	31	30	29	28	27	26	25	24
	TESTPUF	TESTMEM						
Access	R	R						
Reset	0	0						

Bit	23	22	21	20	19	18	17	16
								RESEED
Access								R
Reset								0

Bit	15	14	13	12	11	10	9	8
	GENRAND						WRAP	WGENRAND
Access	R						R	R
Reset	0						0	0

Bit	7	6	5	4	3	2	1	0
	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	
Access	R	R	R		R	R	R	
Reset	0	0	0		0	0	0	

Bit 31 – TESTPUF Test PUF Operation

Bit 30 – TESTMEM Test Memory Operation

Bit 16 – RESEED Reseed Operation

Bit 15 – GENRAND Generate Random Operation

Bit 9 – WRAP Wrap Operation

Bit 8 – WGENRAND Wrap Generated Random Operation

Bit 7 – UNWRAP Unwrap Operation

Bit 6 – GETKEY Get Key Operation

Bit 5 – STOP Stop Operation

Bit 3 – RECO Reconstruct Operation

Bit 2 – START Start Operation

Bit 1 – ENROLL Enroll Operation

60.6.5. PUF Interrupt Enable Register

Name: PUF_IER
Offset: 0x010
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
								INTEN
Access								R/W
Reset								0

Bit 0 – INTEN Interruption Enable

Value	Description
0	Disables the interruption.
1	Enables the PUF to trigger an interruption from the sources selected in PUF_IMR.

60.6.6. PUF Interrupt Mask Register

Name: PUF_IMR
Offset: 0x014
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: Disables the corresponding interrupt source.

1: Enables the corresponding interrupt source.

Bit	31	30	29	28	27	26	25	24
		RESEEDR	RESEEDW					
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
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 – RESEEDR Reseed Action Required Event

Bit 29 – RESEEDW Reseed Warning Event

Bit 6 – DOREQ Data Out Request Event

Bit 5 – DIREQ Data In Request Event

Bit 4 – REJECTED Last Activation Code Rejection Event

Bit 3 – ZEROIZED Zeroized Operation Completed Event

Bit 2 – ERROR Last Operation Fail Event

Bit 1 – OK Last Operation Successful Achievement Event

Bit 0 – BUSY Operation Start Event

60.6.7. PUF Interrupt Status Register

Name: PUF_ISR
Offset: 0x018
Reset: 0x00000000
Property: Read/Write

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt source has not triggered the interrupt line.

1: The corresponding interrupt source triggered the interrupt line.

Writing a 1 to a bit clears the corresponding bit.

Bit	31	30	29	28	27	26	25	24
		RESEEDR	RESEEDW					
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		DOREQ	DIREQ	REJECTED	ZEROIZED	ERROR	OK	BUSY
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 – RESEEDR Reseed Action Required (cleared by writing a 1)

Bit 29 – RESEEDW Reseed Warning (cleared by writing a 1)

Bit 6 – DOREQ Data Out Request (cleared by writing a 1)

Bit 5 – DIREQ Data In Request (cleared by writing a 1)

Bit 4 – REJECTED Last Activation Code Rejection (cleared by writing a 1)

Bit 3 – ZEROIZED Zeroized Operation Completed (cleared by writing a 1)

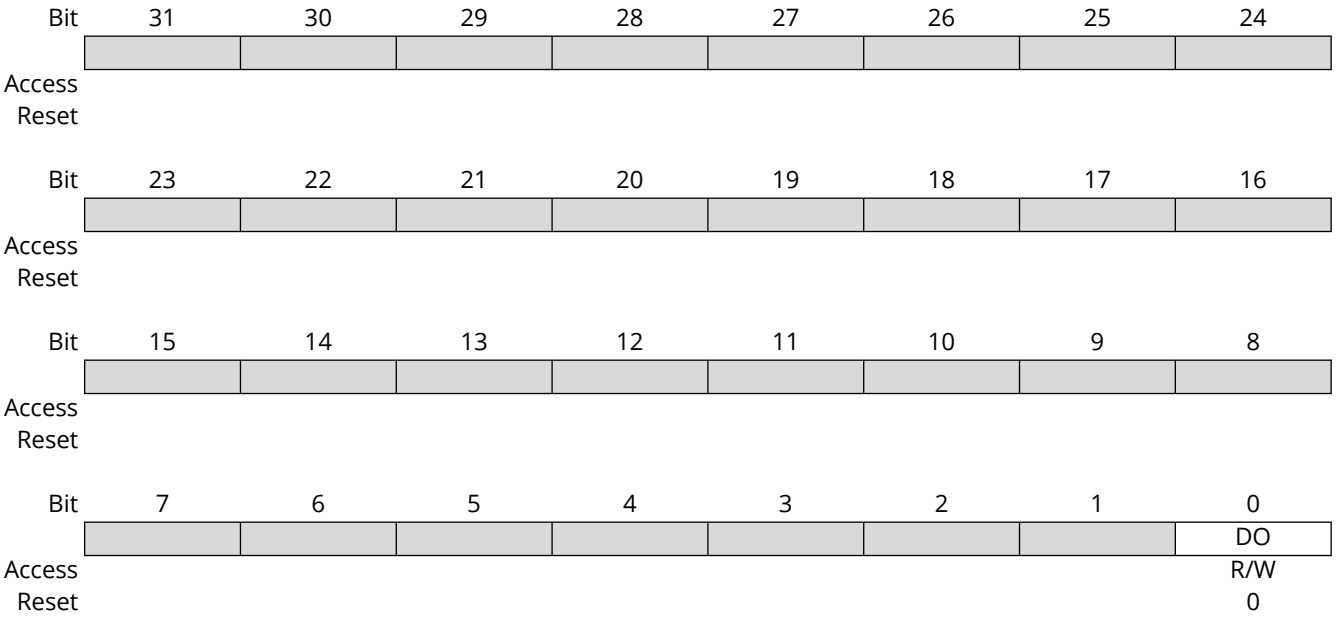
Bit 2 – ERROR Last Operation Fail (cleared by writing a 1)

Bit 1 – OK Last Operation Achievement (cleared by writing a 1)

Bit 0 – BUSY Operation in Progress (cleared by writing a 1)

60.6.8. PUF Data Destination Register

Name: PUF_DATA_DEST
Offset: 0x020
Reset: 0x00000000
Property: Read/Write



Bit 0 – DO Data Output on PUF_DOR

Value	Name	Description
0	DISABLED	PUF_DOR cannot be used to read data from PUF operations.
1	DO_VIA_DOR	PUF_DOR is enabled to read data from PUF operations.

60.6.9. PUF Data Source Register

Name: PUF_DATA_SRC
Offset: 0x024
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
								DI
Access								R/W
Reset								0

Bit 0 – DI Data Input Loaded from PUF_DIR

Value	Name	Description
0	DISABLED	PUF_DIR cannot be used to transfer data to PUF.
1	DI_VIA_DIR	PUF_DIR is enabled to load data to PUF.

60.6.10. PUF Key Index Register

Name: PUF_KEY_INDEX
Offset: 0x028
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
					KI[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bits 3:0 – KI[3:0] Key Index
Provides the Key Index of the key that is currently output.

60.6.11. PUF Data Input Register

Name: PUF_DIR
Offset: 0x0A0
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	DI[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	DI[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	DI[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	DI[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – DI[31:0] Data Input Value
This field must only be written when PUF_SR.DIREQ = 1.

60.6.12. PUF Data Output Register

Name: PUF_DOR
Offset: 0x0A8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DO[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
	DO[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
	DO[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
	DO[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DO[31:0] Data Output Value
This field must only be read when PUF_SR.DOREQ = 1.

60.6.13. PUF Miscellaneous Register

Name: PUF_MISC
Offset: 0x0C0
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
								DEND
Access								R/W
Reset								1

Bit 0 – DEND Data Endianness
This field must be configured once after power-up or system reset.

Value	Name	Description
0	LITTLE	Little-endian mode.
1	BIG	Big-endian mode. This is the default value.

60.6.14. PUF Interface Status Register

Name: PUF_IF_SR
Offset: 0x0D0
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
								BUSERR
Access								R/W
Reset								0

Bit 0 – BUSERR Bus Access Error (cleared by writing a 1)

Value	Name	Description
0	NONE	No incorrect access performed since the last write access (BUSERR=1).
1	ERROR	An incorrect access has been performed. Cleared by writing a 1.

60.6.15. PUF Test Register

Name: PUF_TEST
Offset: 0x0D8
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	BISTALLOW							
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
	BISTERR	BISTOK	BISTACTIVE	BISTRUN				BISTEN
Access	R/W	R/W	R/W	R/W				R/W
Reset	0	0	0	0				0

Bit 31 – BISTALLOW Built-In Self-Test Allowed

Value	Description
0	BIST Start (BISTEN) command is disabled and has no effect.
1	BIST Start command (BISTEN) is allowed.

Bit 7 – BISTERR Built-In Self-Test Error

Value	Description
0	No action
1	If BISTRUN=1 and BISTACTIVE=0, BIST has failed.

Bit 6 – BISTOK Built-In Self-Test Run Passed

Value	Description
0	No action
1	If BISTRUN=1 and BISTACTIVE=0, BIST has passed successfully.

Bit 5 – BISTACTIVE Built-In Self-Test Activity Status

Value	Description
0	BIST has completed if BISTRUN=1.
1	If BISTRUN=1, BIST is in progress.

Bit 4 – BISTRUN Built-In Self-Test Run Status

Value	Description
0	BIST is not active or has not started yet.

Value	Description
1	BIST is enabled.

Bit 0 – BISTEN Built-In Self-Test Enable

Value	Description
0	Disables the BIST, must be performed once BIST completion event occurred (BISTOK=1).
1	Starts the BIST.

60.6.16. PUF Score Register

Name: PUF_PSR
Offset: 0x0DC
Reset: 0x0000000F
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
					SCORE[3:0]			
Access					R	R	R	R
Reset					1	1	1	1

Bits 3:0 – SCORE[3:0] Last Test Score
Provides the PUF score obtained after launching one of the following operations: Test PUF, Enroll, Start or Reconstruct.

60.6.17. PUF Hardware Restrict User Context 0 Register

Name: PUF_HW_RUC0
Offset: 0x0E0
Reset: –
Property: Read-only

Values in this register are static and not managed by the PUF.

Bit	31	30	29	28	27	26	25	24
	RUC31	RUC30	RUC29	RUC28	RUC27	RUC26	RUC25	RUC24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RUC23	RUC22	RUC21	RUC20	RUC19	RUC18	RUC17	RUC16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RUC15	RUC14	RUC13	RUC12	RUC11	RUC10	RUC9	RUC8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RUC7	RUC6	RUC5	RUC4	RUC3	RUC2	RUC1	RUC0
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 – RUCx Restrict User Context Bit x

These bits are controlled by the SFR during the boot sequence and cannot be modified after the boot period.

Value	Description
0	This bit can be used in the user context 0 field.
1	This bit cannot be used in the user context 0 field.

60.6.18. PUF Hardware Restrict User Context 1 Register

Name: PUF_HW_RUC1
Offset: 0x0E4
Reset: –
Property: Read-only

Values in this register are static and not managed by the PUF.

Bit	31	30	29	28	27	26	25	24
	RUC31	RUC30	RUC29	RUC28	RUC27	RUC26	RUC25	RUC24
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	RUC23	RUC22	RUC21	RUC20	RUC19	RUC18	RUC17	RUC16
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	RUC15	RUC14	RUC13	RUC12	RUC11	RUC10	RUC9	RUC8
Access	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	RUC7	RUC6	RUC5	RUC4	RUC3	RUC2	RUC1	RUC0
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 – RUCx Restrict User Context Bit x

These bits are controlled by the SFR during the boot sequence and cannot be modified after the boot period.

Value	Description
0	This bit can be used in the user context 1 field.
1	This bit cannot be used in the user context 1 field.

60.6.19. PUF Hardware Settings Register

Name: PUF_HW_SETTINGS
Offset: 0x0F0
Reset: –
Property: Read-only

Values in this register are static and not managed by the PUF.
The following configuration values are valid for all listed bit names of this register:

0: The corresponding operation is allowed.

1: The corresponding operation is disabled.

Bit	31	30	29	28	27	26	25	24
	TESTPUF	TESTMEM	MEMTEST		EXTSVIADIR		LABTESTSEL	LABTEST
Access	R	R	R		R		R	R
Reset	–	–	–		–		–	–

Bit	23	22	21	20	19	18	17	16
								RESEED
Access								R
Reset								–

Bit	15	14	13	12	11	10	9	8
	GENRAND						WRAP	WGENRAND
Access	R						R	R
Reset	–						–	–

Bit	7	6	5	4	3	2	1	0
	UNWRAP	GETKEY	STOP		RECO	START	ENROLL	
Access	R	R	R		R	R	R	
Reset	–	–	–		–	–	–	

Bit 31 – TESTPUF Test PUF Operation

Bit 30 – TESTMEM Test Memory Operation

Bit 29 – MEMTEST Memory Tests Included in Initialization

Bit 27 – EXTSVIADIR External Entropy Required via PUF_DIR During Reseed Operation

Bit 25 – LABTESTSEL Selection of Lab Test Mode, when LABTEST=0

Bit 24 – LABTEST Initialization to Lab Test Mode

Bit 16 – RESEED Reseed Operation

Bit 15 – GENRAND Generate Random Operation

Bit 9 – WRAP Wrap Operation

Bit 8 – WGENRAND Wrap Generated Random Operation

Bit 7 – UNWRAP Unwrap Operation

Bit 6 – GETKEY Get Key Operation

Bit 5 – STOP Stop Operation

Bit 3 – RECO Reconstruct Operation

Bit 2 – START Start Operation

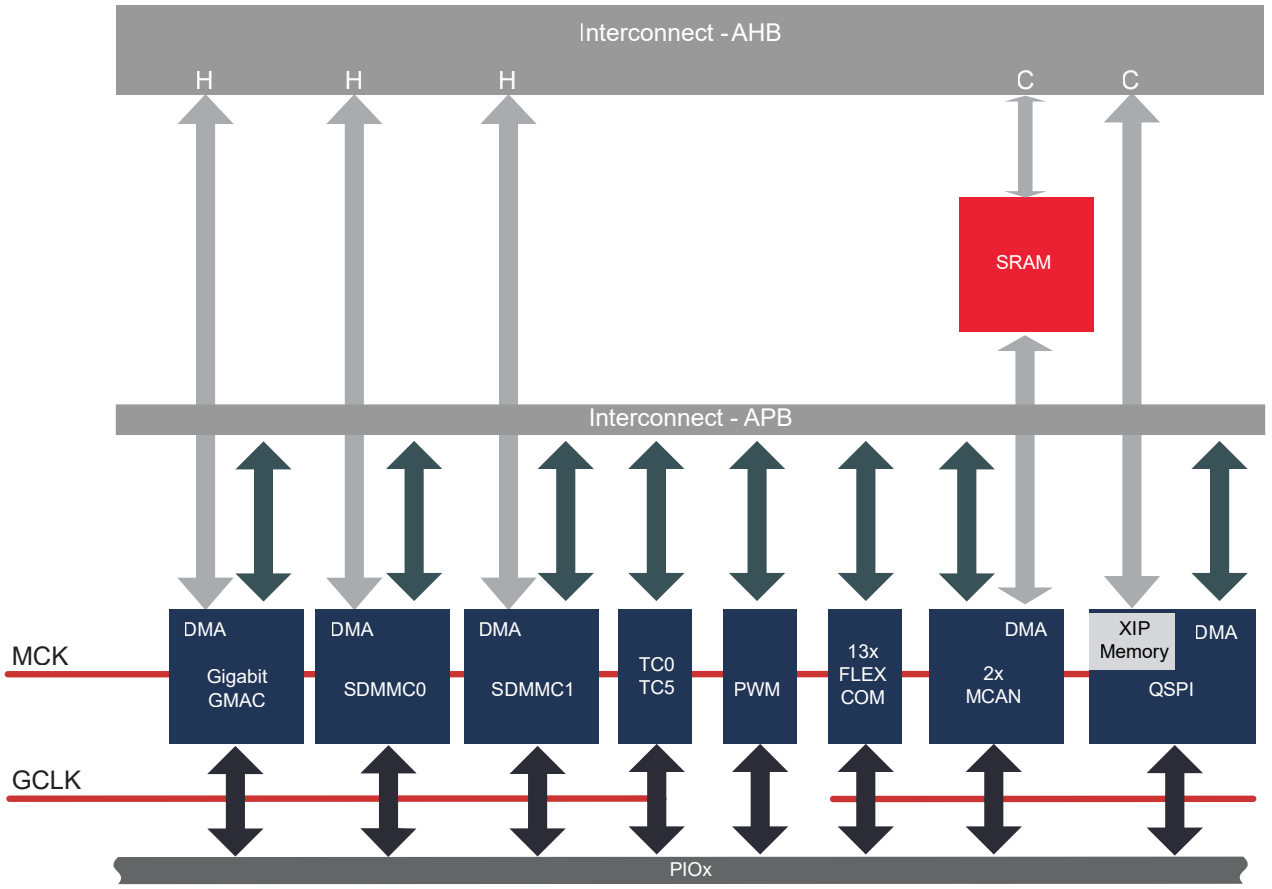
Bit 1 – ENROLL Enroll Operation

Connectivity Subsystem

61. Overview

61.1. Block Diagram

Figure 61.1. Connectivity Subsystem Block Diagram



61.2. Components

- 10/100/1000 Ethernet MAC (GMAC)
- 2x Secure Digital MultiMedia Card Controller (SDMMC0, 1)
- 6x Timer Counter
- Pulse Width Modulation Controller (PWM)
- 13x Flexible Serial Communication Controller (FLEXCOM)
- 2x Controller Area Network (MCAN)
- 1x Octal Serial Peripheral Interface (QSPI)

61.3. FLEXCOM Features

FLEXCOM	Subfunction	0	1	2	3	4	5	6	7	8	9	10	11	12
TWI	Normal/Fast (Client/Host)/FM+	x	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	x
	3 Client ADDR	x	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	x
	Sniffer	x	x	x	x	x	x	x	x	x	x	x	x	x
	FIFO size: 16 words	x	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	x
	Hardware handshaking/RS485	x	x	x	x	x	x	x	x	x	x	x	x	x
	ISO7816	x	x	x	x	x	x	–	–	–	–	–	–	–
	LIN	x	x	x	x	x	x	–	–	–	–	–	–	–
	LON	x	x	x	x	–	–	–	–	–	–	–	–	–
	IrDA	x	x	x	x	x	x	–	–	–	–	–	–	–
	Manchester	x	x	x	x	x	x	–	–	–	–	–	–	–
	FIFO size: 16 words	x	x	x	x	x	x	x	x	x	x	x	x	x
SPI	SPI implemented	x	x	x	x	x	x	–	–	–	–	–	–	–
	4 CS	–	–	–	–	x	x	–	–	–	–	–	–	–
	FIFO size: 16 words	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 part of the system controller.

All connectivity peripherals, 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. QSPI GCLK is a 2x clock that must be programmed to twice the QSCK frequency target.

61.4.1.2. GMAC Features

- GMAC supports RGMII and RMII.
- The media interface is configured using the GMAC_UR.MIM and GMAC_NCR.MIIONRGMII bits. Each interface has a dedicated clock configuration.

Figure 61.2. RMII MAC to PHY Connection Block Diagram

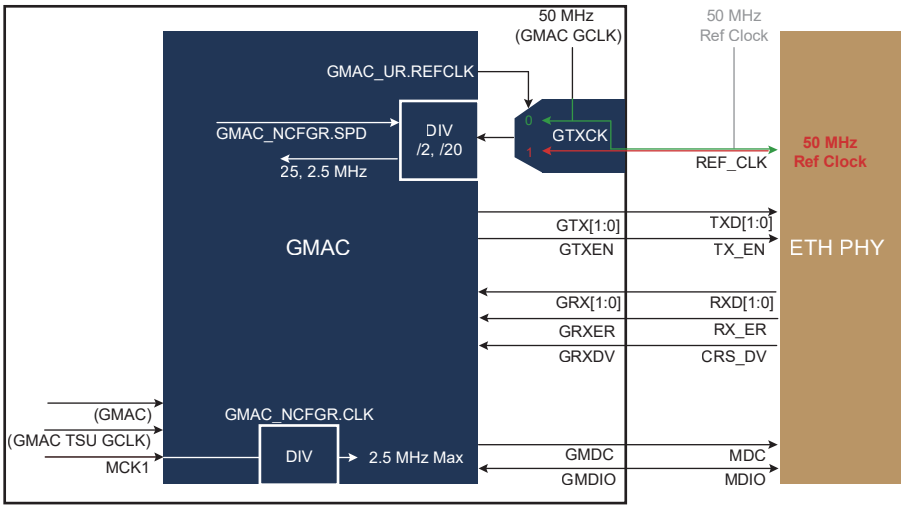
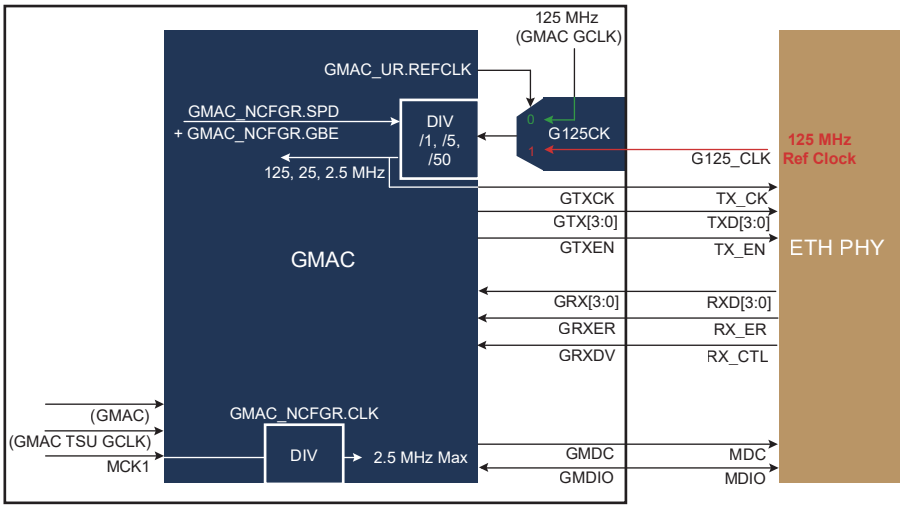


Figure 61.3. RGMII MAC to PHY Connection Block Diagram



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 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's MAC interface complies with the RGMII v1.3 specification that requires on the TX side a typical 2 ns data-to-clock time lapse 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 time lapse 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 have dual functions and are used as HOLD and WP in Dual and Serial I/O modes. 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

None.

62. 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.

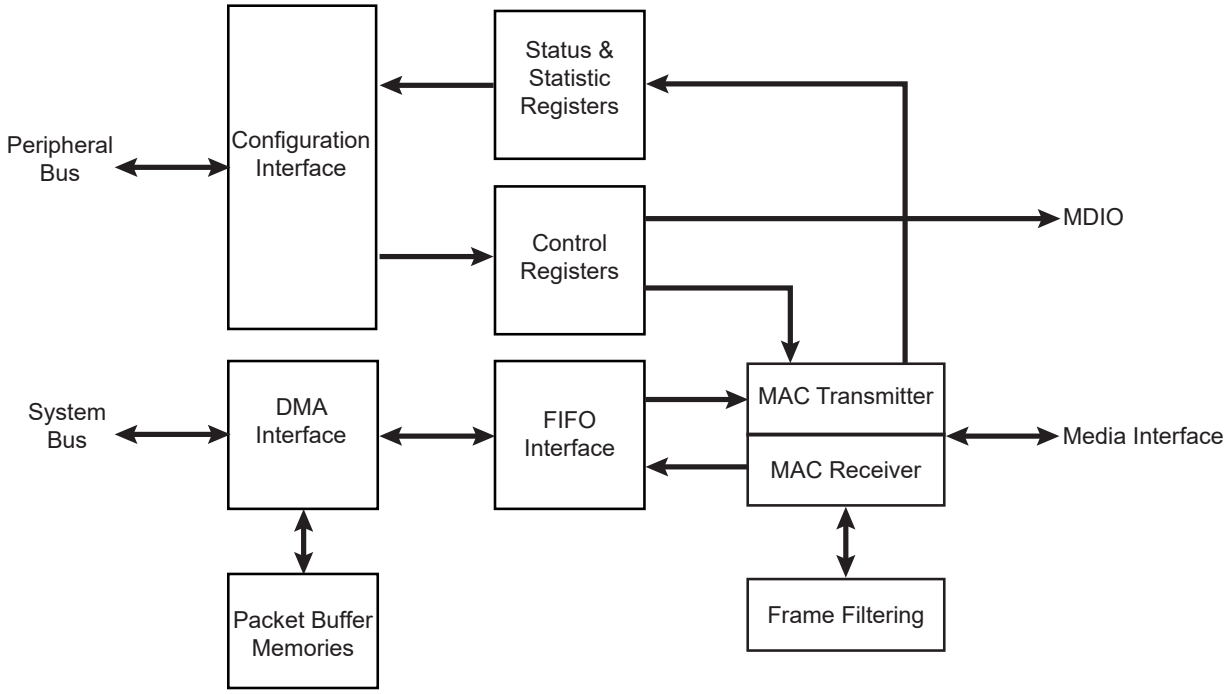
62.2. Embedded Characteristics

- Compatible with IEEE Standard 802.3
- 10, 100 and 1000 Mbps Operation
- Time-Sensitive Networking (TSN) IEEE 802.1Qbv Standard Support for Enhancement for Scheduled Traffic (EnST)
- Supports TSN IEEE 802.1CB Standard for Frame Elimination for Reliability
- Supports TSN IEEE 802.1Qci Standard for Receive (Ingress) Traffic Policy
- Supports TSN IEEE 802.3br Standard for Interspersing Express Traffic
- Supports TSN IEEE 802.1QBu Standard for Frame Preemption
- 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
- 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 (see Table 62.4 for queue-specific sizes)
- 4 Kbytes Transmit Memory and 4 Kbytes Receive Memory Dedicated to TSN Operations
- 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 16383 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



Note: For the TSN interspersing traffic express support, see the figure Interspersing Implementation Block Diagram.

62.4. Signal Interfaces

The GMAC includes the following signal interfaces:

- Media interface supports 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	RMII	RGMII
GTXCK	Transmit Clock or Reference Clock	REFCK	TXCK
G125CK	125 MHz Input Clock	Not Used	125 MHz Ref Clk
GTXEN	Transmit Enable	TXEN	TXCTL
GTX[3:0]	Transmit Data	TXD[1:0]	TXD[3:0]
GRXCK	Receive Clock	Not Used	RXCK
GRXDV	Receive Data Valid	CRSDV	RXCTL

Table 62.1. GMAC Connections to PHY in Different Modes (continued)

Signal Name	Function	RMII	RGMII
GRX[3:0]	Receive Data	RXD[1:0]	RXD[3:0]
GRXER	Receive Error	RXER	Not Used
GMDC	Management Data Clock	MDC	MDC
GMDIO	Management Data Input/Output	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 Duplex 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 16383 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.

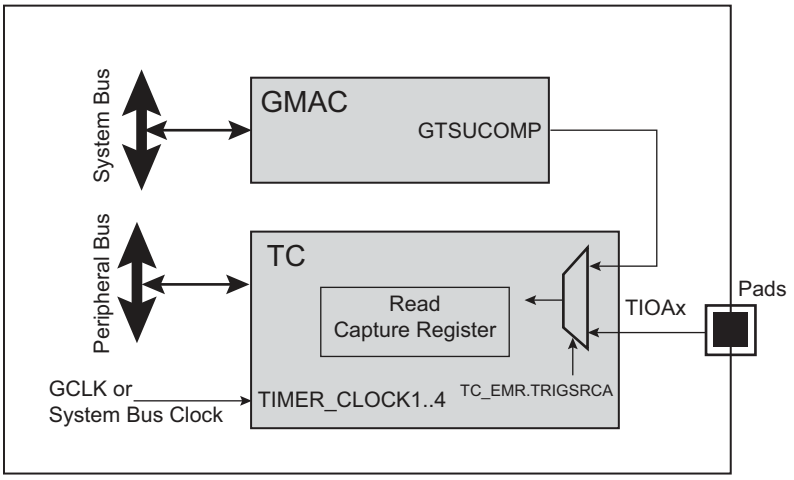
There are additional registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the bits of the seconds value and the 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 – see 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. See 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. 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	–
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

Table 62.2. Receive Buffer Descriptor Entry (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).

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.

Table 62.3. Transmit Buffer Descriptor Entry (continued)

Bit	Function
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
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.

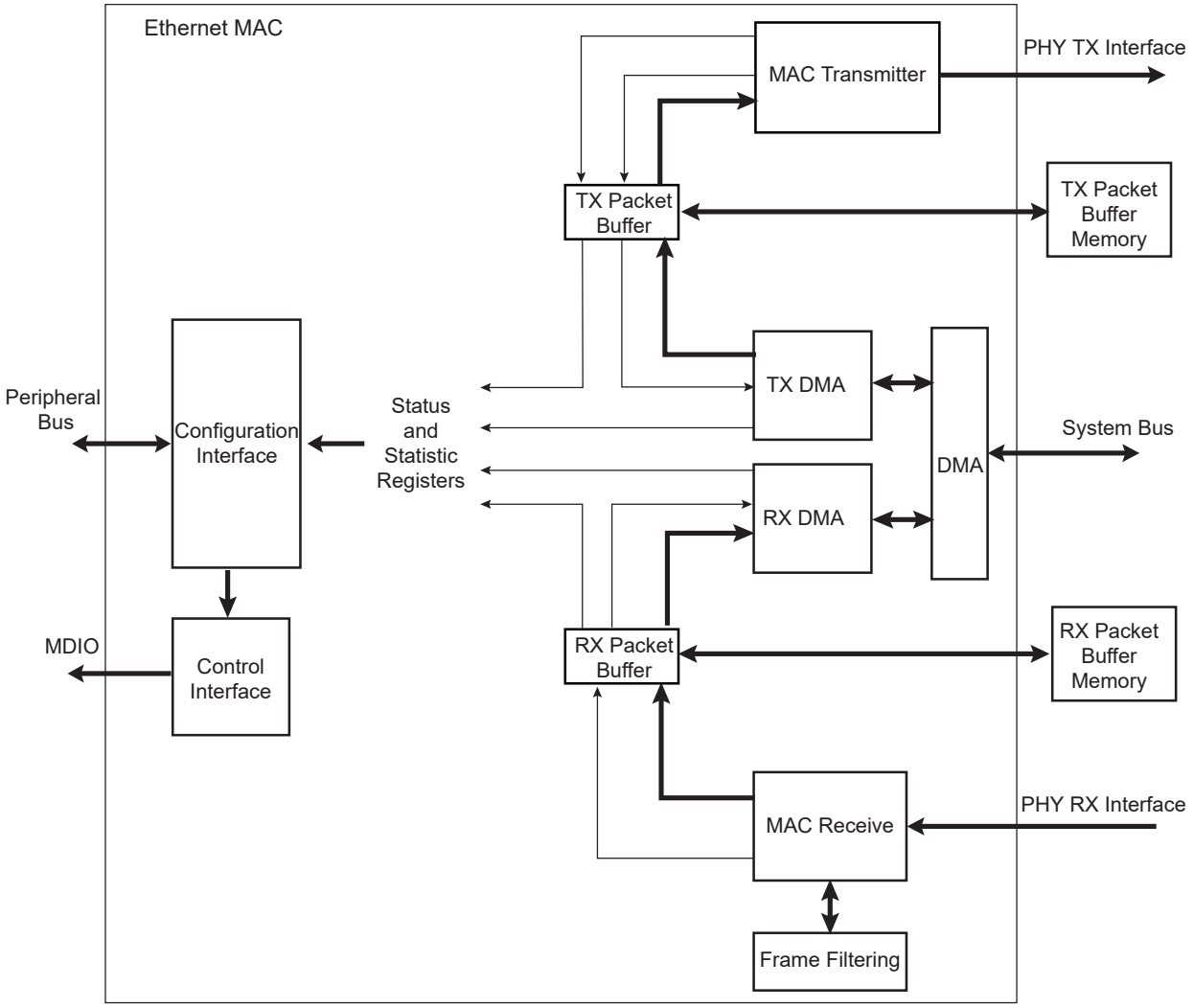
When the DMA is in Full Store and Forward mode, Full packet buffering provides the possibility to:

- 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 will extract data in 32-bit form. All subsequent processing prior to the final output is performed in bytes.

Transmit data can be output using the RMII/RGMII 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 100Mbps, maximum IdleSlope = (100Mbps / 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.

62.6.5.6. Enhancement for Scheduled Traffic (EnST)

IEEE 802.1Qbv is a TSN standard for “Enhancements for Scheduled Traffic” and specifies “time-aware queue-draining procedures” based on “timing derived from IEEE 802.1AS”. It adds transmission gates to the priority queues which allow low-priority queues to be shut down at specific times to allow higher-priority queues immediate access to the network at specific times.

For details, see Time-sensitive Networking Support (TSN).

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, see 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
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)	–

Table 62.6. Example of Sync Frame in 1588 Version 1 Format (continued)

Frame Segment	Value
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

Table 62.8. Example of Sync Frame in 1588 Version 2 (UDP/IPv4) Format (continued)

Frame Segment	Value
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
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)	–

Table 62.11. Example of Pdelay_Resp Frame in 1588 Version 2 (UDP/IPv6) Format (continued)

Frame Segment	Value
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:

- RMII – operates at 10/100 Mbps and uses GTX[1:0] and GRX[1:0].
- RGMIIv1.3 – operates at 10/100/1000 Mbps.

62.6.21. 10/100 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.

The GMAC can be configured to receive jumbo frames up to 16383 bytes by writing the RX Jumbo Frame Max Length register (GMAC_RJFML).

62.6.23. Time-sensitive Networking Support (TSN)

62.6.23.1.IEEE 802.1Qbv: Enhancement for Scheduled Traffic (EnST)

IEEE 802.1Qbv is a TSN standard for “Enhancements for Scheduled Traffic” and specifies “time-aware queue-draining procedures” based on “timing derived from IEEE 802.1AS”. It adds transmission gates to the eight priority queues which allow low priority queues to be shut down at specific times to allow higher priority queues immediate access to the network at specific times.

There are two main use cases for this:

- Firstly to allow guaranteed access for high priority low latency control frames (this is similar to Time Triggered Ethernet previously specified by the SAE in 2011 - AS6802)
- Secondly to allow periodic transmission of AVB traffic such as 1722 talker class A streams which need frames to be transmitted every 125 microseconds

GMAC supports IEEE 802.1Qbv by allowing time-aware control of individual transmit queues. GMAC has the ability to enable and disable transmission on a particular queue on a periodic basis with the on/off cycling starting at a specified TSU clock time.

For each transmit queue, configured up to a maximum of eight, there is a gate on timer and a gate off timer.

The on and off timer values are set in two 17-bit registers. These registers determine the “on” and “off” times for each queue in bytes. The actual “on” and “off” time will be a function of the number of bytes programmed in these registers and the speed of operation. 17 bits allow for a maximum value of about 1.05 milliseconds at 1G speed of operation.

For each transmit queue, configured up to a maximum of eight, there is a 32-bit register to control the start time for IEEE802.1Qbv queuing. The bottom 30 bits define the nanosecond value (as matched against the TSU timer) at which the queue will start and the top two bits the second value.

There is a 16-bit register that enables and disables IEEE 802.1Qbv traffic scheduling on each queue. The bottom 8 bits are for enabling each queue and the top 8 bits are for disabling. The disabling bits start at bit location 16. If the enable and disable bits are set simultaneously for a particular queue, the disable bit has priority. The disable bits are write-only. The enable bits are read/write with a read returning the queue’s status.

When enabled traffic scheduling starts for a particular queue at the time the TSU rolls over to the start time value in the relevant 32-bit register, the queue will then be on for the time in the “on” time register and off for the time in the “off” time register and the sequence will then repeat ignoring the time on the start time register. The functionality is similar to but not identical to the “Cycle Timer state machine” specified in the IEEE 802.1Qbv standard. The state machine in the IEEE 802.1Qbv standard allows complete freedom in setting a queue’s “on” and “off” times. The GMAC assumes that the “on” and “off” times will occur with a fixed period. If it is necessary to change the on/off sequencing, you need to write to the EnST control register to disable the queue and reprogram the “on”, “off”, and start times before restarting EnST on the queue.

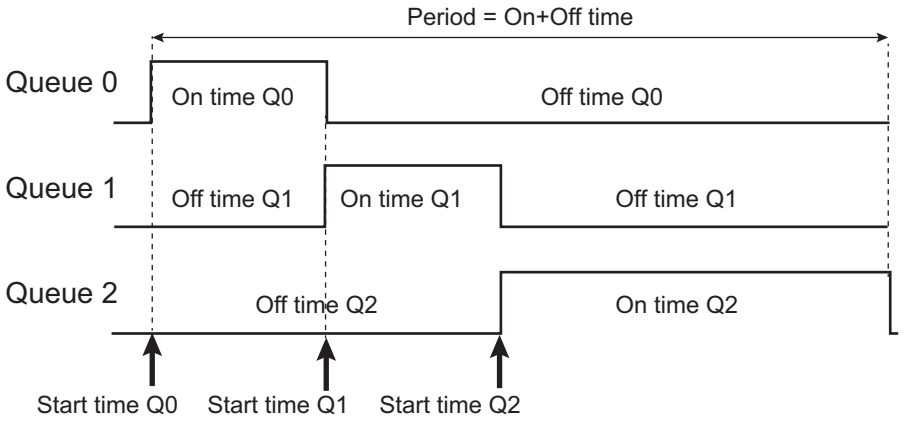
When traffic scheduling is disabled, the queue will no longer be gated and will be scheduled as normal.

If scheduling is enabled for a queue, transmission will not start if the frame to be transmitted will not complete in the remaining on-time available, that is, it is too long. If the frame is too long to ever be transmitted, then the frame will be dropped and an error indicated.

IEEE 802.1Qbv scheduling works in addition to other queue arbitration schemes such as Credit-Based Shaping (CBS) and priority queuing.

The EnST registers (GMAC_ENST_START_Q/ON_Q/OFF_Q) must be set so that the queues are not overlapping when they are open as shown in the following figure.

Figure 62.4. Example of Transmit Scheduling Configuration



To simplify implementation the on-time and the off-time are specified in bytes rather than nanoseconds.

The relation between the on-time (nanoseconds) and the on-time (bytes, the register of the module) is: on-time (bytes) = on-time (nanoseconds) divided by X, where X = 8 for 1 Gbit/s operation, 80 for 100 Mbit/s and 800 for 10 Mbits/s.

The maximum value of on-time and off-time and that can be set is 1,048,568 ns.

62.6.23.2.IEEE 802.1CB: Frame Elimination for Reliability

62.6.23.2.1.Introduction

IEEE 802.1CB “Frame Replication and Elimination for Reliability” is one of the TSN (Time Sensitive Networking) standards developed by the 802.1 working group. Using Frame Replication and Elimination for Reliability (FRER) within a network increases the probability that a given packet will be delivered using multi-paths through the network.

The GMAC supports a subset of this standard and provides the capability for stream identification and frame elimination but does not provide support for the replication of frames.

62.6.23.2.2.Stream Identification

The identification of IEEE 802.1CB streams relies on the use of the existing type 2 screener functionality. Each type 2 screener can be configured to inspect various fields of the receive frame and compared against values programmed into the type 2 screeners and compare registers.

For each implemented IEEE 802.1CB stream function, its associated control register contains the two fields ‘member_stream_1’ and ‘member_stream_2’:

- Each contains a value between 0x0 and 0xf representing the type 2 screener associated with this stream
- Each may point to the same or different type 2 screeners providing the option to eliminate packets which may have been sent across two different paths using different VLAN ID for example.

62.6.23.2.3.Support for FRER Redundancy Tag and Sequence Number Identification

To identify duplicate frames, a sequence number must be extracted from the incoming frame.

IEEE 802.1CB defines a new optional redundancy tag field that will immediately follow either the MAC source address field or the VLAN tag fields of the frame. This redundancy tag is identified by a new Ethertype with a value of 0xF1C1.

GMAC supports Draft 2.5 of the IEEE 802.1CB standard (see sub-clause 7.8). Thus the length of the redundancy tag is 6 bytes:

- The first two bytes are for the Ethertype (F1C1)
- The second two bytes are reserved and are set to zero
- The final two bytes are the 16-bit sequence number.

62.6.23.3.IEEE 802.3br: Interspersing Express Traffic

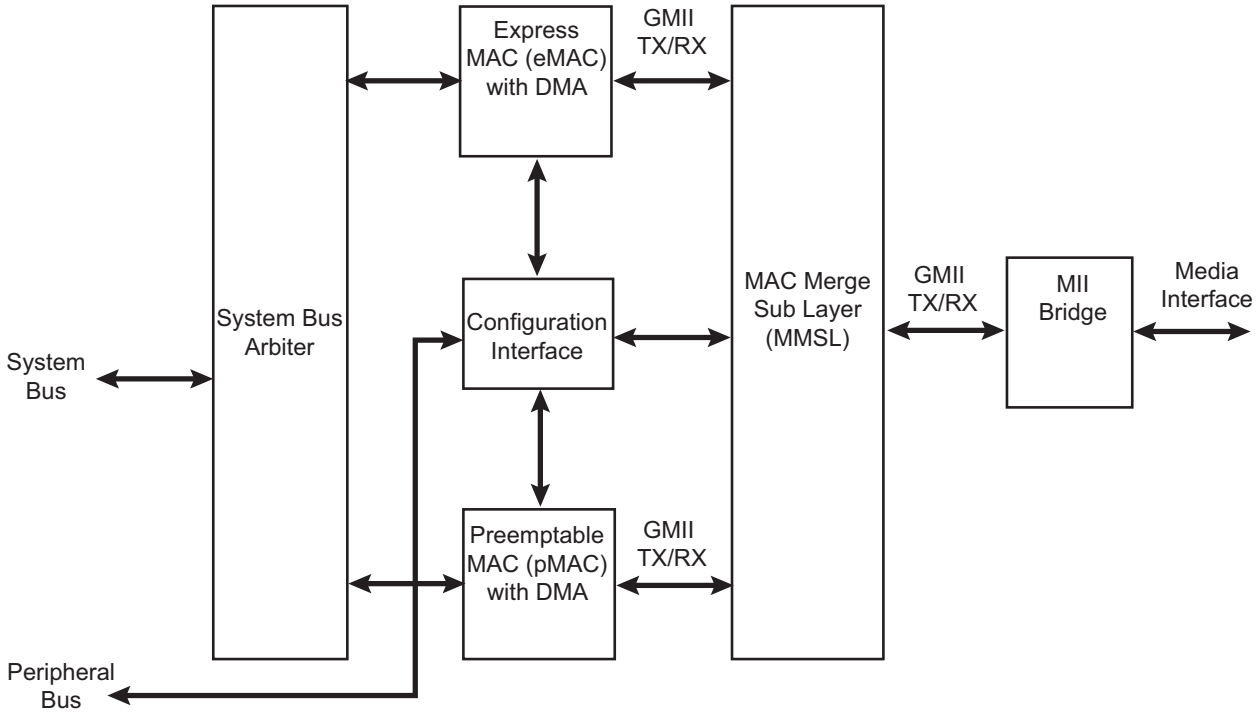
IEEE 802.3br "Interspersing Express Traffic" is one of the TSN standards, which defines a mechanism to transmit an express frame with a minimum delay at the expense of delaying completion of normal priority frames.

This standard is implemented by instantiating two separate MAC modules with related DMA, an MMSL (MAC Merge Sub Layer) and a system bus arbiter. One MAC is termed the express or eMAC and the other is a pre-emptable or pMAC.

The eMAC is designed to carry time sensitive traffic which must be delivered within a known time.

A conceptual view of this structure is shown in the following figure.

Figure 62.5. Interspersing Implementation Block Diagram



62.6.23.3.1.Frame Types

The use of this standard introduces a new format of frame, called an M-frame. M-frames are defined in the IEEE 802.3br standard as frames that have a special type of Start Frame Delimiter (SFD), called a Start mPacket Delimiter (SMD) and have a special type of CRC, called MCRC.

This new frame type is used by the MMSL to differentiate between the different types of traffic to determine whether:

- The frame is for the pMAC or eMAC
- An M-frame may consist of a full Ethernet frame or a fragment of an Ethernet frame

Frames to/from the eMAC are the same as regular Ethernet frames and when received will always contain a complete frame.

Frames to/from the pMAC use a different SFD and may be fragmented if a frame was required to be sent from the eMAC, in which case the pMAC frame is transmitted in between at least two pMAC frame fragments.

62.6.23.3.2.eMAC Configuration

The eMAC has a reduced set of features:

- The number of queues is 1.
- The number of stream functions is 2.
- The number of type 2 screeners and compare registers is 2 and 6 respectively.
- TSU is shared with the pMAC.
- RSC is not supported.

If the IEEE 802.3br functionality is to be used, then cut-thru and half-duplex and pause modes must not be enabled.

62.6.23.3.3.Transmit Interspersing Capability

On the transmit side, the interspersing capability is achieved by preempting M-frames from the pMAC in order to give the priority to Express frames from the eMAC.

If the pMAC is in the process of transmitting a frame and the eMAC requests to transmit, then the M-frame from the pMAC is halted with the M-frame transmission stopped and appended with a 32-bit MCRC before transmission of the express frame commences.

The additional support for IEEE 802.1Qbv also provides the capability to send time-scheduled express frames. This is supported by a set of programmable registers to define when the frame should be transmitted.

There are certain circumstances where a pMAC transmission will not be pre-empted, this is handled in accordance to the IEEE 802.3br standard as defined below:

- There are less than 64 M-frame bytes left to transmit.
- The M-frame transmission started at the moment when an Express packet requested the channel (the minimum number of bytes to be transmitted before preemption can occur is programmable in the field GMAC_MMSL_CR.ADD_FRAG_SIZE). In this case preemption will start after the minimum number of bytes of the M-frame have been transmitted.
- If the preemption function is not enabled.
- If the preemption function is enabled, the verification function is not disabled and the link partner could not complete the handshake protocol indicating that it cannot support the IEEE 802.3br standard.

62.6.23.3.4.Receive Interspersing Capability

The MMSL receiver processes the received M-frames to determine whether the frame is destined for the eMAC or the pMAC and routes them appropriately.

Frames destined for the eMAC look like standard Ethernet frames and are passed un-modified to the eMAC.

Frames destined for the pMAC contain a modified SFD and this new SFD is also used to identify frame fragments in order to reconstruct a complete Ethernet frame for the pMAC.

If a frame for the pMAC is fragmented due to an express frame, the MMSL receiver and the pMAC receive path are halted while the express frame is routed to the eMAC. Once a continuation fragment is received, the MMSL receiver and pMAC receive path resumes operation.

62.6.23.3.5.Verification Process

The verification process is defined by the IEEE 802.3br as a mechanism to validate the support of the 802.3br standard by both ends of the link.

If the preemption capability and the verification process are enabled, after reset a “verify frame” which consists of a special type of M-frame is transmitted. This frame, when received by the link partner should cause the link partner to transmit a ‘Respond Frame’.

The proper reception of this ‘Respond Frame’ completes the verification process and enables the preemption capability.

If no ‘Respond Frame’ is received within 10ms, the process will automatically be repeated another two times before the preemption capability is disabled. If this happens, the verification process may be restarted through the programming registers.

During the time in which the core is waiting for the Respond packet, the receive traffic will be routed to either the pMAC or the eMAC depending on the value of the 802.3br configuration registers.

62.6.23.4.IEEE 802.1Qci: Receive (Ingress) Traffic Policing

The TSN standards enable provisioning the resources in a network in such a way that high priority traffic is guaranteed to get through as long as it does not exceed its frame length and flow rate allocation.

802.1Qci is a policing mechanism that discards frames on ingress (i.e. received frames) if they exceed their allocated frame length or flow rate. The receive queue flush and screener type 2 maximum rate registers are provided for this purpose.

The receive queue flush registers starting at offset 0x0b00 allow frames to be discarded on a per-queue basis when the following events occur:

- All traffic received to a particular queue
- When a receive queue’s buffer descriptor is read with the used bit set
- When a queue’s dedicated memory utilization reaches a programmable depth (specified in 128-byte chunks)
- When a frame is received above a programmable length (specified in bytes)

The screener type 2 maximum rate registers starting at 0x0b40 allow frames to be discarded that exceed a certain flow rate.

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:

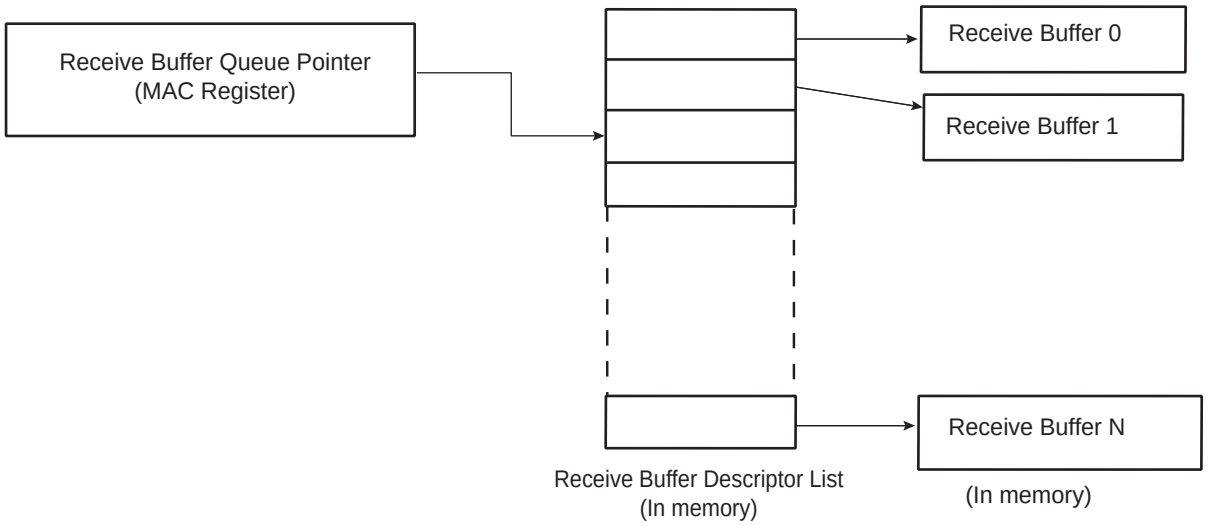
1. Write to Network Control register to disable transmit and receive circuits.
2. Write to Network Control register to change loop back mode.
3. Write to Network Control register to re-enable transmit or receive circuits.
   Note: These writes to the Network Control register cannot be combined in any 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 62.6. 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. Add an extra descriptor to the end of queue with its used bit set (bit 0 in word 0 set to 1). This last descriptor in the queue may also have its wrap bit set (bit 1 in word 0 set to 1) in addition to its used bit.When receive is enabled, at least one entry in the buffer descriptor ring needs its used bit set, so it is not sufficient to set the wrap bit of the last buffer in the queue without also setting its used bit. The GMAC can now prefetch receive descriptors and the used bit is used as an indication to the hardware that all available descriptors have been prefetched.
4. The GMAC can now read transmit data so fast that all data may be read in before it sets the used bit of the first buffer descriptor in the queue.
5. Write address of receive buffer descriptor list and control information to GMAC register receive buffer queue pointer
6. 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. Add an extra descriptor to the end of queue with its used bit set (bit 31 in word 1 set to 1). This last descriptor in the queue may also have its wrap bit set (bit 1 in word 0 set to 1) in addition to its used bit. When transmit is enabled, at least one entry in the buffer descriptor ring must have its used bit set. When the buffer descriptor ring is initialized for the first time, a used bit must be set before or at the buffer descriptor with the wrap bit. Once transmission halts due to reading a used bit, firmware can reuse the transmit buffers and clear the used bits before, including the one with the wrap bit, then restart transmission by writing TSTART in the Network Control register.
4. The GMAC can now read transmit data so fast that all data may be read in before it sets the used bit of the first buffer descriptor in the queue.
5. Write address of transmit buffer descriptor list and control information to GMAC register transmit buffer queue pointer.
6. 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			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								
		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		TXFOMAXB	RXFOMAXB	DDRP
		23:16	DRBS[7:0]							
		15:8			CRCERRREP	INFLASTEN	TXCOEN	TXPBMS	RXBMS[1:0]	
		7:0	ESPA	ESMA		FBLDO[4:0]				
0x14	GMAC_TSR	31:24								
		23:16								
		15:8						TXDMALCK	TXMACLCK	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			RXDMALCK	RXMACLCK	HNO	RXOVR	REC	BNA
0x24	GMAC_ISR	31:24	TXLCK	RXLCK	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x28	GMAC_IER	31:24	TXLCK	RXLCK	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
0x2C	GMAC_IDR	31:24	TXLCK	RXLCK	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	TXLCK	RXLCK	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[9:8]	
		7:0	RPB1ADR[7:0]							
0x48	GMAC_RJFML	31:24								
		23:16								
		15:8			FML[13:8]					
		7:0	FML[7:0]							
0x4C ... 0x5B	Reserved									
0x5C	GMAC_INTM	31:24								
		23:16	TXINTMOD[7:0]							
		15:8								
		7:0	RXINTMOD[7:0]							

Register Summary (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 ... 0x67	Reserved									
0x68	GMAC_LCKUP_CFGR	31:24	TXDMA_LCKUP_EN	TXMAC_LCKUP_EN	RXDMA_LCKUP_EN	RXMAC_LCKUP_EN	LCKUP_REC_EN	DMA_LOCKUP_TIME[10:8]		
		23:16	DMA_LOCKUP_TIME[7:0]							
		15:8	PRESCALER[15:8]							
		7:0	PRESCALER[7:0]							
0x6C	GMAC_LCKUP_TIME	31:24						TX_MAC_LOCKUP_TIME[10:8]		
		23:16	TX_MAC_LOCKUP_TIME[7:0]							
		15:8	RX_MAC_LOCKUP_TIME[15:8]							
		7:0	RX_MAC_LOCKUP_TIME[7:0]							
0x70	GMAC_TXDMA_LCKUP_CR	31:24								
		23:16								
		15:8								
		7:0			LCKUP_EN_Q5	LCKUP_EN_Q4	LCKUP_EN_Q3	LCKUP_EN_Q2	LCKUP_EN_Q1	LCKUP_EN_Q0
0x74 ... 0x7B	Reserved									
0x7C	GMAC_RX_WATERMARK	31:24	RX_LOW_WATERMARK[15:8]							
		23:16	RX_LOW_WATERMARK[7:0]							
		15:8	RX_HIGH_WATERMARK[15:8]							
		7:0	RX_HIGH_WATERMARK[7:0]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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									
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]							
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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 ... 0x026F	Reserved									
0x0270	GMAC_RXLPI	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
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 ... 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	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0410	GMAC_ISRPQ5	31:24								
		23:16								
		15:8					HRESP			
		7:0	TCOMP	TFC	RLEX			RXUBR	RCOMP	
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]							TXBQDIS
0x0444	GMAC_TBQBAPQ2	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							TXBQDIS
0x0448	GMAC_TBQBAPQ3	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							TXBQDIS
0x044C	GMAC_TBQBAPQ4	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							TXBQDIS
0x0450	GMAC_TBQBAPQ5	31:24	TXBQBA[29:22]							
		23:16	TXBQBA[21:14]							
		15:8	TXBQBA[13:6]							
		7:0	TXBQBA[5:0]							TXBQDIS
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]							RXBQDIS
0x0484	GMAC_RBQBAPQ2	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							RXBQDIS
0x0488	GMAC_RBQBAPQ3	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							RXBQDIS

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x048C	GMAC_RBQBAPQ4	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							RXBQDIS
0x0490	GMAC_RBQBAPQ5	31:24	RXBQBA[29:22]							
		23:16	RXBQBA[21:14]							
		15:8	RXBQBA[13:6]							
		7:0	RXBQBA[5:0]							RXBQDIS
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]					
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									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]	
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									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0590	GMAC_TQBWRL0	31:24	ALLOCQ3[7:0]							
		23:16	ALLOCQ2[7:0]							
		15:8	ALLOCQ1[7:0]							
		7:0	ALLOCQ0[7:0]							
0x0594 ... 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									
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	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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									
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									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]							
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]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]						
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]						
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
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]						
0x07C0 ... 0x07FF	Reserved									
0x0800	GMAC_ENST_START_Q0	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0804	GMAC_ENST_START_Q1	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							
0x0808	GMAC_ENST_START_Q2	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							
0x080C	GMAC_ENST_START_Q3	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							
0x0810	GMAC_ENST_START_Q4	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							
0x0814	GMAC_ENST_START_Q5	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							
0x0818 ... 0x081F	Reserved									
0x0820	GMAC_ENST_ON_Q0	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x0824	GMAC_ENST_ON_Q1	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x0828	GMAC_ENST_ON_Q2	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x082C	GMAC_ENST_ON_Q3	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x0830	GMAC_ENST_ON_Q4	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0834	GMAC_ENST_ON_Q5	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x0838 ... 0x083F	Reserved									
0x0840	GMAC_ENST_OFF_Q0	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x0844	GMAC_ENST_OFF_Q1	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x0848	GMAC_ENST_OFF_Q2	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x084C	GMAC_ENST_OFF_Q3	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x0850	GMAC_ENST_OFF_Q4	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x0854	GMAC_ENST_OFF_Q5	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x0858 ... 0x087F	Reserved									
0x0880	GMAC_ENST_CR	31:24								
		23:16								
		15:8								
		7:0			EN_Q5	EN_Q4	EN_Q3	EN_Q2	EN_Q1	EN_Q0
0x0884 ... 0x089F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x08A0	GMAC_FRER_TIMEOUT	31:24								
		23:16								
		15:8	TIMEOUT[15:8]							
		7:0	TIMEOUT[7:0]							
0x08A4	GMAC_FRER_REDTAG	31:24	STRIP_R_TAG	SIX_BYTE_TAG						
		23:16								
		15:8	RED_TAG[15:8]							
		7:0	RED_TAG[7:0]							
0x08A8 ... 0x08BF	Reserved									
0x08C0	GMAC_FRER_CTRL_A1	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x08C4	GMAC_FRER_CTRL_B1	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					
0x08C8	GMAC_FRER_STAT_A1	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							
0x08CC	GMAC_FRER_STAT_B1	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x08D0	GMAC_FRER_CTRL_A2	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x08D4	GMAC_FRER_CTRL_B2	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					
0x08D8	GMAC_FRER_STAT_A2	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x08DC	GMAC_FRER_STAT_B2	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x08E0	GMAC_FRER_CTRL_A3	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x08E4	GMAC_FRER_CTRL_B3	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					
0x08E8	GMAC_FRER_STAT_A3	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							
0x08EC	GMAC_FRER_STAT_B3	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x08F0	GMAC_FRER_CTRL_A4	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x08F4	GMAC_FRER_CTRL_B4	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					
0x08F8	GMAC_FRER_STAT_A4	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							
0x08FC	GMAC_FRER_STAT_B4	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x0900 ... 0x0AFF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0B00	GMAC_RX_FLUSH_Q0	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B04	GMAC_RX_FLUSH_Q1	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B08	GMAC_RX_FLUSH_Q2	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B0C	GMAC_RX_FLUSH_Q3	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B10	GMAC_RX_FLUSH_Q4	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B14	GMAC_RX_FLUSH_Q5	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x0B18 ... 0x0B3F	Reserved									
0x0B40	GMAC_SCR2_RATE_LIMIT0	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B44	GMAC_SCR2_RATE_LIMIT1	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0B48	GMAC_SCR2_RATE_LIMIT2	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B4C	GMAC_SCR2_RATE_LIMIT3	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B50	GMAC_SCR2_RATE_LIMIT4	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B54	GMAC_SCR2_RATE_LIMIT5	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B58	GMAC_SCR2_RATE_LIMIT6	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B5C	GMAC_SCR2_RATE_LIMIT7	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x0B60 ... 0x0B7F	Reserved									
0x0B80	GMAC_SCR2_RATE_STATUS	31:24								
		23:16								
		15:8								
		7:0	EXCESS_RATE_R7	EXCESS_RATE_R6	EXCESS_RATE_R5	EXCESS_RATE_R4	EXCESS_RATE_R3	EXCESS_RATE_R2	EXCESS_RATE_R1	EXCESS_RATE_R0
0x0B84 ... 0x0EFF	Reserved									
0x0F00	GMAC_MMSL_CR	31:24								
		23:16								
		15:8								
		7:0			ROUTE_RX_TO_PMAC	RESTART_VER	PRE_ENABLE	VERIFY_DISABLE	ADD_FRAG_SIZE[1:0]	
0x0F04	GMAC_MMSL_SR	31:24								
		23:16								
		15:8						SMD_ERR	FRER_COUNT_ERR	SMDC_ERR
		7:0	SMDS_ERR	RCV_V_ERR	RCV_R_ERR	VERIFY_STATUS[2:0]			RESPOND_STATUS	PRE_ACTIVE

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0F08	GMAC_MMSL_ESR	31:24								
		23:16	SMD_ERROR_COUNT[7:0]							
		15:8								
		7:0	ASS_ERROR_COUNT[7:0]							
0x0F0C	GMAC_MMSL_ASS_OK	31:24								
		23:16								ASS_OK_COUNT[16]
		15:8	ASS_OK_COUNT[15:8]							
		7:0	ASS_OK_COUNT[7:0]							
0x0F10	GMAC_MMSL_RXFRAG_CNT	31:24								
		23:16								FRAG_COUNT_RX[16]
		15:8	FRAG_COUNT_RX[15:8]							
		7:0	FRAG_COUNT_RX[7:0]							
0x0F14	GMAC_MMSL_TXFRAG_CNT	31:24								
		23:16								FRAG_COUNT_TX[16]
		15:8	FRAG_COUNT_TX[15:8]							
		7:0	FRAG_COUNT_TX[7:0]							
0x0F18	GMAC_MMSL_ISR	31:24								
		23:16								
		15:8								
		7:0			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
0x0F1C	GMAC_MMSL_IER	31:24								
		23:16								
		15:8								
		7:0			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
0x0F20	GMAC_MMSL_IDR	31:24								
		23:16								
		15:8								
		7:0			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
0x0F24	GMAC_MMSL_IMR	31:24								
		23:16								
		15:8								
		7:0			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
0x0F28 ... 0x0FFF	Reserved									
0x1000	GMAC_EMAC_NCR	31:24		IFGQAVCRED			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	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1004	GMAC_EMAC_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
0x1008	GMAC_EMAC_NSR	31:24								
		23:16								
		15:8								
		7:0	RXLPIS	PFCPAUSN				IDLE	MDIO	
0x100C ... 0x100F	Reserved									
0x1010	GMAC_EMAC_DCFGR	31:24			TXBD_EXTENDED	RXBD_EXTENDED		TXFOMAXB	RXFOMAXB	DDRP
		23:16	DRBS[7:0]							
		15:8			CRCERRREP	INFLASTEN	TXCOEN	TXPBMS	RXBMS[1:0]	
		7:0	ESPA	ESMA		FBLDO[4:0]				
0x1014	GMAC_EMAC_TSR	31:24								
		23:16								
		15:8						TXDMALCK	TXMACLCK	HRESP
		7:0	LCO		TXCOMP	TFC	TXGO	RLE	COL	UBR
0x1018	GMAC_EMAC_RBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							RXQDIS
0x101C	GMAC_EMAC_TBQB	31:24	ADDR[29:22]							
		23:16	ADDR[21:14]							
		15:8	ADDR[13:6]							
		7:0	ADDR[5:0]							TXQDIS
0x1020	GMAC_EMAC_RSR	31:24								
		23:16								
		15:8								
		7:0			RXDMALCK	RXMACLCK	HNO	RXOVR	REC	BNA
0x1024	GMAC_EMAC_ISR	31:24	TXLCK	RXLCK	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
0x1028	GMAC_EMAC_IER	31:24	TXLCK	RXLCK	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
0x102C	GMAC_EMAC_IDR	31:24	TXLCK	RXLCK	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1030	GMAC_EMAC_IMR	31:24	TXLCK	RXLCK	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
0x1034 ... 0x1037	Reserved									
0x1038	GMAC_EMAC_RPQ	31:24								
		23:16								
		15:8	RPQ[15:8]							
		7:0	RPQ[7:0]							
0x103C	GMAC_EMAC_TPQ	31:24	P1TPQ[15:8]							
		23:16	P1TPQ[7:0]							
		15:8	TPQ[15:8]							
		7:0	TPQ[7:0]							
0x1040	GMAC_EMAC_TPSF	31:24	ENTXP							
		23:16								
		15:8							TPB1ADR[9:8]	
		7:0	TPB1ADR[7:0]							
0x1044	GMAC_EMAC_RPSF	31:24	ENRXP							
		23:16								
		15:8							RPB1ADR[9:8]	
		7:0	RPB1ADR[7:0]							
0x1048	GMAC_EMAC_RJFML	31:24								
		23:16								
		15:8			FML[13:8]					
		7:0	FML[7:0]							
0x104C ... 0x1053	Reserved									
0x1054	GMAC_EMAC_AMP	31:24								
		23:16								USE_FROM
		15:8	AW2W_MAX_PIPELINE[7:0]							
		7:0	AR2R_MAX_PIPELINE[7:0]							
0x1058 ... 0x105B	Reserved									
0x105C	GMAC_EMAC_INTM	31:24								
		23:16	TXINTMOD[7:0]							
		15:8								
		7:0	RXINTMOD[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1060	GMAC_EMAC_SYSWT	31:24								
		23:16								
		15:8	SYSWKUPTIME[15:8]							
		7:0	SYSWKUPTIME[7:0]							
0x1064 ... 0x1067	Reserved									
0x1068	GMAC_EMAC_LCKUP_CFGR	31:24	TXDMA_LCKUP_EN	TXMAC_LCKUP_EN	RXDMA_LCKUP_EN	RXMAC_LCKUP_EN	LCKUP_REC_EN	DMA_LOCKUP_TIME[10:8]		
		23:16	DMA_LOCKUP_TIME[7:0]							
		15:8	PRESCALER[15:8]							
		7:0	PRESCALER[7:0]							
0x106C	GMAC_EMAC_LCKUP_TIME	31:24						TX_MAC_LOCKUP_TIME[10:8]		
		23:16	TX_MAC_LOCKUP_TIME[7:0]							
		15:8	RX_MAC_LOCKUP_TIME[15:8]							
		7:0	RX_MAC_LOCKUP_TIME[7:0]							
0x1070	GMAC_EMAC_TXDMA_LCKUP_CR	31:24								
		23:16								
		15:8								
		7:0			LCKUP_EN_Q5	LCKUP_EN_Q4	LCKUP_EN_Q3	LCKUP_EN_Q2	LCKUP_EN_Q1	LCKUP_EN_Q0
0x1074 ... 0x107B	Reserved									
0x107C	GMAC_EMAC_RX_WATERMARK	31:24	RX_LOW_WATERMARK[15:8]							
		23:16	RX_LOW_WATERMARK[7:0]							
		15:8	RX_HIGH_WATERMARK[15:8]							
		7:0	RX_HIGH_WATERMARK[7:0]							
0x1080	GMAC_EMAC_HRB	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1084	GMAC_EMAC_HRT	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1088	GMAC_EMAC_SAB0	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x108C	GMAC_EMAC_SAT0	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1090	GMAC_EMAC_SAB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1094	GMAC_EMAC_SAT1	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1098	GMAC_EMAC_SAB2	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x109C	GMAC_EMAC_SAT2	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10A0	GMAC_EMAC_SAB3	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10A4	GMAC_EMAC_SAT3	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10A8	GMAC_EMAC_TIDM1	31:24	ENID1							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10AC	GMAC_EMAC_TIDM2	31:24	ENID2							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B0	GMAC_EMAC_TIDM3	31:24	ENID3							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B4	GMAC_EMAC_TIDM4	31:24	ENID4							
		23:16								
		15:8	TID[15:8]							
		7:0	TID[7:0]							
0x10B8	GMAC_EMAC_WOL	31:24								
		23:16					MTI	SA1	ARP	MAG
		15:8	IP[15:8]							
		7:0	IP[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10BC	GMAC_EMAC_IPGS	31:24								
		23:16								
		15:8	FL[15:8]							
		7:0	FL[7:0]							
0x10C0	GMAC_EMAC_SVLAN	31:24	ESVLAN							
		23:16								
		15:8	VLAN_TYPE[15:8]							
		7:0	VLAN_TYPE[7:0]							
0x10C4	GMAC_EMAC_TPFCP	31:24								
		23:16								
		15:8	PQ[7:0]							
		7:0	PEV[7:0]							
0x10C8	GMAC_EMAC_SAMB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10CC	GMAC_EMAC_SAMT1	31:24								
		23:16								
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x10D0	GMAC_EMAC_AMRX	31:24	MSBADDR[3:0]							
		23:16								
		15:8								
		7:0					MSBADDRMSK[3:0]			
0x10D4	GMAC_EMAC_RXUDAR	31:24	RXUDA[31:24]							
		23:16	RXUDA[23:16]							
		15:8	RXUDA[15:8]							
		7:0	RXUDA[7:0]							
0x10D8	GMAC_EMAC_TXUDAR	31:24	TXUDA[31:24]							
		23:16	TXUDA[23:16]							
		15:8	TXUDA[15:8]							
		7:0	TXUDA[7:0]							
0x10DC	GMAC_EMAC_NSC	31:24								
		23:16			NANOSEC[21:16]					
		15:8	NANOSEC[15:8]							
		7:0	NANOSEC[7:0]							
0x10E0	GMAC_EMAC_SCL	31:24	SEC[31:24]							
		23:16	SEC[23:16]							
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							
0x10E4	GMAC_EMAC_SCH	31:24								
		23:16								
		15:8	SEC[15:8]							
		7:0	SEC[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x10E8	GMAC_EMAC_EFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10EC	GMAC_EMAC_EFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F0	GMAC_EMAC_PEFTSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F4	GMAC_EMAC_PEFRSH	31:24								
		23:16								
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x10F8 ... 0x10FF	Reserved									
0x1100	GMAC_EMAC_OTLO	31:24	TXO[31:24]							
		23:16	TXO[23:16]							
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x1104	GMAC_EMAC_OTHI	31:24								
		23:16								
		15:8	TXO[15:8]							
		7:0	TXO[7:0]							
0x1108	GMAC_EMAC_FT	31:24	FTX[31:24]							
		23:16	FTX[23:16]							
		15:8	FTX[15:8]							
		7:0	FTX[7:0]							
0x110C	GMAC_EMAC_BCFT	31:24	BFTX[31:24]							
		23:16	BFTX[23:16]							
		15:8	BFTX[15:8]							
		7:0	BFTX[7:0]							
0x1110	GMAC_EMAC_MFT	31:24	MFTX[31:24]							
		23:16	MFTX[23:16]							
		15:8	MFTX[15:8]							
		7:0	MFTX[7:0]							
0x1114	GMAC_EMAC_PFT	31:24								
		23:16								
		15:8	PFTX[15:8]							
		7:0	PFTX[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1118	GMAC_EMAC_BFT64	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x111C	GMAC_EMAC_TBFT127	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1120	GMAC_EMAC_TBFT255	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1124	GMAC_EMAC_TBFT511	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1128	GMAC_EMAC_TBFT1023	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x112C	GMAC_EMAC_TBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1130	GMAC_EMAC_GTBFT1518	31:24	NFTX[31:24]							
		23:16	NFTX[23:16]							
		15:8	NFTX[15:8]							
		7:0	NFTX[7:0]							
0x1134	GMAC_EMAC_TUR	31:24								
		23:16								
		15:8							TXUNR[9:8]	
		7:0	TXUNR[7:0]							
0x1138	GMAC_EMAC_SCF	31:24								
		23:16							SCOL[17:16]	
		15:8	SCOL[15:8]							
		7:0	SCOL[7:0]							
0x113C	GMAC_EMAC_MCF	31:24								
		23:16							MCOL[17:16]	
		15:8	MCOL[15:8]							
		7:0	MCOL[7:0]							
0x1140	GMAC_EMAC_EC	31:24								
		23:16								
		15:8							XCOL[9:8]	
		7:0	XCOL[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1144	GMAC_EMAC_LC	31:24								
		23:16								
		15:8							LCOL[9:8]	
		7:0	LCOL[7:0]							
0x1148	GMAC_EMAC_DTF	31:24								
		23:16							DEFT[17:16]	
		15:8	DEFT[15:8]							
		7:0	DEFT[7:0]							
0x114C	GMAC_EMAC_CSE	31:24								
		23:16								
		15:8							CSR[9:8]	
		7:0	CSR[7:0]							
0x1150	GMAC_EMAC_ORLO	31:24	RXO[31:24]							
		23:16	RXO[23:16]							
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							
0x1154	GMAC_EMAC_ORHI	31:24								
		23:16								
		15:8	RXO[15:8]							
		7:0	RXO[7:0]							
0x1158	GMAC_EMAC_FR	31:24	FRX[31:24]							
		23:16	FRX[23:16]							
		15:8	FRX[15:8]							
		7:0	FRX[7:0]							
0x115C	GMAC_EMAC_BCFR	31:24	BFRX[31:24]							
		23:16	BFRX[23:16]							
		15:8	BFRX[15:8]							
		7:0	BFRX[7:0]							
0x1160	GMAC_EMAC_MFR	31:24	MFRX[31:24]							
		23:16	MFRX[23:16]							
		15:8	MFRX[15:8]							
		7:0	MFRX[7:0]							
0x1164	GMAC_EMAC_PFR	31:24								
		23:16								
		15:8	PFRX[15:8]							
		7:0	PFRX[7:0]							
0x1168	GMAC_EMAC_BFR64	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x116C	GMAC_EMAC_TBFR127	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1170	GMAC_EMAC_TBFR255	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1174	GMAC_EMAC_TBFR511	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1178	GMAC_EMAC_TBFR1023	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x117C	GMAC_EMAC_TBFR1518	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1180	GMAC_EMAC_TMXBFR	31:24	NFRX[31:24]							
		23:16	NFRX[23:16]							
		15:8	NFRX[15:8]							
		7:0	NFRX[7:0]							
0x1184	GMAC_EMAC_UFR	31:24								
		23:16								
		15:8							UFRX[9:8]	
		7:0	UFRX[7:0]							
0x1188	GMAC_EMAC_OFR	31:24								
		23:16								
		15:8							OFRX[9:8]	
		7:0	OFRX[7:0]							
0x118C	GMAC_EMAC_JR	31:24								
		23:16								
		15:8							JRX[9:8]	
		7:0	JRX[7:0]							
0x1190	GMAC_EMAC_FCSE	31:24								
		23:16								
		15:8							FCKR[9:8]	
		7:0	FCKR[7:0]							
0x1194	GMAC_EMAC_LFFE	31:24								
		23:16								
		15:8							LFER[9:8]	
		7:0	LFER[7:0]							
0x1198	GMAC_EMAC_RSE	31:24								
		23:16								
		15:8							RXSE[9:8]	
		7:0	RXSE[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x119C	GMAC_EMAC_AE	31:24								
		23:16								
		15:8								AER[9:8]
		7:0	AER[7:0]							
0x11A0	GMAC_EMAC_RRE	31:24								
		23:16								RXRER[17:16]
		15:8	RXRER[15:8]							
		7:0	RXRER[7:0]							
0x11A4	GMAC_EMAC_ROE	31:24								
		23:16								
		15:8								RXOVR[9:8]
		7:0	RXOVR[7:0]							
0x11A8	GMAC_EMAC_IHCE	31:24								
		23:16								
		15:8								
		7:0	HCKER[7:0]							
0x11AC	GMAC_EMAC_TCE	31:24								
		23:16								
		15:8								
		7:0	TCKER[7:0]							
0x11B0	GMAC_EMAC_UCE	31:24								
		23:16								
		15:8								
		7:0	UCKER[7:0]							
0x11B4	GMAC_EMAC_FLRXPCR	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x11B8 ... 0x11BB	Reserved									
0x11BC	GMAC_EMAC_TISUBN	31:24	LSBTIR[7:0]							
		23:16								
		15:8	MSBTIR[15:8]							
		7:0	MSBTIR[7:0]							
0x11C0	GMAC_EMAC_TSH	31:24								
		23:16								
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x11C4 ... 0x11CF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x11D0	GMAC_EMAC_TSL	31:24	TCS[31:24]							
		23:16	TCS[23:16]							
		15:8	TCS[15:8]							
		7:0	TCS[7:0]							
0x11D4	GMAC_EMAC_TN	31:24			TNS[29:24]					
		23:16	TNS[23:16]							
		15:8	TNS[15:8]							
		7:0	TNS[7:0]							
0x11D8	GMAC_EMAC_TA	31:24	ADJ		ITDT[29:24]					
		23:16	ITDT[23:16]							
		15:8	ITDT[15:8]							
		7:0	ITDT[7:0]							
0x11DC	GMAC_EMAC_TI	31:24								
		23:16	NIT[7:0]							
		15:8	ACNS[7:0]							
		7:0	CNS[7:0]							
0x11E0	GMAC_EMAC_EFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11E4	GMAC_EMAC_EFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11E8	GMAC_EMAC_EFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11EC	GMAC_EMAC_EFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F0	GMAC_EMAC_PEFTSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F4	GMAC_EMAC_PEFTN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x11F8	GMAC_EMAC_PEFRSL	31:24	RUD[31:24]							
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x11FC	GMAC_EMAC_PEFRN	31:24			RUD[29:24]					
		23:16	RUD[23:16]							
		15:8	RUD[15:8]							
		7:0	RUD[7:0]							
0x1200 ... 0x125F	Reserved									
0x1260	GMAC_EMAC_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]							
0x1264	GMAC_EMAC_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]							
0x1268	GMAC_EMAC_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]							
0x126C ... 0x126F	Reserved									
0x1270	GMAC_EMAC_RXLPI	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x1274	GMAC_EMAC_RXLPITIME	31:24								
		23:16	LPITIME[23:16]							
		15:8	LPITIME[15:8]							
		7:0	LPITIME[7:0]							
0x1278	GMAC_EMAC_TXLPI	31:24								
		23:16								
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							
0x127C	GMAC_EMAC_TXLPITIME	31:24								
		23:16	LPITIME[23:16]							
		15:8	LPITIME[15:8]							
		7:0	LPITIME[7:0]							
0x1280 ... 0x12DF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x12E0	GMAC_EMAC_QOS_CFG	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]			
0x12E4 ... 0x12FF	Reserved									
0x1300	GMAC_EMAC_ASAB0	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1304	GMAC_EMAC_ASAT0	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1308	GMAC_EMAC_ASAB1	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x130C	GMAC_EMAC_ASAT1	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1310	GMAC_EMAC_ASAB2	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1314	GMAC_EMAC_ASAT2	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1318	GMAC_EMAC_ASAB3	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x131C	GMAC_EMAC_ASAT3	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1320	GMAC_EMAC_ASAB4	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1324	GMAC_EMAC_ASAT4	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1328	GMAC_EMAC_ASAB5	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x132C	GMAC_EMAC_ASAT5	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1330	GMAC_EMAC_ASAB6	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1334	GMAC_EMAC_ASAT6	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1338	GMAC_EMAC_ASAB7	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x133C	GMAC_EMAC_ASAT7	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1340	GMAC_EMAC_ASAB8	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1344	GMAC_EMAC_ASAT8	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1348	GMAC_EMAC_ASAB9	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x134C	GMAC_EMAC_ASAT9	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1350	GMAC_EMAC_ASAB10	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1354	GMAC_EMAC_ASAT10	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1358	GMAC_EMAC_ASAB11	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x135C	GMAC_EMAC_ASAT11	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1360	GMAC_EMAC_ASAB12	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1364	GMAC_EMAC_ASAT12	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1368	GMAC_EMAC_ASAB13	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x136C	GMAC_EMAC_ASAT13	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1370	GMAC_EMAC_ASAB14	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1374	GMAC_EMAC_ASAT14	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1378	GMAC_EMAC_ASAB15	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x137C	GMAC_EMAC_ASAT15	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1380	GMAC_EMAC_ASAB16	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1384	GMAC_EMAC_ASAT16	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1388	GMAC_EMAC_ASAB17	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x138C	GMAC_EMAC_ASAT17	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1390	GMAC_EMAC_ASAB18	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1394	GMAC_EMAC_ASAT18	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x1398	GMAC_EMAC_ASAB19	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x139C	GMAC_EMAC_ASAT19	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13A0	GMAC_EMAC_ASAB20	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13A4	GMAC_EMAC_ASAT20	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x13A8	GMAC_EMAC_ASAB21	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13AC	GMAC_EMAC_ASAT21	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13B0	GMAC_EMAC_ASAB22	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13B4	GMAC_EMAC_ASAT22	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13B8	GMAC_EMAC_ASAB23	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13BC	GMAC_EMAC_ASAT23	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13C0	GMAC_EMAC_ASAB24	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13C4	GMAC_EMAC_ASAT24	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13C8	GMAC_EMAC_ASAB25	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13CC	GMAC_EMAC_ASAT25	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13D0	GMAC_EMAC_ASAB26	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x13D4	GMAC_EMAC_ASAT26	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13D8	GMAC_EMAC_ASAB27	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13DC	GMAC_EMAC_ASAT27	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13E0	GMAC_EMAC_ASAB28	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13E4	GMAC_EMAC_ASAT28	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13E8	GMAC_EMAC_ASAB29	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13EC	GMAC_EMAC_ASAT29	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13F0	GMAC_EMAC_ASAB30	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13F4	GMAC_EMAC_ASAT30	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13F8	GMAC_EMAC_ASAB31	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x13FC	GMAC_EMAC_ASAT31	31:24			FILTBMASK[5:0]					
		23:16								FILTSORD
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1400 ... 0x14BB	Reserved									
0x14BC	GMAC_EMAC_CBSCR	31:24								
		23:16								
		15:8								
		7:0							QBE	QAE
0x14C0	GMAC_EMAC_CBSISQA	31:24	IS[31:24]							
		23:16	IS[23:16]							
		15:8	IS[15:8]							
		7:0	IS[7:0]							
0x14C4	GMAC_EMAC_CBSISQB	31:24	IS[31:24]							
		23:16	IS[23:16]							
		15:8	IS[15:8]							
		7:0	IS[7:0]							
0x14C8	GMAC_EMAC_TQUBA	31:24	TQUBA[31:24]							
		23:16	TQUBA[23:16]							
		15:8	TQUBA[15:8]							
		7:0	TQUBA[7:0]							
0x14CC	GMAC_EMAC_TXBDCTRL	31:24								
		23:16								
		15:8								
		7:0			TSMODE[1:0]					
0x14D0	GMAC_EMAC_RXBDCTRL	31:24								
		23:16								
		15:8								
		7:0			TSMODE[1:0]					
0x14D4	GMAC_EMAC_RQUBA	31:24	RQUBA[31:24]							
		23:16	RQUBA[23:16]							
		15:8	RQUBA[15:8]							
		7:0	RQUBA[7:0]							
0x14D8 ... 0x14FF	Reserved									
0x1500	GMAC_EMAC_ST1R	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]		
0x1504 ... 0x153F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1540	GMAC_EMAC_ST2R0	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]	
0x1544	GMAC_EMAC_ST2R1	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]	
0x1548 ... 0x157F	Reserved									
0x1580	GMAC_EMAC_TSCTL	31:24								
		23:16								
		15:8								
		7:0							TXSQ[1:0]	
0x1584 ... 0x158F	Reserved									
0x1590	GMAC_EMAC_TQBWRL	31:24								
		23:16								
		15:8								
		7:0	ALLOCQ0[7:0]							
0x1594 ... 0x159F	Reserved									
0x15A0	GMAC_EMAC_TQSA	31:24								
		23:16								
		15:8								
		7:0						SEGALLOCQ0[2:0]		
0x15A4 ... 0x16FF	Reserved									
0x1700	GMAC_EMAC_ST2CW0R0	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x1704	GMAC_EMAC_ST2CW1R0	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1708	GMAC_EMAC_ST2CW0R1	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x170C	GMAC_EMAC_ST2CW1R1	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1710	GMAC_EMAC_ST2CW0R2	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x1714	GMAC_EMAC_ST2CW1R2	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1718	GMAC_EMAC_ST2CW0R3	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x171C	GMAC_EMAC_ST2CW1R3	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1720	GMAC_EMAC_ST2CW0R4	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x1724	GMAC_EMAC_ST2CW1R4	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1728	GMAC_EMAC_ST2CW0R5	31:24	COMPVAL[15:8]							
		23:16	COMPVAL[7:0]							
		15:8	MASKVAL[15:8]							
		7:0	MASKVAL[7:0]							
0x172C	GMAC_EMAC_ST2CW1R5	31:24								
		23:16								
		15:8							DISMASK	OFFSSTRT[1]
		7:0	OFFSSTRT[0]	OFFSVAL[6:0]						
0x1730 ... 0x17FF	Reserved									
0x1800	GMAC_EMAC_ENST_START	31:24	START_SEC[1:0]		START_NSEC[29:24]					
		23:16	START_NSEC[23:16]							
		15:8	START_NSEC[15:8]							
		7:0	START_NSEC[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1804 ... 0x181F	Reserved									
0x1820	GMAC_EMAC_ENST_ON	31:24								
		23:16								ON_TIME[16]
		15:8	ON_TIME[15:8]							
		7:0	ON_TIME[7:0]							
0x1824 ... 0x183F	Reserved									
0x1840	GMAC_EMAC_ENST_OFF	31:24								
		23:16								OFF_TIME[16]
		15:8	OFF_TIME[15:8]							
		7:0	OFF_TIME[7:0]							
0x1844 ... 0x187F	Reserved									
0x1880	GMAC_EMAC_ENST_CR	31:24								
		23:16								
		15:8								
		7:0								EN_Q0
0x1884 ... 0x189F	Reserved									
0x18A0	GMAC_EMAC_FRER_TIMEOUT	31:24								
		23:16								
		15:8	TIMEOUT[15:8]							
		7:0	TIMEOUT[7:0]							
0x18A4	GMAC_EMAC_FRER_REDTAG	31:24	STRIP_R_TAG	SIX_BYTE_TAG						
		23:16								
		15:8	RED_TAG[15:8]							
		7:0	RED_TAG[7:0]							
0x18A8 ... 0x18BF	Reserved									
0x18C0	GMAC_EMAC_FRER_CTRL1_A	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x18C4	GMAC_EMAC_FRER_CTRL1_B	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x18C8	GMAC_EMAC_FRER_STAT1_A	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							
0x18CC	GMAC_EMAC_FRER_STAT1_B	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x18D0	GMAC_EMAC_FRER_CTRL2_A	31:24	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
		23:16								OFFSET_VALUE[8]
		15:8	OFFSET_VALUE[7:0]							
		7:0	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[3:0]			
0x18D4	GMAC_EMAC_FRER_CTRL2_B	31:24								
		23:16								
		15:8				SEQ_NUM_LENGTH[4:0]				
		7:0			SEQ_REC_WINDOW[5:0]					
0x18D8	GMAC_EMAC_FRER_STAT2_A	31:24							VEC_REC_ROGUE[9:8]	
		23:16	VEC_REC_ROGUE[7:0]							
		15:8							LATENT_ERRS[9:8]	
		7:0	LATENT_ERRS[7:0]							
0x18DC	GMAC_EMAC_FRER_STAT2_B	31:24								
		23:16	SEQRST_COUNT[7:0]							
		15:8							OUT_OF_ORDER[9:8]	
		7:0	OUT_OF_ORDER[7:0]							
0x18E0 ... 0x1AFF	Reserved									
0x1B00	GMAC_EMAC_RX_FLUSH_Q	31:24	MAX_VAL[15:8]							
		23:16	MAX_VAL[7:0]							
		15:8								
		7:0					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
0x1B04 ... 0x1B3F	Reserved									
0x1B40	GMAC_EMAC_SCR2_RATE_LIMIT0	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							
0x1B44	GMAC_EMAC_SCR2_RATE_LIMIT1	31:24	MAX_RATE_VAL[15:8]							
		23:16	MAX_RATE_VAL[7:0]							
		15:8	INTERVAL_TIME[15:8]							
		7:0	INTERVAL_TIME[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1B48 ... 0x1B7F	Reserved									
0x1B80	GMAC_EMAC_SCR2_RATE_STATUS	31:24								
		23:16								
		15:8								
		7:0							EXCESS_RATE_R1	EXCESS_RATE_R0

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			OSSCORR	EXTSELRQEN	PFCCTL	OSSMODE
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
		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 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.
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.

Value	Description
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
Must always be written to '0'.

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 16383 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	RXLPIS	PFCPAUSN				IDLE	MDIO	
Access	R	R				R	R	
Reset	0	0				1	x	

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	–	Reserved
1	RMII	Selects RMII mode.
2	RGMII	Selects RGMII mode.

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		TXFOMAXB	RXFOMAXB	DDRP
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
	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
			CRCERRREP	INFLASTEN	TXCOEN	TXPBMS	RXBMS[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
	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 26 – TXFOMAXB Force Transmit Max Burst Length
Forces the transmit DMA to always issue max length bursts on EOP (end of packet) or EOB (end of buffer) transfers as defined in FBLDO, even when there is less than max burst data bytes to read. Residual data read is ignored. Does not apply to bursts that break 1k boundary rule.

Bit 25 – RXFOMAXB Force Receive Max Burst Length
Forces the receive DMA to always issue max length bursts on EOP (end of packet) or EOB (end of buffer) transfers, even if there is less than max burst real packet data required to write. Any extra bytes of pad data is set to 0x00. Does not apply to bursts that break 1k boundary rule.

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 13 – 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 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 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)
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
						TXDMALCK	TXMACLCK	HRESP
Access						R/W	R/W	R/W
Reset						0	0	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 10 – TXDMALCK Transmit DMA Lockup (Clear by Writing a 1)
Set when lockup has been detected on the DMA transmit path. Writing a one clears this bit.

Bit 9 – TXMACLCK Transmit MAC Lockup (Clear by Writing a 1)
Set when lockup has been detected on the MAC transmit path. Writing a one clears this bit.

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.

The descriptors must be aligned at 32-bit boundaries and the descriptors are written to using two individual non sequential accesses.

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.

The descriptors must be aligned at 32-bit boundaries and the descriptors are read from memory using two individual non sequential accesses.

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
			RXDMALCK	RXMACLCK	HNO	RXOVR	REC	BNA
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – RXDMALCK Receive DMA Lockup (Clear by Writing a 1)
Set when lockup has been detected on the DMA receive path. Writing a one clears this bit.

Bit 4 – RXMACLCK Receive MAC Lockup (Clear by Writing a 1)
Set when lockup has been detected on the MAC receive path. Writing a one clears this bit.

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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
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
	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access	R	R	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access	R	R	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset	0	0	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

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[9:8]	
Access							R/W	R/W
Reset							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 9:0 – RPB1ADR[9:0] 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: 0x00002800
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	0	1	0	0	0

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	0	0	0	0	0	0	0	0

Bits 13:0 – FML[13:0] Frame Max Length
Rx jumbo frame maximum length.

62.8.20. 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.

62.8.21. 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).

62.8.22. GMAC Lockup Configuration Register

Name: GMAC_LCKUP_CFGR
Offset: 0x068
Reset: 0x07FFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TXDMA_LCKUP_EN	TXMAC_LCKUP_EN	RXDMA_LCKUP_EN	RXMAC_LCKUP_EN	LCKUP_REC_EN	DMA_LOCKUP_TIME[10:8]		
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	1

Bit	23	22	21	20	19	18	17	16
	DMA_LOCKUP_TIME[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
	PRESCALER[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
	PRESCALER[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 – TXDMA_LCKUP_EN Transmit DMA Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the transmit DMA.
1	Enables the monitor that detects lockups in the transmit DMA.

Bit 30 – TXMAC_LCKUP_EN Transmit MAC Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the transmit MAC.
1	Enables the monitor that detects lockups in the transmit MAC.

Bit 29 – RXDMA_LCKUP_EN Receive DMA Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the receive DMA.
1	Enables the monitor that detects lockups in the receive DMA.

Bit 28 – RXMAC_LCKUP_EN Receive MAC Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the receive MAC.
1	Enables the monitor that detects lockups in the receive MAC.

Bit 27 – LCKUP_REC_EN Lockup Recovery Enable

Value	Description
0	No effect.
1	Forces the MAC in Reset mode when a lockup is detected on the transmit or receive data paths.

Bits 26:16 – DMA_LOCKUP_TIME[10:0] Timeout Value for Receive and Transmit DMA
Defines the timeout value for receive and transmit DMA lockup detection, defined as a multiple of the prescaler value (PRESCALER). The MAC lockup time is defined in a separate configuration register (GMAC_LCKUP_TIME).

Bits 15:0 – PRESCALER[15:0] Prescaler Value for Timeout
Defines the prescaler value which is a multiple of the transmit clock.

62.8.23. GMAC MAC Lockup Time Register

Name: GMAC_LCKUP_TIME
Offset: 0x06C
Reset: 0x07FFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						TX_MAC_LOCKUP_TIME[10:8]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	23	22	21	20	19	18	17	16
	TX_MAC_LOCKUP_TIME[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
	RX_MAC_LOCKUP_TIME[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
	RX_MAC_LOCKUP_TIME[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 26:16 – TX_MAC_LOCKUP_TIME[10:0] Transmit MAC Lockup Detector Time
Defines the timeout value for the transmit MAC lockup detection defined as a multiple of the prescaler value (GMAC_LCKUP_CFGR.PRESCALER).

Bits 15:0 – RX_MAC_LOCKUP_TIME[15:0] Receive MAC Lockup Detector Time
Defines the timeout value for the receive MAC lockup detection defined as a multiple of the prescaler value (GMAC_LCKUP_CFGR.PRESCALER).

62.8.24. GMAC Transmit DMA Lockup Control Register

Name: GMAC_TXDMA_LCKUP_CR
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			LCKUP_EN_Q5	LCKUP_EN_Q4	LCKUP_EN_Q3	LCKUP_EN_Q2	LCKUP_EN_Q1	LCKUP_EN_Q0
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 – LCKUP_EN_Qx Transmit DMA Lockup Detector Enable for Queue x

Value	Description
0	Disables the transmit DMA lockup timer for queue x. The number of outstanding packets is still counted but the actual timer does not run.
1	Enables the transmit DMA lockup timer for queue x.

62.8.25. GMAC Receive Watermark Register

Name: GMAC_RX_WATERMARK
Offset: 0x07C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RX_LOW_WATERMARK[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
	RX_LOW_WATERMARK[7:0]							
Access	R/W	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
	RX_HIGH_WATERMARK[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
	RX_HIGH_WATERMARK[7:0]							
Access	R/W	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 – RX_LOW_WATERMARK[15:0] Receive Watermark Low Threshold
Defines the low threshold of the receive memory for which a zero length pause frame is transmitted when the memory fill level falls below the threshold (the action is performed only if the last pause frame transmitted was non-zero).

Bits 15:0 – RX_HIGH_WATERMARK[15:0] Receive Watermark High Threshold
Defines the high threshold of the receive memory for which a zero length pause frame is transmitted when the memory fill level exceeds the threshold.

62.8.26. 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.

62.8.27. 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.

62.8.28. 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.

62.8.29. 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.

62.8.30. 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.

62.8.31. 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.

62.8.32. 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.

62.8.33. 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.

62.8.34. 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.

62.8.35. 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.

62.8.36. 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.

62.8.37. 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.

62.8.38. 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.

62.8.39. 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.

62.8.40. 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.

62.8.41. 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.

62.8.42. 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.

62.8.43. 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].

62.8.44. 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.

62.8.45. 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.

62.8.46. 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.

62.8.47. 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.

62.8.48. 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.

62.8.49. 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).

62.8.50. 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.

62.8.51. 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.

62.8.52. 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.

62.8.53. 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.

62.8.54. 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.

62.8.55. 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.

62.8.56. 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.

62.8.57. 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.

62.8.58. 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.

62.8.59. 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.

62.8.60. 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.

62.8.61. 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.

62.8.62. 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.

62.8.63. 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.

62.8.64. 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.

62.8.65. 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.

62.8.66. 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.

62.8.67. 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.

62.8.68. 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.

62.8.69. 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.

62.8.70. 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.

62.8.71. 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.

62.8.72. 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.

62.8.73. 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.

62.8.74. 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.

62.8.75. 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.

62.8.76. 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.

62.8.77. 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.

62.8.78. 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.

62.8.79. 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.

62.8.80. 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.

62.8.81. 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.

62.8.82. 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.

62.8.83. 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.

62.8.84. 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.

62.8.85. 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.

62.8.86. 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.

62.8.87. 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.

62.8.88. 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.

62.8.89. 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								

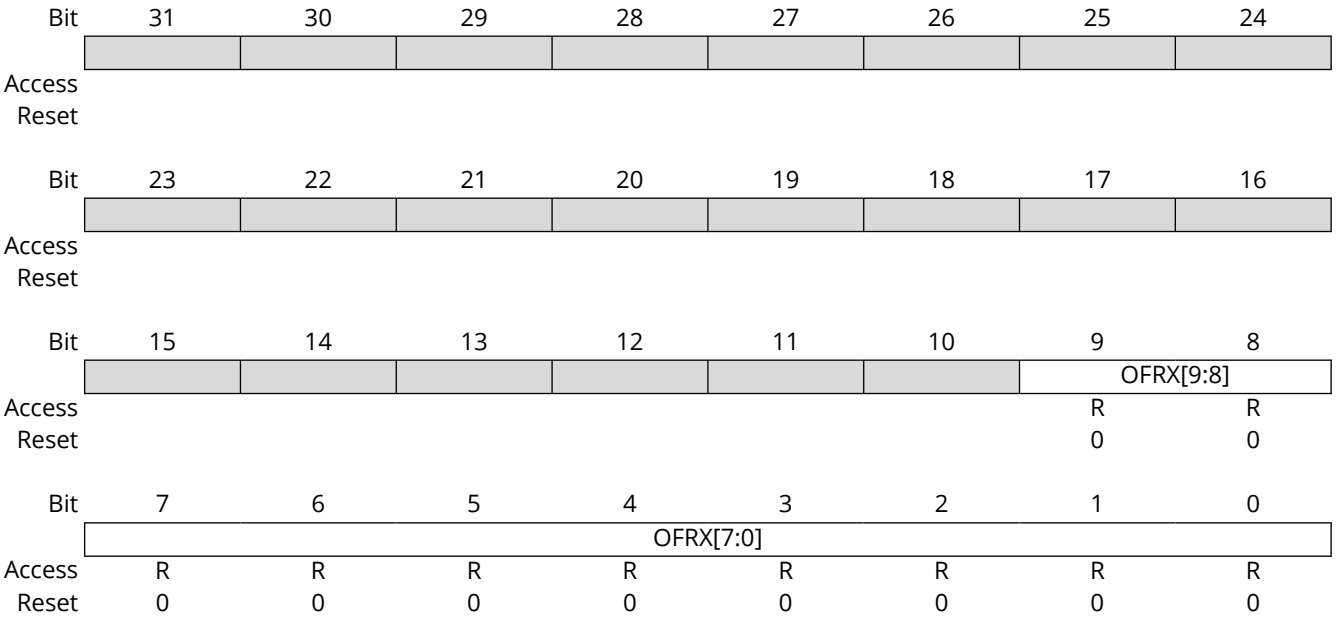
Bit	15	14	13	12	11	10	9	8
							UFRX[9:8]	
Access							R	R
Reset							0	0

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.

62.8.90. GMAC Oversized Frames Received Register

Name: GMAC_OFR
Offset: 0x188
Reset: 0x00000000
Property: Read-only



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.

62.8.91. 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.

62.8.92. 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.

62.8.93. 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.

62.8.94. 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.

62.8.95. 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.

62.8.96. 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
							RXRER[17:16]	
Access							R	R
Reset							0	0

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

Bits 17:0 – RXRER[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.

62.8.97. 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.

62.8.98. 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, 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.

62.8.99. 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) and do not have a CRC error, an alignment error, nor a symbol error.

62.8.100.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, 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.

62.8.101.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.

62.8.102.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.

62.8.103.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.

62.8.104.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.

62.8.105.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.

62.8.106.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.

62.8.107.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.

62.8.108.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.

62.8.109.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.

62.8.110.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.

62.8.111.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.

62.8.112.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.

62.8.113.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.

62.8.114.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.

62.8.115.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.

62.8.116.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 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 bus 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

62.8.121.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]							TXBQDIS
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 – TXBQBA[29:0] Transmit Buffer Queue Base Address
Written with the address of the start of the transmit queue.

Bit 0 – TXBQDIS Transmit Buffer Queue Disable

Value	Description
0	No effect.
1	Disables the transmit queue. This can be used to reduce the number of active queues and must be changed only while transmit is disabled.

62.8.122.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]							RXBQDIS
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 – RXBQBA[29:0] Receive Buffer Queue Base Address
Written with the address of the start of the receive queue.

Bit 0 – RXBQDIS Receive Buffer Queue Disable

Value	Description
0	No effect.
1	Disables the receive queue. This can be used to reduce the number of active queues and must be changed only while receive is disabled.

62.8.123.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.

62.8.124.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.

62.8.125.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.

62.8.126.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.

62.8.127.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.

62.8.128.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.

62.8.129.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.

62.8.130.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.

62.8.131.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.

62.8.132.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.

Value	Description
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.

62.8.133.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

62.8.134.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).

62.8.135.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 16 segments can be granted.

62.8.136.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

62.8.137.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

62.8.138.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

62.8.139.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.

62.8.140.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.

62.8.141.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.

62.8.142.GMAC ENST Start Time Queue Register x

Name: GMAC_ENST_START_Qx
Offset: 0x0800 + x*0x04 [x=0..5]
Reset: 0x0001FFFF
Property: R/W

Bit	31	30	29	28	27	26	25	24
	START_SEC[1:0]		START_NSEC[29:24]					
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
	START_NSEC[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	1

Bit	15	14	13	12	11	10	9	8
	START_NSEC[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
	START_NSEC[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 31:30 – START_SEC[1:0] Seconds for Start Time
Defines the seconds for the absolute start time of the queue.

Bits 29:0 – START_NSEC[29:0] Nanoseconds for Start Time
Defines the nanoseconds for the absolute start time of the queue.

62.8.143.GMAC ENST On Time Queue Register x

Name: GMAC_ENST_ON_Qx
Offset: 0x0820 + x*0x04 [x=0..5]
Reset: 0x0001FFFF
Property: R/W

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ON_TIME[16]
Access								R/W
Reset								1

Bit	15	14	13	12	11	10	9	8
	ON_TIME[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
	ON_TIME[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 16:0 – ON_TIME[16:0] Time When Queue is Open
Defines the seconds of the time for which the queue is open.

62.8.144.GMAC ENST Off Time Queue Register x

Name: GMAC_ENST_OFF_Qx
Offset: 0x0840 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: R/W

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

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

Bit	15	14	13	12	11	10	9	8
	OFF_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
	OFF_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 16:0 – OFF_TIME[16:0] Time When Queue is Blocked
Defines the seconds of the time for which the queue is blocked.

62.8.145.GMAC ENST Control Register

Name: GMAC_ENST_CR
Offset: 0x880
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
			EN_Q5	EN_Q4	EN_Q3	EN_Q2	EN_Q1	EN_Q0
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 – EN_Qx Enhanced Scheduled Traffic Enable for Queue x

Value	Description
0	Disables the enhanced scheduled traffic for queue x.
1	Enables the enhanced scheduled traffic for queue x. EMAC has only 1 queue and ENST is enabled by writing EN_Q0.

62.8.146.GMAC Frame Elimination Timeout Register

Name: GMAC_FRER_TIMEOUT
Offset: 0x8A0
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
	TIMEOUT[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
	TIMEOUT[7:0]							
Access	R/W	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 – TIMEOUT[15:0] Sequence Recovery Timer Restart Period for Credit Based Streams
Determines when the sequence recovery timers are restarted and used by all of the configured CB streams. It is programmed as a count of 8192 rx_clk periods. 8192 rx_clk periods is 65.536 microseconds at gigabit speed and 327.68 microseconds at 100M speed. This allows a max timeout value of about 4 seconds at gigabit speed.

62.8.147.GMAC Frame Elimination Redundancy Tag Register

Name: GMAC_FRER_REDTAG
Offset: 0x8A4
Reset: 0x4000F1C1
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	STRIP_R_TAG	SIX_BYTE_TAG						
Access	R/W	R/W						
Reset	0	1						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RED_TAG[15:8]							
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	1

Bit	7	6	5	4	3	2	1	0
	RED_TAG[7:0]							
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	1

Bit 31 – STRIP_R_TAG Stripping Redundancy Tag Enable

Value	Description
0	Disables the stripping function. When the statistics counters need to reflect the actual number of octets received, then the stripping functionality must be disabled.
1	Enables the stripping function, the receive octet counters reflect post deletion frame size so the frame elimination functionality is transparent to higher level management.

Bit 30 – SIX_BYTE_TAG Six-byte Tag Enable

Value	Description
0	Defines a four-byte tag as per 802.1CB standard revision 2.4 and earlier.
1	Enables the six-byte tag as per 802.1CB standard revision 2.4 and later.

Bits 15:0 – RED_TAG[15:0] Redundancy Tag (R-TAG)
Defines the Ethertype value used to identify the redundancy tag (R-TAG).

62.8.148.GMAC Frame Elimination Control A Register x

Name: GMAC_FRER_CTRL_Ax
Offset: 0x08C0 + (x-1)*0x10 [x=1..4]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
								OFFSET_VALUE[8]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OFFSET_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

Bit	7	6	5	4	3	2	1	0
	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[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 – EN_ELIMINATION 802.1CB Elimination of Received Frames Enable

Value	Description
0	Disables the elimination of received frames.
1	Enables the elimination of received frames.

Bit 30 – EN_VECTOR_REC_ALG 802.1CB Vector Recovery Algorithm Enable

Value	Description
0	Enables the match recovery algorithm.
1	Enables the vector recovery algorithm.

Bit 29 – EN_SEQRECRST_TIMER 802.1CB Sequence Recovery Reset Timer Enable

Value	Description
0	Disables the sequence recovery reset timer.
1	Enables the sequence recovery reset timer.

Bit 28 – USE_R_TAG Redundancy Tag Enable

Value	Description
0	Identifies bottom of sequence number with OFFSET_VALUE.
1	Identifies sequence number with redundancy tag.

Bits 16:8 – OFFSET_VALUE[8:0] Offset in Bytes from Start Packet Delimiter to MSB for 802.1CB Sequence Number

Defines the offset value in bytes from the start packet delimiter to the most significant byte of the 802.1CB sequence number. 9-bit width allows a TCP sequence number to be used with IPv6. This value must only be changed when EN_ELIMINATION=0.

Bits 7:4 – MEMBER_STREAM_2[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

Bits 3:0 – MEMBER_STREAM_1[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

62.8.149.GMAC Frame Elimination Control B Register x

Name: GMAC_FRER_CTRL_Bx
Offset: 0x08C4 + (x-1)*0x10 [x=1..4]
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
				SEQ_NUM_LENGTH[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
			SEQ_REC_WINDOW[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 12:8 – SEQ_NUM_LENGTH[4:0] Number of Significant Bits of the 802.1CB Sequence Number
Defines the number of LSBs to consider in the 802.1CB sequence number.
The vector recovery algorithm does not work for values lower than 4. The value 0 and values greater than 16 are equivalent to 16.

Bits 5:0 – SEQ_REC_WINDOW[5:0] Vector Recovery Window
Defines the window size used by the vector recovery algorithm to determine whether to reject a packet. A value of zero means the entire history vector is used.

62.8.150.GMAC Frame Elimination Statistics A Register x

Name: GMAC_FRER_STAT_Ax
Offset: 0x08C8 + (x-1)*0x10 [x=1..4]
Reset: 0x00000000
Property: Read-only

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
							VEC_REC_ROGUE[9:8]	
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VEC_REC_ROGUE[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
							LATENT_ERRS[9:8]	
Access							R	R
Reset							0	0

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

Bits 25:16 – VEC_REC_ROGUE[9:0] Number of Dropped Frames (Clear on read)
Returns the number of frames dropped by the vector recovery algorithm for being out of range.

Bits 9:0 – LATENT_ERRS[9:0] Number of Sequence Numbers Seen Without a Duplicate (Clear on read)
Returns the number of sequence numbers seen without a duplicate.
The count is updated when a frame is dropped from the history vector. So the update only happens after a new frame is received.

62.8.151.GMAC Frame Elimination Statistics B Register x

Name: GMAC_FRER_STAT_Bx
Offset: 0x08CC + (x-1)*0x10 [x=1..4]
Reset: 0x00000000
Property: Read-only

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SEQRST_COUNT[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
							OUT_OF_ORDER[9:8]	
Access							R	R
Reset							0	0

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

Bits 23:16 – SEQRST_COUNT[7:0] Number of Times the Sequence Recovery Reset Timer Decrements to Zero (Clear on read)
Returns the number of times the sequence recovery reset timer decrements to zero.

Bits 9:0 – OUT_OF_ORDER[9:0] Out of Order Sequence Numbers Received (Clear on read)
Returns of out of order sequence numbers received. Incremented when a frame is accepted but the sequence number is not +1 of the highest stored value.

62.8.152.GMAC Receive Queue Flush Register x

Name: GMAC_RX_FLUSH_Qx
Offset: 0x0B00 + x*0x04 [x=0..5]
Reset: 0x00000000
Property: R/W

This register defines the traffic policing mode of operation. Each mode can be set simultaneously with the exception of bits 2 and 3, which are exclusive. If bits 2 and 3 are both set, then only bit 3 is treated as active

Bit	31	30	29	28	27	26	25	24
	MAX_VAL[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
	MAX_VAL[7:0]							
Access	R/W	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
					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:16 – MAX_VAL[15:0] Maximum Value for the Received Frame Size or Number of 128-Byte Chunk
Defines the maximum value for the received frame size when the bit LIMIT_FRAME_SIZE=1 or the number of 128 byte chunks of data received for this queue and already stored in the memory of the queue awaiting DMA memory writes when LIMIT_NUM_BYTES=1 and LIMIT_FRAME_SIZE=0.

Bit 3 – LIMIT_FRAME_SIZE Maximum Frame Length Received

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	When set, MAX_VAL indicates the maximum frame length in bytes that may be received. Frames exceeding this length will be dropped. This traffic policing function is relevant to the 802.1Qci standard which specifies stream filtering based on a maximum service data unit (SDU) size.

Bit 2 – LIMIT_NUM_BYTES Limitation of the Number of 128-Byte Chunk of Data Stored in the Memory of this Queue

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	Limits the number of 128-byte chunks of data received for this queue and already stored in the memory of the queue awaiting DMA memory writes to the value defined in the field MAX_VAL.

Bit 1 – DROP_ON_RESRC_ERR Drop on Resource Error

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	If a free DMA descriptor for this queue cannot be obtained (also referred to as lack of descriptor resource and occurs when the software either cannot free up descriptors quickly enough to meet the receive traffic rate or has deliberately decided not to free any descriptors), all new frames received on this queue will be automatically discarded.

Bit 0 – DROP_ALL Drop All Frames

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	Drops all frames of this queue.

62.8.153.GMAC Screening Type 2 Rate Limit Register x

Name: GMAC_SCR2_RATE_LIMITx
Offset: 0x0B40 + x*0x04 [x=0..7]
Reset: 0x00000000
Property: R/W

This register manages the traffic policing function relevant to the 802.1Qci standard and associated to type 2 screener.

Bit	31	30	29	28	27	26	25	24
	MAX_RATE_VAL[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
	MAX_RATE_VAL[7:0]							
Access	R/W	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
	INTERVAL_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
	INTERVAL_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 31:16 – MAX_RATE_VAL[15:0] Maximum Rate Value for the Interval Time
Defines the maximum number of bytes the type 2 screener is permitted to match in the programmed interval time.

Bits 15:0 – INTERVAL_TIME[15:0] Interval Time for Maximum Rate Checking
Defines the period for which the total number of bytes of received frames matched by the screener are accumulated and compared with the value configured in MAX_RATE_VAL. When the value exceeds MAX_RATE_VAL, then the current frame and frames subsequently matched will dropped until an interval time passes where MAX_RATE_VAL is not exceeded. When MAX_RATE_VAL=0, then no rate limiting will be performed. The interval time is specified in units of 64 receive clock periods.

62.8.154.GMAC Screening Type 2 Rate Status Register

Name: GMAC_SCR2_RATE_STATUS
Offset: 0x0B80
Reset: 0x00000000
Property: Read-only

This register manages the traffic policing function relevant to the 802.1Qci standard and associated to type 2 screener.

Bit	31	30	29	28	27	26	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
	EXCESS_RATE_R7	EXCESS_RATE_R6	EXCESS_RATE_R5	EXCESS_RATE_R4	EXCESS_RATE_R3	EXCESS_RATE_R2	EXCESS_RATE_R1	EXCESS_RATE_R0
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 – EXCESS_RATE_Rx Excessive Screener Rate Register x

Value	Description
0	No excessive rate in screener since the last read of GMAC_SCR2_RATE_STATUS.
1	A screener rate limiting mechanism has been triggered since the last read of GMAC_SCR2_RATE_STATUS.

62.8.155.GMAC MMSL Control Register

Name: GMAC_MMSL_CR
Offset: 0xF00
Reset: 0x00000020
Property: Read/Write

This register contains the control bits for the 802.3br MAC Merge Sublayer (MMSL).

Bit	31	30	29	28	27	26	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
			ROUTE_RX_TO_PMAC	RESTART_VER	PRE_ENABLE	VERIFY_DISABLE	ADD_FRAG_SIZE[1:0]	
Access			R/W	W	R/W	R/W	R/W	R/W
Reset			1	0	0	0	0	0

Bit 5 – ROUTE_RX_TO_PMAC Route Received Frames to Preemptive MAC

Value	Description
0	Routes all received frames to express MAC (eMAC).
1	Routes all received frames to preemptive MAC (pMAC).

Bit 4 – RESTART_VER Restart the Verification Procedure (Write-only)

Value	Description
0	No effect.
1	Restarts the verification procedure. Writing a one to this bit has no effect if any of the following are true: PRE_ENABLE=0, VERIFY_DISABLE=1 or the verification procedure is already in progress. When preemption is active, ROUTE_RX_TO_PMAC must be set to zero before writing to the RESTART_VER bit. This bit always returns zero when read.

Bit 3 – PRE_ENABLE Preemption Operation Enable

Value	Description
0	Disables the preemption operation and verifies mpackets will not be responded to or sent. If a preemption is in progress, the current preemption will complete and no further preemption will occur.
1	Enables the preemption by starting the verification procedure when VERIFY_DISABLE=0. Preemption can occur only once the verification process completes. If VERIFY_DISABLE=1, preemption can occur immediately.

Bit 2 – VERIFY_DISABLE 802.3br Support Check for the Link Partner

Value	Description
0	Enables the 802.3br support check for the link partner.

Value	Description
1	Disables the 802.3br support check for the link partner and preemption is enabled as soon as PRE_ENABLE=1.

Bits 1:0 – ADD_FRAG_SIZE[1:0] PMAC Minimum Number of Bytes before Preemption

Value	Name	Description
0	64_BYTES	A minimum of 64 bytes can be sent by pMAC before any preemption. The value can be changed only when PRE_ENABLE is cleared.
1	128_BYTES	A minimum of 128 bytes can be sent by pMAC before any preemption. The value can be changed only when PRE_ENABLE is cleared.
2	192_BYTES	A minimum of 192 bytes can be sent by pMAC before any preemption. The value can be changed only when PRE_ENABLE is cleared.
3	256_BYTES	A minimum of 256 bytes can be sent by pMAC before any preemption. The value can be changed only when PRE_ENABLE is cleared.

62.8.156.GMAC MMSL Status Register

Name: GMAC_MMSL_SR
Offset: 0xF04
Reset: 0x00000000
Property: Read-only

This register contains the status bits for the 802.3br MAC Merge Sublayer (MMSL).

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						SMD_ERR	FRER_COUNT_ERR	SMDC_ERR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	SMDS_ERR	RCV_V_ERR	RCV_R_ERR	VERIFY_STATUS[2:0]			RESPOND_STATUS	PRE_ACTIVE
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 10 – SMD_ERR Illegal Start Mpacket Delimiter Received (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An illegal SMD (different from Express, Verify, Response, Start Preemptible or Continuation Preemptible SMD) has been received since the last read of GMAC_MMSL_SR.

Bit 9 – FRER_COUNT_ERR Frame Counter Error (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	A frame counter error occurred since the last read of GMAC_MMSL_SR. A frame counter error occurs when the SMD-C received indicates a different frame count than expected (i.e. the fragment belongs to another frame and not to the Start packet already received before this) or when a fragment error occurred, which means the field following the SMD-C received was encoding a different fragment count than it was supposed to be.

Bit 8 – SMDC_ERR SMD-C Received When Waiting an SMD-S (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An SMD-C has been received when an SMD-S was expected, since the last read of GMAC_MMSL_SR.

Bit 7 – SMDS_ERR SMD-S Received When Waiting an SMD-C (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.

Value	Description
1	An SMD-S has been received when an SMD-C was expected, since the last read of GMAC_MMSL_SR.

Bit 6 – RCV_V_ERR Incorrect Verification Mpacket Received (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An incorrect verification mPacket has been received since the last read of GMAC_MMSL_SR.

Bit 5 – RCV_R_ERR Incorrect Response Mpacket Received (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An incorrect response mPacket has been received since the last read of GMAC_MMSL_SR.

Bits 4:2 – VERIFY_STATUS[2:0] Verification Status

Value	Name	Description
0	INIT_VERIFICATION	Initialization
1	VERIFICATION_IDLE	Idle
2	SEND_VERIFY	Sending a verify command
3	WAIT_FOR_RESPONSE	Waiting for a response
4	VERIFIED	Verified
5	VERIFY_FAIL	Failure during the verify operation

Bit 1 – RESPOND_STATUS Response Status

Value	Name	Description
0	R_IDLE	Idle
1	SEND_RESPOND	Sending

Bit 0 – PRE_ACTIVE Preemption Status

Value	Description
0	Preemption is inactive.
1	Preemption is active when the verification process is completed or when GMAC_MMSL_CR.VERIFY_DISABLE=1 and GMAC_MMSL_CR.PRE_ACTIVE=1.

62.8.157.GMAC MMSL Error Statistics Register

Name: GMAC_MMSL_ESR
Offset: 0xF08
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
	SMD_ERROR_COUNT[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
	ASS_ERROR_COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – SMD_ERROR_COUNT[7:0] Number of Rejected Frames or Fragments due to unknown SMD (Clear on read)

Returns the number of received MAC frames or fragments rejected due to unknown SMD value or arriving with an SMD-C when no frame is in progress, since the last read of the GMAC_MMSL_ESR.

Bits 7:0 – ASS_ERROR_COUNT[7:0] Number of Frames with Reassembly Errors (Clear on read)

Returns the number of frames with reassembly errors. The counter is incremented every time the ASSEMBLY_ERROR state in the receive processing state diagram is entered, since the last read of the GMAC_MMSL_ESR.

62.8.158.GMAC MMSL Frame Re-Assembled OK Register

Name: GMAC_MMSL_ASS_OK
Offset: 0xF0C
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
								ASS_OK_COUNT[16]
Access								R
Reset								0

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

Bits 16:0 – ASS_OK_COUNT[16:0] Number of Correctly Re-Assembled Frames (Clear on read)
Returns the number of correctly re-assembled frames that were delivered to the MAC since the last read of the GMAC_MMSL_ASS_OK.

62.8.159.GMAC MMSL Received Fragments Count Register

Name: GMAC_MMSL_RXFRAG_CNT
Offset: 0xF10
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
								FRAG_COUNT_RX[16]
Access								R
Reset								0

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

Bits 16:0 – FRAG_COUNT_RX[16:0] Number of Received Fragments (Clear on read)
Returns the number of additional mPackets received due to preemption since the last read of the GMAC_MMSL_ASS_OK. The counter does not roll if its maximum value is reached.

62.8.160.GMAC MMSL Transmitted Fragments Count Register

Name: GMAC_MMSL_TXFRAG_CNT
Offset: 0xF14
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
								FRAG_COUNT_TX[16]
Access								R
Reset								0

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

Bits 16:0 – FRAG_COUNT_TX[16:0] Number of Transmitted Fragments (Clear on read)
Returns the number of additional mPackets transmitted due to preemption since the last read of the GMAC_MMSL_ASS_OK. The counter does not roll if its maximum value is reached.

62.8.161.GMAC MMSL Interrupt Status Register

Name: GMAC_MMSL_ISR
Offset: 0xF18
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
			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – SMD_ERR Illegal SMD Received (Clear on read)

Value	Description
0	No illegal SMD received since the last read of GMAC_MMSL_ISR.
1	Illegal SMD received (different from Express, Verify, Response, Start Preemptible or Continuation Preemptible) since the last read of GMAC_MMSL_ISR

Bit 4 – FR_COUNT_ERR Illegal SMD Received (Clear on read)

Value	Description
0	No frame count error detected since the last read of GMAC_MMSL_ISR.
1	The SMD-C received since the last read of GMAC_MMSL_ISR indicates a different frame count than expected (i.e. the fragment belongs to another frame and not to the Start packet already received before this) or that a fragment error hoccurred, meaning that the field following the SMD-C received was encoding a different fragment count than it was supposed to be.

Bit 3 – SMDC_ERR SMD-C Received When Waiting an SMD-S (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An SMD-C has been received when an SMD-S was expected since the last read of GMAC_MMSL_SR.

Bit 2 – SMDS_ERR SMD-S Received When Waiting an SMD-C (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An SMD-S has been received when an SMD-C was expected since the last read of GMAC_MMSL_SR.

Bit 1 – RCV_V_ERR Incorrect Verification Mpacket Received (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An incorrect verification mPacket has been received since the last read of GMAC_MMSL_SR.

Bit 0 – RCV_R_ERR Incorrect Response Mpacket Received (Clear on read)

Value	Description
0	No error since the last read of GMAC_MMSL_SR.
1	An incorrect response mPacket has been received since the last read of GMAC_MMSL_SR.

62.8.162.GMAC MMSL Interrupt Enable Register

Name: GMAC_MMSL_IER
Offset: 0xF1C
Reset: 0x00000000
Property: Read/Write

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
			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – SMD_ERR Illegal SMD Received

Bit 4 – FR_COUNT_ERR Illegal SMD Received

Bit 3 – SMDC_ERR SMD-C Received When Waiting an SMD-S

Bit 2 – SMDS_ERR SMD-S Received When Waiting an SMD-C

Bit 1 – RCV_V_ERR Incorrect Verification Mpacket Received

Bit 0 – RCV_R_ERR Incorrect Response Mpacket Received

62.8.163.GMAC MMSL Interrupt Disable Register

Name: GMAC_MMSL_IDR
Offset: 0xF20
Reset: 0x00000000
Property: Read/Write

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
			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – SMD_ERR Illegal SMD Received

Bit 4 – FR_COUNT_ERR Illegal SMD Received

Bit 3 – SMDC_ERR SMD-C Received When Waiting an SMD-S

Bit 2 – SMDS_ERR SMD-S Received When Waiting an SMD-C

Bit 1 – RCV_V_ERR Incorrect Verification Mpacket Received

Bit 0 – RCV_R_ERR Incorrect Response Mpacket Received

62.8.164.GMAC MMSL Interrupt Mast Register

Name: GMAC_MMSL_IMR
Offset: 0xF24
Reset: 0x00000000
Property: Read-only

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt source is disabled

1: corresponding interrupt source 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
			SMD_ERR	FR_COUNT_ERR	SMDC_ERR	SMDS_ERR	RCV_V_ERR	RCV_R_ERR
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – SMD_ERR Illegal SMD Received

Bit 4 – FR_COUNT_ERR Illegal SMD Received

Bit 3 – SMDC_ERR SMD-C Received When Waiting an SMD-S

Bit 2 – SMDS_ERR SMD-S Received When Waiting an SMD-C

Bit 1 – RCV_V_ERR Incorrect Verification Mpacket Received

Bit 0 – RCV_R_ERR Incorrect Response Mpacket Received

62.8.165.GMAC Express MAC Network Control Register

Name: GMAC_EMAC_NCR
Offset: 0x1000
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		IFGQAVCRED			OSSCORR	EXTSELRQEN	PFCCTL	OSSMODE
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
		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	R/W	R/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 27 – OSSCORR 1588 One Step Sync Mode Correction Field

Value	Description
0	Disables updating the correction field of PTP 1588 version 2 sync frames by adding current TSU timer value.
1	Enables updating 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.
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 (Write-only)

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 (Write-only)

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

Value	Description
0	Disables PFC Priority Based Pause Reception capabilities.
1	Enables PFC Priority Based Pause Reception capabilities by enabling the PFC negotiation and recognition of priority-based pause frames.

Bit 15 – SRTSM Store Receive Timestamp to Memory

Value	Description
0	Normal operation.
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 (Write-only)

Value	Description
0	No effect.
1	Generates a pause frame with zero quantum to be transmitted.

Bit 11 – TXPF Transmit Pause Frame (Write-only)

Value	Description
0	No effect.
1	Generates a pause frame to be transmitted.

Bit 10 – THALT Transmit Halt (Write-only)

Value	Description
0	No effect.
1	Halts transmission as soon as any ongoing frame transmission ends.

Bit 9 – TSTART Start Transmission (Write-only)

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 (Write-only)
Writing a one increments all the statistics registers by one for test purposes.

Bit 5 – CLRSTAT Clear Statistics Registers (Write-only)

Value	Description
0	No effect.
1	Clears the statistics registers.

Bit 4 – MPE Management Port Enable

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

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 Loopback Local

Value	Description
0	Normal operating mode (no loopback).
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 loopback. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loopback.

62.8.166.GMAC Express MAC Network Configuration Register

Name: GMAC_EMAC_NCFGR
Offset: 0x1004
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

Value	Description
0	
1	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

Value	Description
0	Rejects frames with non-standard preamble.
1	Accepts frames with non-standard preamble.

Bit 28 – IPGSEN Inter Packet Gap Stretch Enable

Value	Description
0	The transmit IPG cannot be increased.
1	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

Value	Description
0	Normal operation, frames with FCS/CRC errors are rejected.
1	Frames with FCS/CRC errors are rejected. FCS error statistics are still collected for frames with bad FCS and FCS status is recorded in frame's DMA descriptor.

Bit 25 – EFRHD Enable Frames Received in Half Duplex

Value	Description
0	Disables frames to be received in Half Duplex mode while transmitting.
1	Enables frames to be received in Half Duplex mode while transmitting.

Bit 24 – RXCOEN Receive Checksum Offload Enable

Value	Description
0	Disables the receive checksum engine. Frames with bad IP, TCP or UDP checksums are accepted.
1	Enables the receive checksum engine. Frames with bad IP, TCP or UDP checksums are discarded.

Bit 23 – DCPF Disable Copy of Pause Frames

Value	Description
0	Copies pause frames to the system memory.
1	Prevents valid pause frames being copied to memory. 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] Always Written to 0
Must be always be written to '0'.

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

Value	Description
0	Includes the received frame check sequence (last 4 bytes) when writing to memory.
1	Excludes the received frame check sequence (last 4 bytes) when writing to memory. The frame length indicated will be reduced by four bytes in this mode.

Bit 16 – LFERD Length Field Error Frame Discard

Value	Description
0	Accepts frames with a measured length shorter than the extracted length field
1	Discards frames with a measured length shorter than the extracted length field (as indicated by bytes 13 and 14 in a non-VLAN tagged frame). 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

Value	Description
0	Does not pause the transmission when a non-zero 802.3 classic pause frame is received.
1	Pauses transmission when a non-zero 802.3 classic pause frame is received and PFC has not been negotiated.

Bit 12 – RTY Retry Test0

Value	Description
0	Normal operation.
1	The backoff between collisions will always be one slot time and 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

Value	Description
0	Operates in 10/100 Mbps mode.
1	Operates in Gigabit mode.

Bit 8 – MAXFS 1536 Maximum Frame Size

Value	Description
0	Rejects frame sizes above 1518 bytes.
1	Accepts frames up to 1536 bytes in length. Normally the GMAC would reject any frame above 1518 bytes.

Bit 7 – UNIHEN Unicast Hash Enable

Value	Description
0	Rejects unicast frames.
1	Accepts unicast frames 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

Value	Description
0	Rejects multicast frames.
1	Accepts multicast frames 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

Value	Description
0	Accepts broadcast frames.
1	Rejects frames addressed to the broadcast address of all ones.

Bit 4 – CAF Copy All Frames

Value	Description
0	Discards invalid frames.
1	Accepts all valid frames.

Bit 3 – JFRAME Jumbo Frame Size

Value	Description
0	Disables jumbo frames.
1	Enables jumbo frames up to 16383 bytes to be accepted. The default length is 10240 bytes.

Bit 2 – DNVLAN Discard Non-VLAN FRAMES

Value	Description
0	Passes all frames to address matching logic
1	Passes only VLAN tagged frames to the address matching logic.

Bit 1 – FD Full Duplex

Value	Description
0	Half-duplex mode.
1	The transmit block ignores the state of collision and carrier sense and allows receive while transmitting.

Bit 0 – SPD Speed

Value	Description
0	MAC operates at 10 Mbps.
1	MAC operates at 100 Mbps.

62.8.167.GMAC Express MAC Network Status Register

Name: GMAC_EMAC_NSR
Offset: 0x1008
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
	RXLPIS	PFCPAUSN				IDLE	MDIO	
Access	R	R				R	R	
Reset								

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.

62.8.168.GMAC Express MAC DMA Configuration Register

Name: GMAC_EMAC_DCFGR
Offset: 0x1010
Reset: 0x00020004
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
			TXBD_EXTENDED	RXBD_EXTENDED		TXFOMAXB	RXFOMAXB	DDRP
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
	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	1	0

Bit	15	14	13	12	11	10	9	8
			CRCERRREP	INFLASTEN	TXCOEN	TXPBMS	RXBMS[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
	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	1	0	0

Bit 29 – TXBD_EXTENDED Transmit Buffer Descriptor Extended Mode
See GMAC_EMAC_TXBDCTRL 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_EMAC_RXBDCTRL for a description of the features.

Value	Description
0	Disables Receive Buffer Data Extended mode.
1	Enables Receive Buffer Data Extended mode.

Bit 26 – TXFOMAXB Force Transmit Max Burst Length
Forces the transmit DMA to always issue max length bursts on EOP (end of packet) or EOB (end of buffer) transfers as defined in FBLDO, even when there is less than max burst data bytes to read. Residual data read is ignored. Does not apply to bursts that break 1k boundary rule.

Bit 25 – RXFOMAXB Force Receive Max Burst Length
Forces the receive DMA to always issue max length bursts on EOP (end of packet) or EOB (end of buffer) transfers, even if there is less than max burst real packet data required to write. Any extra bytes of pad data is set to 0x00. Does not apply to bursts that break 1k boundary rule.

Bit 24 – DDRP DMA Discard Receive Packets

When set, the GMAC DMA will automatically discard receive packets from the receiver packet buffer memory when no system memory resource is available.
When low, the received packets will remain to be stored in the GMAC local memory packet buffer until system memory buffer resource next 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 13 – 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 12 – INFLASTEN Infinite Size for Last Buffer Enable

Set to '1', 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 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

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 '1' 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

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

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	LITTLE_ENDIAN	Selects Little-endian endianism for system bus transfers.
1	BIG_ENDIAN	Selects swapped endianism for system bus transfers. FBLDO: Fixed Burst Length for DMA Data Operations:

Bits 4:0 – FBLDO[4:0] Fixed Burst Length for DMA Data Operations

Value	Name	Description
0	–	Reserved
1	SINGLE	Always uses single access on system bus
2	–	Reserved
4	INCR4	Attempt to use 4-beat bursts on system bus (Default)
8	INCR8	Attempt to use 8-beat bursts on system bus bursts
16	INCR16	Attempt to use 16-beat bursts on system bus bursts 50

62.8.169.GMAC Express MAC Transmit Status Register

Name: GMAC_EMAC_TSR
Offset: 0x1014
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
						TXDMALCK	TXMACLCK	HRESP
Access						R/W	R/W	R/W
Reset						0	0	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/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit 10 – TXDMALCK Transmit DMA Lockup (Clear by Writing a 1)
Set when lockup has been detected on the DMA transmit path. Writing a one clears this bit.

Bit 9 – TXMACLCK Transmit MAC Lockup (Clear by Writing a 1)
Set when lockup has been detected on the MAC transmit path. Writing a one clears this bit.

Bit 8 – HRESP System Bus Response (Clear by Writing a 1)
Set when the DMA block sees a system bus error. Writing a one clears this bit.

Bit 7 – LCO Late Collision Occurred (Clear by Writing a 1)
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 (Clear by Writing a 1)
Set when a frame has been transmitted. Writing a one clears this bit.

Bit 4 – TFC Transmit Frame Corruption Due to System Bus Error (Clear by Writing a 1)
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 a single frame is too large for the 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 (Clear by Writing a 1)
Writing a one clears this bit.

Bit 1 – COL Collision Occurred (Clear by Writing a 1)
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 (Clear by Writing a 1)
Set when a transmit buffer descriptor is read with its used bit set. Writing a one clears this bit.

62.8.170.GMAC Express EMAC Receive Buffer Queue Base Address Register

Name: GMAC_EMAC_RBQB
Offset: 0x1018
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.

The descriptors must be aligned at 32-bit boundaries and the descriptors are written to using two single accesses.

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.

62.8.171.GMAC Express MAC Transmit Buffer Queue Base Address Register

Name: GMAC_EMAC_TBQB
Offset: 0x101C
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.

The descriptors must be aligned at 32-bit boundaries and the descriptors are read from memory using two individual non sequential accesses.

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.

62.8.172.GMAC Express EMAC Receive Status Register

Name: GMAC_EMAC_RSR
Offset: 0x1020
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
			RXDMALCK	RXMACLCK	HNO	RXOVR	REC	BNA
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit 5 – RXDMALCK Receive DMA Lockup (Clear by Writing a 1)
Set when lockup has been detected on the DMA receive path. Writing a one clears this bit.

Bit 4 – RXMACLCK Receive MAC Lockup (Clear by Writing a 1)
Set when lockup has been detected on the MAC receive path. Writing a one clears this bit.

Bit 3 – HNO System Bus Error (Clear by Writing a 1)
Set when the DMA block sees a system bus error. Writing a one clears this bit.

Bit 2 – RXOVR Receive Overrun (Clear by Writing a 1)
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 (Clear by Writing a 1)
One or more frames have been received and placed in memory. Writing a one clears this bit.

Bit 0 – BNA Buffer Not Available (Clear by Writing a 1)
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.

62.8.173.GMAC Express EMAC Interrupt Status Register

Name: GMAC_EMAC_ISR
Offset: 0x1024
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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
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
	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 31 – TXLCK Transmit Path Locked (Cleared on read)
Set when any monitors (MAC or DMA) detect a lockup.

Bit 30 – RXLCK Receive Path Locked (Cleared on read)
Set when any monitors (MAC or DMA) detect a lockup.

Bit 29 – TSUTIMCOMP TSU Timer Comparison (Cleared on read)
Indicates when the TSU timer count value is equal to the programmed value.

Bit 28 – WOL Wake On LAN (Cleared on read)
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 (Cleared on read)
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.

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 a 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 writeback has not completed when another status writeback 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. Cleared on read.

Bit 0 – MFS Management Frame Sent (Cleared on read)
The PHY Maintenance register has completed its operation.

62.8.174.GMAC Express EMAC Interrupt Enable Register

Name: GMAC_EMAC_IER
Offset: 0x1028
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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access	W	W	W	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

Bit 29 – TSUTIMCOMP TSU Timer Comparison

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

62.8.175.GMAC Express EMAC Interrupt Disable Register

Name: GMAC_EMAC_IDR
Offset: 0x102C
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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
Access	W	W	W	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

Bit 29 – TSUTIMCOMP TSU Timer Comparison

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

62.8.176.GMAC Express EMAC Interrupt Mask Register

Name: GMAC_EMAC_IMR
Offset: 0x1030
Reset: 0xFFFFFFFF
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
	TXLCK	RXLCK	TSUTIMCOMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
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
	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 31 – TXLCK Transmit Path Lockup Detected

Bit 30 – RXLCK Receive Path Lockup Detected

Bit 29 – TSUTIMCOMP TSU Timer Comparison

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

62.8.177.GMAC Express EMAC Receive Pause Quantum Register

Name: GMAC_EMAC_RPQ
Offset: 0x1038
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.

62.8.178.GMAC Express EMAC Transmit Pause Quantum Register

Name: GMAC_EMAC_TPQ
Offset: 0x103C
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.

62.8.179.GMAC Express MAC TX Partial Store and Forward Register

Name: GMAC_EMAC_TPSF
Offset: 0x1040
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[9:8]	
Access							R/W	R/W
Reset							1	1

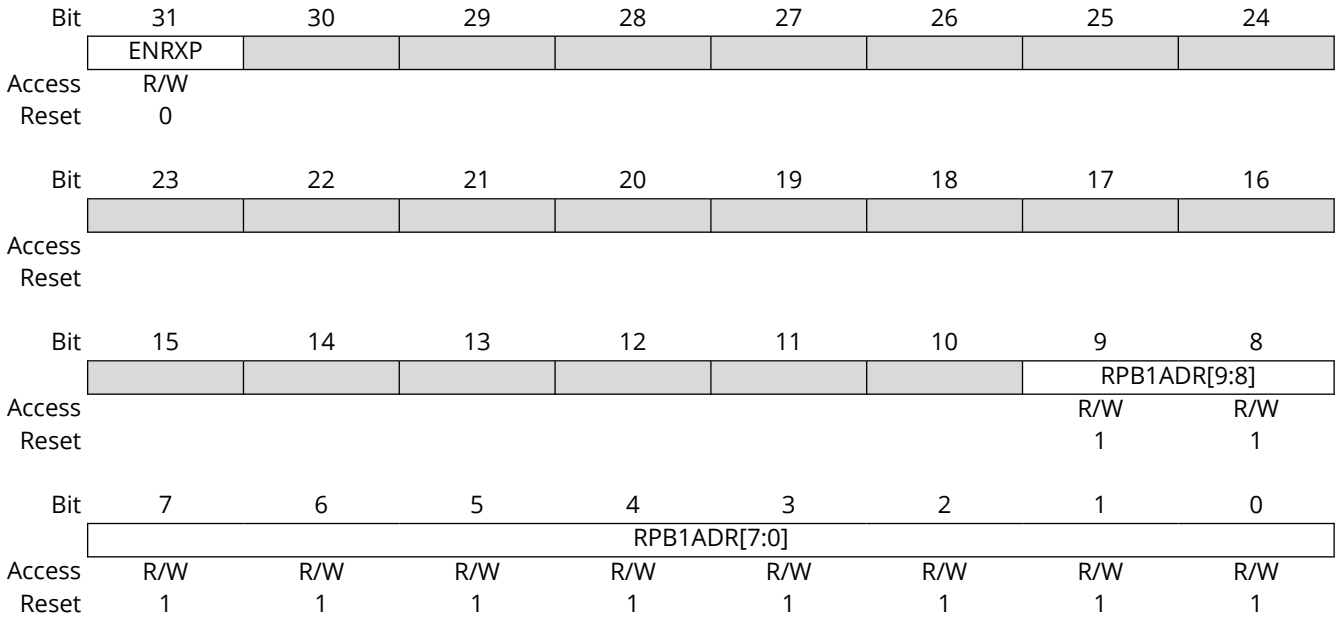
Bit	7	6	5	4	3	2	1	0
	TPB1ADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – ENTXP Enable TX Partial Store and Forward Operation

Bits 9:0 – TPB1ADR[9:0] Transmit Partial Store and Forward Address Watermark value. Reset = 1.

62.8.180.GMAC Express MAC RX Partial Store and Forward Register

Name: GMAC_EMAC_RPSF
Offset: 0x1044
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[9:8]	
Access							R/W	R/W
Reset							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 9:0 – RPB1ADR[9:0] Receive Partial Store and Forward Address Watermark value. Reset = 1.

62.8.181.GMAC Express MAC RX Jumbo Frame Max Length Register

Name: GMAC_EMAC_RJFML
Offset: 0x1048
Reset: 0x00002800
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	0	1	0	0	0

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	0	0	0	0	0	0	0	0

Bits 13:0 – FML[13:0] Frame Max Length
Rx jumbo frame maximum length.

62.8.182.GMAC Express MAC System Bus Max Pipeline Register

Name: GMAC_EMAC_AMP
Offset: 0x1054
Reset: 0x00000101
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	1

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	1

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.

62.8.183.GMAC Express MAC Interrupt Moderation Register

Name: GMAC_EMAC_INTM
Offset: 0x105C
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_EMAC_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_EMAC_ISR.RCOMP. A non-zero value indicates receive interrupt moderation will be performed.

62.8.184.GMAC Express MAC System Wake-Up Time Register

Name: GMAC_EMAC_SYSWT
Offset: 0x1060
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 the Interrupt registers (each interval is equivalent to eight GTXCLK periods and varies with data rate).

62.8.185.GMAC Express EMAC Lockup Configuration Register

Name: GMAC_EMAC_LCKUP_CFGR
Offset: 0x1068
Reset: 0x07FFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	TXDMA_LCKUP_EN	TXMAC_LCKUP_EN	RXDMA_LCKUP_EN	RXMAC_LCKUP_EN	LCKUP_REC_EN	DMA_LOCKUP_TIME[10:8]		
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	1

Bit	23	22	21	20	19	18	17	16
	DMA_LOCKUP_TIME[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
	PRESCALER[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
	PRESCALER[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 – TXDMA_LCKUP_EN Transmit DMA Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the transmit DMA.
1	Enables the monitor that detects lockups in the transmit DMA.

Bit 30 – TXMAC_LCKUP_EN Transmit MAC Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the transmit MAC.
1	Enables the monitor that detects lockups in the transmit MAC.

Bit 29 – RXDMA_LCKUP_EN Receive DMA Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the receive DMA.
1	Enables the monitor that detects lockups in the receive DMA.

Bit 28 – RXMAC_LCKUP_EN Receive MAC Lockup Detector Enable

Value	Description
0	Disables the monitor that detects lockups in the receive MAC.
1	Enables the monitor that detects lockups in the receive MAC.

Bit 27 – LCKUP_REC_EN Lockup Recovery Enable

Value	Description
0	No effect.
1	Forces the EMAC in Reset mode when a lockup.is detected on the transmit or receive data paths.

Bits 26:16 – DMA_LOCKUP_TIME[10:0] Timeout Value for Receive and Transmit DMA
Defines the timeout value for receive and transmit DMA lockup detection defined as a multiple of the prescaler value (PRESCALER). The MAC lockup time is defined in a separate configuration register (GMAC_LCKUP_TIME).

Bits 15:0 – PRESCALER[15:0] Prescaler Value for Timeout
Defines the prescaler value which is a multiple of the transmit clock.

62.8.186.GMAC Express MAC Lockup Time Register

Name: GMAC_EMAC_LCKUP_TIME
Offset: 0x106C
Reset: 0x07FFFFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
						TX_MAC_LOCKUP_TIME[10:8]		
Access						R/W	R/W	R/W
Reset						1	1	1

Bit	23	22	21	20	19	18	17	16
	TX_MAC_LOCKUP_TIME[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
	RX_MAC_LOCKUP_TIME[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
	RX_MAC_LOCKUP_TIME[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 26:16 – TX_MAC_LOCKUP_TIME[10:0] Transmit MAC Lockup Detector Time
Defines the timeout value for the transmit MAC lockup detection defined as a multiple of the prescaler value (GMAC_EMAC_LCKUP_CFGR.PRESCALER).

Bits 15:0 – RX_MAC_LOCKUP_TIME[15:0] Receive MAC Lockup Detector Time
Defines the timeout value for the receive MAC lockup detection defined as a multiple of the prescaler value (GMAC_EMAC_LCKUP_CFGR.PRESCALER).

62.8.187.GMAC Express MAC Transmit DMA Lockup Control Register

Name: GMAC_EMAC_TXDMA_LCKUP_CR
Offset: 0x1070
Reset: 0x07FFFFFF
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
			LCKUP_EN_Q5	LCKUP_EN_Q4	LCKUP_EN_Q3	LCKUP_EN_Q2	LCKUP_EN_Q1	LCKUP_EN_Q0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			1	1	1	1	1	1

Bits 0, 1, 2, 3, 4, 5 – LCKUP_EN_Qx Transmit DMA Lockup Detector Enable for Queue x

Value	Description
0	Disables the transmit DMA lockup timer for queue x. The number of outstanding packets are still counted but the actual timer does not run.
1	Enables the transmit DMA lockup timer for queue x.

62.8.188.GMAC Express MAC Receive Watermark Register

Name: GMAC_EMAC_RX_WATERMARK
Offset: 0x107C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RX_LOW_WATERMARK[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
	RX_LOW_WATERMARK[7:0]							
Access	R/W	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
	RX_HIGH_WATERMARK[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
	RX_HIGH_WATERMARK[7:0]							
Access	R/W	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 – RX_LOW_WATERMARK[15:0] Receive Watermark Low Threshold
Defines the low threshold of the receive memory for which a zero length pause frame is transmitted when the memory fill level falls below the threshold (the action is performed only if the last pause frame transmitted was non-zero).

Bits 15:0 – RX_HIGH_WATERMARK[15:0] Receive Watermark High Threshold
Defines the high threshold of the receive memory for which a zero length pause frame is transmitted when the memory fill level exceeds the threshold.

62.8.189.GMAC Express MAC Hash Register Bottom

Name: GMAC_EMAC_HRB
Offset: 0x1080
Reset: 0x00000000
Property: Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the 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.

62.8.190.GMAC Express MAC Hash Register Top

Name: GMAC_EMAC_HRT
Offset: 0x1084
Reset: 0x00000000
Property: Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the 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.

62.8.191.GMAC Express MAC Specific Address Bottom Register x

Name: GMAC_EMAC_SABx
Offset: 0x1088 + x*0x08 [x=0..3]
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
Least significant 32 bits of the destination address, that is, bits 31:0. Bit 0 indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.192.GMAC Express MAC Specific Address Top Register x

Name: GMAC_EMAC_SATx
Offset: 0x108C + x*0x08 [x=0..3]
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

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
The most significant bits of the destination address, that is, bits 47:32.

62.8.193.GMAC Express MAC Type ID Match 1 Register

Name: GMAC_EMAC_TIDM1
Offset: 0x10A8
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.

62.8.194.GMAC Express EMAC Type ID Match 2 Register

Name: GMAC_EMAC_TIDM2
Offset: 0x10AC
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.

62.8.195.GMAC Express MAC Type ID Match 3 Register

Name: GMAC_EMAC_TIDM3
Offset: 0x10B0
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.

62.8.196.GMAC Express MAC Type ID Match 4 Register

Name: GMAC_EMAC_TIDM4
Offset: 0x10B4
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.

62.8.197.GMAC Express MAC Wake on LAN Register

Name: GMAC_EMAC_WOL
Offset: 0x10B8
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.

62.8.198.GMAC Express MAC IPG Stretch Register

Name: GMAC_EMAC_IPGS
Offset: 0x10BC
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 interpacket gap. 1 is added to bits 15:8 to prevent a divide by zero. See MAC Transmit Block.

62.8.199.GMAC Express MAC Stacked VLAN Register

Name: GMAC_EMAC_SVLAN
Offset: 0x10C0
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.

62.8.200.GMAC Express MAC Transmit PFC Pause Register

Name: GMAC_EMAC_TPFCP
Offset: 0x10C4
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 ‘1’ 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 ‘1’ 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].

62.8.201.GMAC Express MAC Specific Address 1 Mask Bottom Register

Name: GMAC_EMAC_SAMB1
Offset: 0x10C8
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 ‘1’ masks the corresponding bit in the Specific Address 1 register.

62.8.202.GMAC Express MAC Specific Address Mask 1 Top Register

Name: GMAC_EMAC_SAMT1
Offset: 0x10CC
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 ‘1’ masks the corresponding bit in the Specific Address 1 register.

62.8.203.Address Express MAC Mask for RX Data Buffer Accesses Register

Name: GMAC_EMAC_AMRX
Offset: 0x10D0
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 will be ignored while the DMA is currently processing a receive packet. It will only affect 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 cleared, the associated value stored in bits 31:28 have no effect on the address used for receive data buffer accesses.
Any changes to this register will be ignored while the DMA is currently processing a receive packet. It will only affect the next full packet to be written to external memory.

62.8.204.PTP Express MAC RX Unicast IP Destination Address Register

Name: GMAC_EMAC_RXUDAR
Offset: 0x10D4
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.

62.8.205.PTP Express MAC TX Unicast IP Destination Address Register

Name: GMAC_EMAC_TXUDAR
Offset: 0x10D8
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.

62.8.206.GMAC Express MAC 1588 Timer Nanosecond Comparison Register

Name: GMAC_EMAC_NSC
Offset: 0x10DC
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).

62.8.207.GMAC Express EMAC 1588 Timer Second Comparison Low Register

Name: GMAC_EMAC_SCL
Offset: 0x10E0
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.

62.8.208.GMAC Express MAC 1588 Timer Second Comparison High Register

Name: GMAC_EMAC_SCH
Offset: 0x10E4
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.

62.8.209.GMAC Express MAC PTP Event Frame Transmitted Seconds High Register

Name: GMAC_EMAC_EFTSH
Offset: 0x10E8
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.

62.8.210.GMAC Express MAC PTP Event Frame Received Seconds High Register

Name: GMAC_EMAC_EFRSH
Offset: 0x10EC
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.

62.8.211.GMAC Express MAC PTP Peer Event Frame Transmitted Seconds High Register

Name: GMAC_EMAC_PEFTSH
Offset: 0x10F0
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.

62.8.212.GMAC Express MAC PTP Peer Event Frame Received Seconds High Register

Name: GMAC_EMAC_PEFRSH
Offset: 0x10F4
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.

62.8.213.GMAC Express MAC Octets Transmitted Low Register

Name: GMAC_EMAC_OTLO
Offset: 0x1100
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.

62.8.214.GMAC Express MAC Octets Transmitted High Register

Name: GMAC_EMAC_OTHI
Offset: 0x1104
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.

62.8.215.GMAC Express MAC Frames Transmitted Register

Name: GMAC_EMAC_FT
Offset: 0x1108
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. Counts the number of frames successfully transmitted, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.216.GMAC Express MAC Broadcast Frames Transmitted Register

Name: GMAC_EMAC_BCFT
Offset: 0x110C
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. Counts the number of broadcast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.217.GMAC Express MAC Multicast Frames Transmitted Register

Name: GMAC_EMAC_MFT
Offset: 0x1110
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
Counts the number of multicast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.218.GMAC Express MAC Pause Frames Transmitted Register

Name: GMAC_EMAC_PFT
Offset: 0x1114
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
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.

62.8.219.GMAC Express MAC 64 Byte Frames Transmitted Register

Name: GMAC_EMAC_BFT64
Offset: 0x1118
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
Counts the number of 64 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.

62.8.220.GMAC Express MAC 65 to 127 Byte Frames Transmitted Register

Name: GMAC_EMAC_TBFT127
Offset: 0x111C
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
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.

62.8.221.GMAC Express MAC 128 to 255 Byte Frames Transmitted Register

Name: GMAC_EMAC_TBFT255
Offset: 0x1120
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
Counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.222.GMAC Express MAC 256 to 511 Byte Frames Transmitted Register

Name: GMAC_EMAC_TBFT511
Offset: 0x1124
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
Counts the number of 256 to 511 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.223.GMAC Express MAC 512 to 1023 Byte Frames Transmitted Register

Name: GMAC_EMAC_TBFT1023
Offset: 0x1128
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
Counts the number of 512 to 1023 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.224.GMAC Express MAC 1024 to 1518 Byte Frames Transmitted Register

Name: GMAC_EMAC_TBFT1518
Offset: 0x112C
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
Counts the number of 1024 to 1518 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

62.8.225.GMAC Express MAC Greater Than 1518 Byte Frames Transmitted Register

Name: GMAC_EMAC_GTBFT1518
Offset: 0x1130
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
Counts the number of 1518 or above byte frames successfully transmitted without error i.e., no underrun and not too many retries.

62.8.226.GMAC Express MAC Transmit Underruns Register

Name: GMAC_EMAC_TUR
Offset: 0x1134
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
Counts the number of frames not transmitted due to a transmit underrun. If this register is incremented, no other statistics register is incremented.

62.8.227.GMAC Express MAC Single Collision Frames Register

Name: GMAC_EMAC_SCF
Offset: 0x1138
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
Counts the number of frames experiencing a single collision before being successfully transmitted i.e., no underrun.

62.8.228.GMAC Express MAC Multiple Collision Frames Register

Name: GMAC_EMAC_MCF
Offset: 0x113C
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
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.

62.8.229.GMAC Express MAC Excessive Collisions Register

Name: GMAC_EMAC_EC
Offset: 0x1140
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
Counts the number of frames that failed to be transmitted because they experienced 16 collisions.

62.8.230.GMAC Express MAC Late Collisions Register

Name: GMAC_EMAC_LC
Offset: 0x1144
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.

62.8.231.GMAC Express MAC Deferred Transmission Frames Register

Name: GMAC_EMAC_DTF
Offset: 0x1148
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
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.

62.8.232.GMAC Express MAC Carrier Sense Errors Register

Name: GMAC_EMAC_CSE
Offset: 0x114C
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

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.

62.8.233.GMAC Express MAC Octets Received Low Register

Name: GMAC_EMAC_ORLO
Offset: 0x1150
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.

62.8.234.GMAC Express MAC Octets Received High Register

Name: GMAC_EMAC_ORHI
Offset: 0x1154
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.

62.8.235.GMAC Express MAC Frames Received Register

Name: GMAC_EMAC_FR
Offset: 0x1158
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. Counts the number of frames successfully received. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

62.8.236.GMAC Express MAC Broadcast Frames Received Register

Name: GMAC_EMAC_BCFR
Offset: 0x115C
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. 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.

62.8.237.GMAC Express MAC Multicast Frames Received Register

Name: GMAC_EMAC_MFR
Offset: 0x1160
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
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.

62.8.238.GMAC Express MAC Pause Frames Received Register

Name: GMAC_EMAC_PFR
Offset: 0x1164
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
Counts the number of pause frames received without error.

62.8.239.GMAC Express MAC 64 Byte Frames Received Register

Name: GMAC_EMAC_BFR64
Offset: 0x1168
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
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.

62.8.240.GMAC Express MAC 65 to 127 Byte Frames Received Register

Name: GMAC_EMAC_TBFR127
Offset: 0x116C
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
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.

62.8.241.GMAC Express MAC 128 to 255 Byte Frames Received Register

Name: GMAC_EMAC_TBFR255
Offset: 0x1170
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
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.

62.8.242.GMAC Express MAC 256 to 511 Byte Frames Received Register

Name: GMAC_EMAC_TBFR511
Offset: 0x1174
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
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.

62.8.243.GMAC Express MAC 512 to 1023 Byte Frames Received Register

Name: GMAC_EMAC_TBFR1023
Offset: 0x1178
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
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.

62.8.244.GMAC Express MAC 1024 to 1518 Byte Frames Received Register

Name: GMAC_EMAC_TBFR1518
Offset: 0x117C
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
Counts the number of 1024 to 1518 byte frames successfully received without error, i.e., no underrun and not too many retries.

62.8.245.GMAC Express MAC 1519 to Maximum Byte Frames Received Register

Name: GMAC_EMAC_TMXBFR
Offset: 0x1180
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
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. SeeGMAC_NCFGR.

62.8.246.GMAC Express MAC Undersized Frames Received Register

Name: GMAC_EMAC_UFR
Offset: 0x1184
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, 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.

62.8.247.GMAC Express MAC Oversized Frames Received Register

Name: GMAC_EMAC_OFR
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
							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
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.

62.8.248.GMAC Express MAC Jabbers Received Register

Name: GMAC_EMAC_JR
Offset: 0x118C
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.

62.8.249.GMAC Express MAC Frame Check Sequence Errors Register

Name: GMAC_EMAC_FCSE
Offset: 0x1190
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
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.

62.8.250.GMAC Express MAC Length Field Frame Errors Register

Name: GMAC_EMAC_LFFE
Offset: 0x1194
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
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.

62.8.251.GMAC Express MAC Receive Symbol Errors Register

Name: GMAC_EMAC_RSE
Offset: 0x1198
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

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.

62.8.252.GMAC Express MAC Alignment Errors Register

Name: GMAC_EMAC_AE
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
							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
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.

62.8.253.GMAC Express MAC Receive Resource Errors Register

Name: GMAC_EMAC_RRE
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
							RXRER[17:16]	
Access							R	R
Reset							0	0

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

Bits 17:0 – RXRER[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 Express MAC reads a buffer descriptor with its ownership (or used) bit set.

62.8.254.GMAC Express MAC Receive Overruns Register

Name: GMAC_EMAC_ROE
Offset: 0x11A4
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
Counts the number of frames that are address recognized but were not copied to memory due to a receive overrun.

62.8.255.GMAC Express MAC IP Header Checksum Errors Register

Name: GMAC_EMAC_IHCE
Offset: 0x11A8
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
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, 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.

62.8.256.GMAC Express MAC TCP Checksum Errors Register

Name: GMAC_EMAC_TCE
Offset: 0x11AC
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
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, 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.

62.8.257.GMAC Express MAC UDP Checksum Errors Register

Name: GMAC_EMAC_UCE
Offset: 0x11B0
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
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, 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.

62.8.258.GMAC Express MAC Flushed Received Packets Count Register

Name: GMAC_EMAC_FLRXPCR
Offset: 0x11B4
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.

62.8.259.GMAC Express MAC 1588 Timer Increment Sub-nanoseconds Register

Name: GMAC_EMAC_TISUBN
Offset: 0x11BC
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 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.

62.8.260.GMAC Express MAC 1588 Timer Seconds High Register

Name: GMAC_EMAC_TSH
Offset: 0x11C0
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.

62.8.261.GMAC Express MAC 1588 Timer Seconds Low Register

Name: GMAC_EMAC_TSL
Offset: 0x11D0
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.

62.8.262.GMAC Express MAC 1588 Timer Nanoseconds Register

Name: GMAC_EMAC_TN
Offset: 0x11D4
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.

62.8.263.GMAC Express MAC 1588 Timer Adjust Register

Name: GMAC_EMAC_TA
Offset: 0x11D8
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 ‘1’ to subtract from the 1588 timer. Write as ‘0’ 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.

62.8.264.GMAC Express MAC 1588 Timer Increment Register

Name: GMAC_EMAC_TI
Offset: 0x11DC
Reset: 0x0000_0000
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

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.

62.8.265.GMAC Express MAC PTP Event Frame Transmitted Seconds Low Register

Name: GMAC_EMAC_EFTSL
Offset: 0x11E0
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
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.

62.8.266.GMAC Express MAC PTP Event Frame Transmitted Nanoseconds Register

Name: GMAC_EMAC_EFTN
Offset: 0x11E4
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
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.

62.8.267.GMAC Express MAC PTP Event Frame Received Seconds Low Register

Name: GMAC_EMAC_EFRSL
Offset: 0x11E8
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
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.

62.8.268.GMAC Express MAC PTP Event Frame Received Nanoseconds Register

Name: GMAC_EMAC_EFRN
Offset: 0x11EC
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
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.

62.8.269.GMAC Express MAC PTP Peer Event Frame Transmitted Seconds Low Register

Name: GMAC_EMAC_PEFTSL
Offset: 0x11F0
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
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.

62.8.270.GMAC Express MAC PTP Peer Event Frame Transmitted Nanoseconds Register

Name: GMAC_EMAC_PEFTN
Offset: 0x11F4
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
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.

62.8.271.GMAC Express MAC PTP Peer Event Frame Received Seconds Low Register

Name: GMAC_EMAC_PEFRSL
Offset: 0x11F8
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
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.

62.8.272.GMAC Express MAC PTP Peer Event Frame Received Nanoseconds Register

Name: GMAC_EMAC_PEFRN
Offset: 0x11FC
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
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.

62.8.273.GMAC Express MAC Transmit Pause Quantum 1 Register

Name: GMAC_EMAC_TXPQUANT1
Offset: 0x1260
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.

62.8.274.GMAC Express MAC Transmit Pause Quantum 2 Register

Name: GMAC_EMAC_TXPQUANT2
Offset: 0x1264
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.

62.8.275.GMAC Express MAC Transmit Pause Quantum 3 Register

Name: GMAC_EMAC_TXPQUANT3
Offset: 0x1268
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.

62.8.276.GMAC Express MAC Received LPI Transitions

Name: GMAC_EMAC_RXLPI
Offset: 0x1270
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.

62.8.277.GMAC Express MAC Received LPI Time

Name: GMAC_EMAC_RXLPITIME
Offset: 0x1274
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)
Increments once every 16 MCK cycles when the bit LPI Indication (bit 7) is set in the Network Status register.

62.8.278.GMAC Express MAC Transmit LPI Transitions

Name: GMAC_EMAC_TXLPI
Offset: 0x1278
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.

62.8.279.GMAC Express MAC Transmit LPI Time

Name: GMAC_EMAC_TXLPITIME
Offset: 0x127C
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)
Increments once every 16 MCK cycles when the bit Enable LPI Transmission (bit 19) is set in the Network Control register.

62.8.280.GMAC Express MAC Quality of Service Configuration Register

Name: GMAC_EMAC_QOS_CFG
Offset: 0x12E0
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.

62.8.281.GMAC Express MAC Additional Specific Address Bottom Register x

Name: GMAC_EMAC_ASABx
Offset: 0x1300 + x*0x08 [x=0..31]
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
Least significant 32 bits of the destination address, that is, bits 31:0. Bit 0 indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.

62.8.282.GMAC Express MAC Additional Specific Address Top Register x

Name: GMAC_EMAC_ASATx
Offset: 0x1304 + x*0x08 [x=0..31]
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

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
The most significant bits of the destination address, that is, bits 47:32.

62.8.283.GMAC Express MAC Credit-Based Shaping Control Register

Name: GMAC_EMAC_CBSCR
Offset: 0x14BC
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.

62.8.284.GMAC Express MAC Credit-Based Shaping IdleSlope Register for Queue A

Name: GMAC_EMAC_CBSISQA
Offset: 0x14C0
Reset: 0x00000000
Property: Read/Write

Credit-based shaping must be disabled in GMAC_EMAC_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.

62.8.285.GMAC Express MAC Credit-Based Shaping IdleSlope Register for Queue B

Name: GMAC_EMAC_CBSISQB
Offset: 0x14C4
Reset: 0x00000000
Property: Read/Write

Credit-based shaping must be disabled in GMAC_EMAC_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/s = 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.

62.8.286.GMAC Express MAC Transmit Queue Upper Buffer Address Register

Name: GMAC_EMAC_TQUBA
Offset: 0x14C8
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.

62.8.287.GMAC Express MAC Transmit Buffer Data Control Register

Name: GMAC_EMAC_TXBDCTRL
Offset: 0x14CC
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								

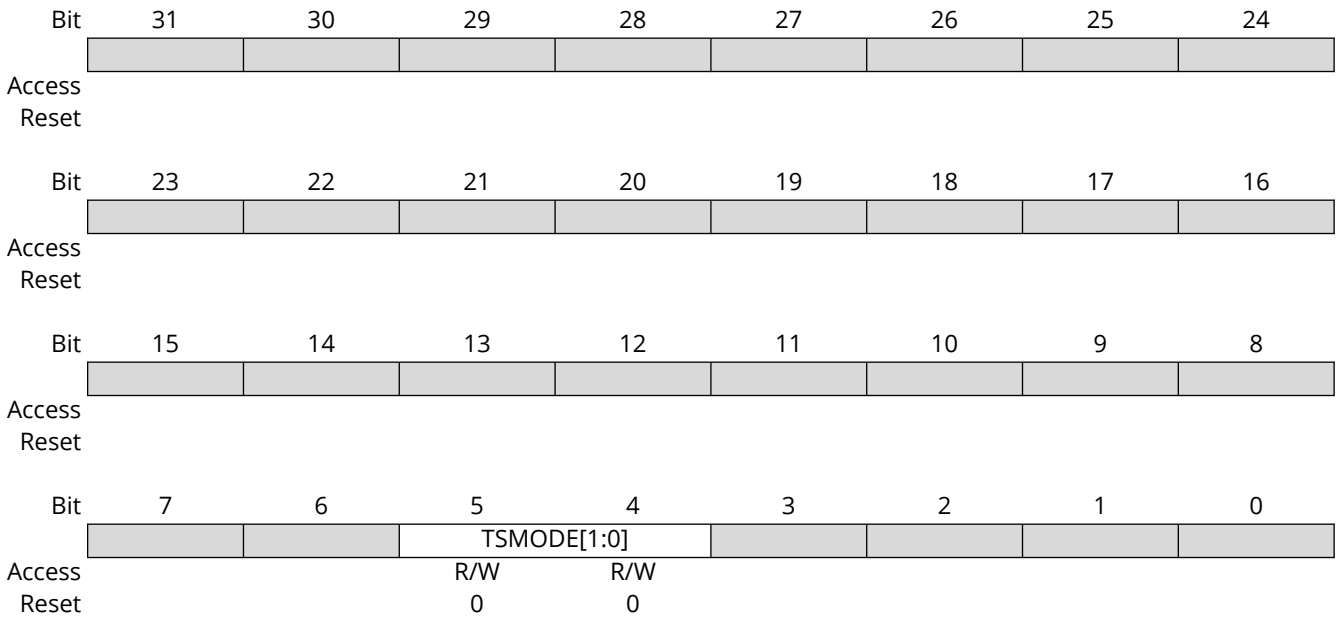
Bit	7	6	5	4	3	2	1	0
			TSMODE[1:0]					
Access			R/W	R/W				
Reset			0	0				

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.

62.8.288.GMAC Express MAC Receive Buffer Data Control Register

Name: GMAC_EMAC_RXBDCTRL
Offset: 0x14D0
Reset: 0x00000000
Property: Read/Write



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.

62.8.289.GMAC Express MAC Receive Queue Upper Buffer Address Register

Name: GMAC_EMAC_RQUBA
Offset: 0x14D4
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.

62.8.290.GMAC Express MAC Screening Type 1 Register

Name: GMAC_EMAC_ST1R
Offset: 0x1500
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.

62.8.291.GMAC Express MAC Screening Type 2 Register x

Name: GMAC_EMAC_ST2Rx
Offset: 0x1540 + x*0x04 [x=0..1]
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_EMAC_ST2CW0R and GMAC_EMAC_ST2CW1R. 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_EMAC_ST2CW0R and GMAC_EMAC_ST2CW1R. 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.

Value	Description
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_EMAC_ST2CW0R and GMAC_EMAC_ST2CW1R. 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.

62.8.292.GMAC Express MAC Transmit Schedule Control Register

Name: GMAC_EMAC_TSCTL
Offset: 0x1580
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								

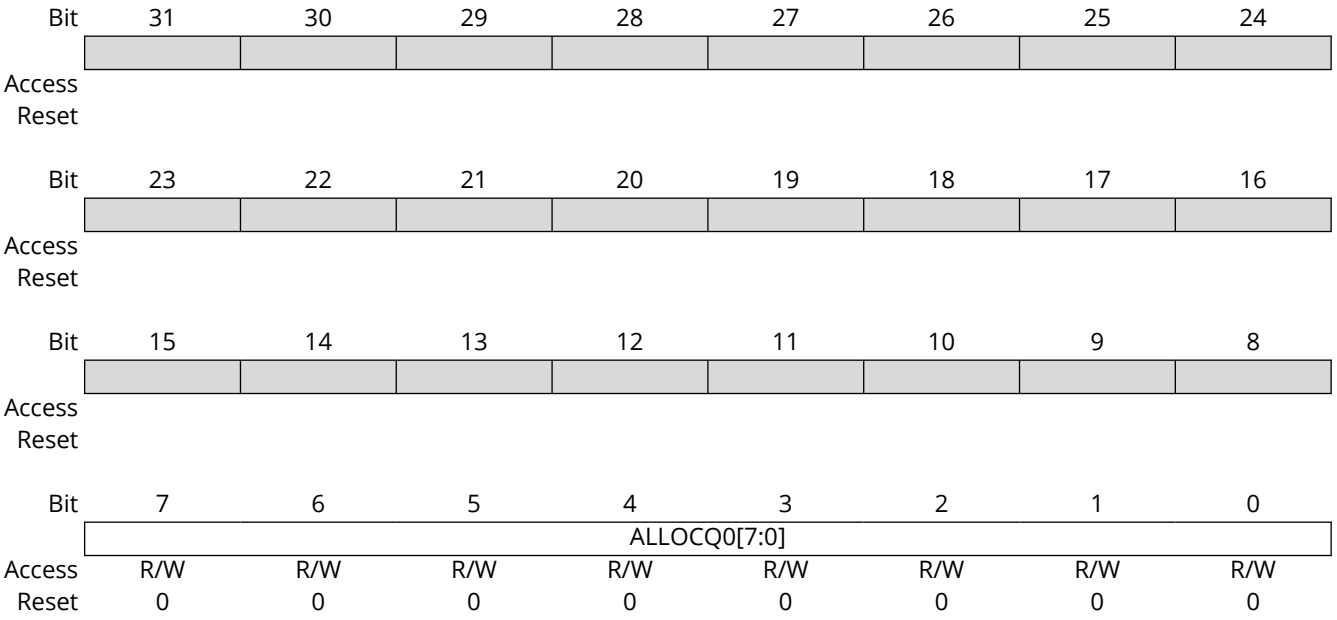
Bit	7	6	5	4	3	2	1	0
							TXSQ[1:0]	
Access							R/W	R/W
Reset							0	0

Bits 1:0 – TXSQ[1:0] Transmit Schedule for Q0

Value	Name	Description
0	FP	Fixed Priority.
1	CBS	CBS Enabled only valid if CBS capability selected.
2	DWRR	DWRR enabled.
3	ETS	ETS enabled.

62.8.293.GMAC Express MAC Transmit Queue Bandwidth Rate Limit Register

Name: GMAC_EMAC_TQBWRL
Offset: 0x1590
Reset: 0
Property: R/W



Bit	31	30	29	28	27	26	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
	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 7:0 – ALLOCQ0[7:0] DWRR Weighting or ETS Bandwidth Allocation for EMAC
Defines the value of Deficit Weighted Round Robin (DWRR) or Enhanced Transmission Selection (ETS - 802.1Qaz).

62.8.294.GMAC Express MAC Transmit Queue Segment Allocation Register

Name: GMAC_EMAC_TQSA
Offset: 0x15A0
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
						SEGALLOCQ0[2:0]		
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:0 – SEGALLOCQ0[2:0] Segment Allocation for EMAC
Number of segments allocated to EMAC. This should be entered as a log 2; for example, entering a value of 2 grants 4 segments. A maximum of 16 segments can be granted.
There is no need to change this register for the EMAC.

62.8.295.GMAC Express MAC Screening Type 2 Compare Word 0 Register x

Name: GMAC_EMAC_ST2CW0Rx
Offset: 0x1700 + x*0x08 [x=0..5]
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.

62.8.296.GMAC Express MAC Screening Type 2 Compare Word 1 Register x

Name: GMAC_EMAC_ST2CW1Rx
Offset: 0x1704 + x*0x08 [x=0..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
							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_EMAC_ST2CW0R contains a 2-byte compare value with a 2-byte mask value or a 4-byte compare value.

Value	Description
0	GMAC_EMAC_ST2CW0R contains a 2-byte compare value with a 2-byte mask value.
1	GMAC_EMAC_ST2CW0R 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.

62.8.297.GMAC Express MAC ENST Start Time Queue Register

Name: GMAC_EMAC_ENST_START
Offset: 0x1800
Reset: 0x0001FFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	START_SEC[1:0]		START_NSEC[29:24]					
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
	START_NSEC[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	1

Bit	15	14	13	12	11	10	9	8
	START_NSEC[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
	START_NSEC[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 31:30 – START_SEC[1:0] Seconds for Start Time
Defines the seconds for the absolute start time of the queue.

Bits 29:0 – START_NSEC[29:0] Nanoseconds for Start Time
Defines the nanoseconds for the absolute start time of the queue.

62.8.298.GMAC Express MAC ENST On Time Queue Register

Name: GMAC_EMAC_ENST_ON
Offset: 0x1820
Reset: 0x0001FFFF
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
								ON_TIME[16]
Access								R/W
Reset								1

Bit	15	14	13	12	11	10	9	8
	ON_TIME[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
	ON_TIME[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 16:0 – ON_TIME[16:0] Time for which the Queue is to be open
Defines the seconds for the time for which the queue is open.

62.8.299.GMAC Express MAC ENST Off Time Queue Register

Name: GMAC_EMAC_ENST_OFF
Offset: 0x1840
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
								OFF_TIME[16]
Access								R/W
Reset								0

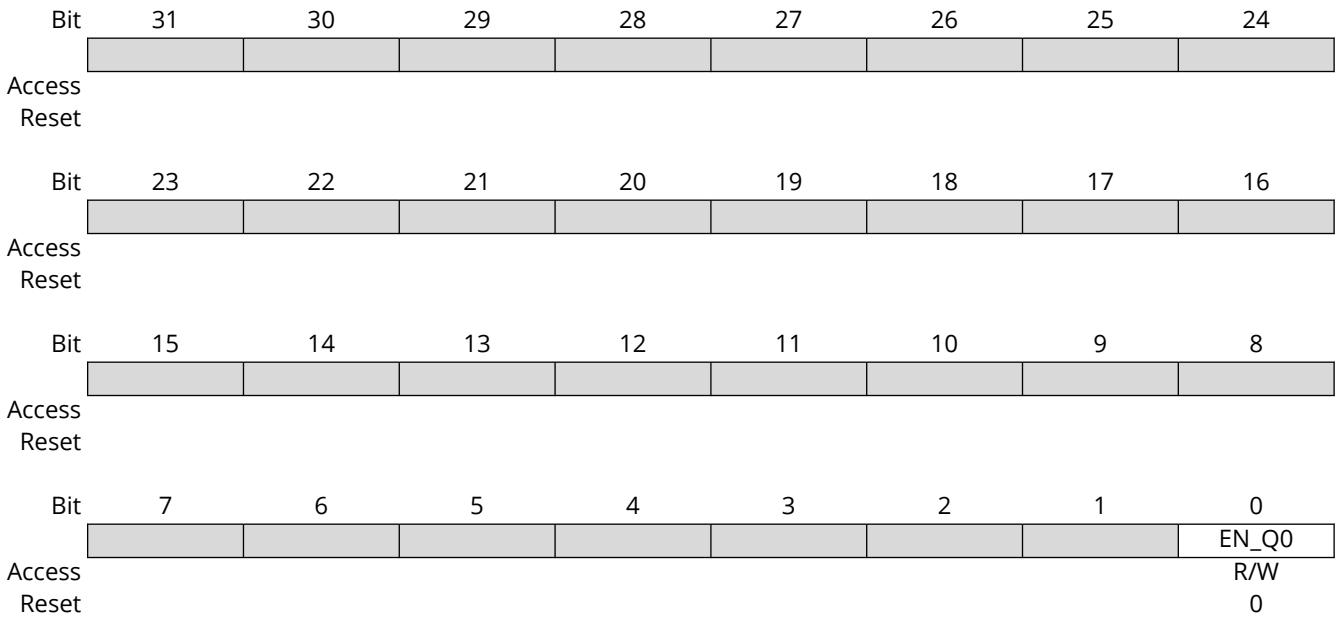
Bit	15	14	13	12	11	10	9	8
	OFF_TIME[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	OFF_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 16:0 – OFF_TIME[16:0] Time for which the Queue is to be Blocked
Defines the seconds for the time for which the queue is blocked.

62.8.300.GMAC Express MAC ENST Control Register

Name: GMAC_EMAC_ENST_CR
Offset: 0x1880
Reset: 0x00000000
Property: Read/Write



Bit 0 – EN_Q0 Enhanced Scheduled Traffic Enable for EMAC

Value	Description
0	Disables the enhanced scheduled traffic for EMAC.
1	Enables the enhanced scheduled traffic for EMAC. EMAC has only 1 queue and ENST is enabled by writing EN_Q0.

62.8.301.GMAC Express MAC Frame Elimination Timeout Register

Name: GMAC_EMAC_FRER_TIMEOUT
Offset: 0x18A0
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
	TIMEOUT[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
	TIMEOUT[7:0]							
Access	R/W	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 – TIMEOUT[15:0] Sequence Recovery Timer Restart Period for Credit Based Streams
Determines when the sequence recovery timers are restarted and used by all of the configured CB streams. It is programmed as a count of 8192 rx_clk periods. 8192 rx_clk periods is 65.536 microseconds at gigabit speed and 327.68 microseconds at 100M speed, this allows a max timeout value of about 4 seconds at gigabit speed.

62.8.302.GMAC Express MAC Frame Elimination Redundancy Tag Register

Name: GMAC_EMAC_FRER_REDTAG
Offset: 0x18A4
Reset: 0x4000F1C1
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	STRIP_R_TAG	SIX_BYTE_TAG						
Access	R/W	R/W						
Reset	0	1						

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RED_TAG[15:8]							
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	1

Bit	7	6	5	4	3	2	1	0
	RED_TAG[7:0]							
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	1

Bit 31 – STRIP_R_TAG Stripping Redundancy Tag Enable

Value	Description
0	Disables the stripping function. When the statistics counters need to reflect the actual number of octets received, then the stripping functionality must be disabled.
1	Enables the stripping function, the receive octet counters reflect post deletion frame size so the frame elimination functionality is transparent to higher level management.

Bit 30 – SIX_BYTE_TAG Six-byte Tag Enable

Value	Description
0	Defines a four-byte tag as per 802.1CB standard revision 2.4 and earlier.
1	Enables the six-byte tag as per 802.1CB standard revision 2.4 and later.

Bits 15:0 – RED_TAG[15:0] Redundancy Tag (R-TAG)
Defines the Ethertype value used to identify the redundancy tag (R-TAG).

62.8.303.GMAC Express MAC Frame Elimination Control1 A Register

Name: GMAC_EMAC_FRER_CTRL1_A
Offset: 0x18C0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
								OFFSET_VALUE[8]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OFFSET_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

Bit	7	6	5	4	3	2	1	0
	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[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 – EN_ELIMINATION 802.1CB Elimination of Received Frames Enable

Value	Description
0	Disables the elimination of received frames.
1	Enables the elimination of received frames.

Bit 30 – EN_VECTOR_REC_ALG 802.1CB Vector Recovery Algorithm Enable

Value	Description
0	Enables the match recovery algorithm.
1	Enables the vector recovery algorithm.

Bit 29 – EN_SEQRECRST_TIMER 802.1CB Sequence Recovery Reset Timer Enable

Value	Description
0	Disables the sequence recovery reset timer.
1	Enables the sequence recovery reset timer.

Bit 28 – USE_R_TAG Redundancy Tag Enable

Value	Description
0	Identifies bottom of sequence number with OFFSET_VALUE.
1	Identifies sequence number with redundancy tag.

Bits 16:8 – OFFSET_VALUE[8:0] Offset in Bytes from Start Packet Delimiter to MSB for 802.1CB Sequence Number

Defines the offset value in bytes from the start packet delimiter to the most significant byte of the 802.1CB sequence number. 9-bit width allows a TCP sequence number to be used with IPv6. This value must only be changed when EN_ELIMINATION=0.

Bits 7:4 – MEMBER_STREAM_2[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

Bits 3:0 – MEMBER_STREAM_1[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

62.8.304.GMAC Express MAC Frame Elimination Control1 B Register

Name: GMAC_EMAC_FRER_CTRL1_B
Offset: 0x18C4
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
				SEQ_NUM_LENGTH[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
			SEQ_REC_WINDOW[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 12:8 – SEQ_NUM_LENGTH[4:0] Number of Significant Bits of the 802.1CB Sequence Number
Defines the number of LSBs to consider in the 802.1CB sequence number.
The vector recovery algorithm does not work for values lower than 4. The value 0 and values greater than 16 are equivalent to 16.

Bits 5:0 – SEQ_REC_WINDOW[5:0] Vector Recovery Window
Defines the window size used by the vector recovery algorithm to determine whether to reject a packet. A value of zero means the entire history vector is used.

62.8.305.GMAC Express MAC Frame Elimination Statistics1 A Register

Name: GMAC_EMAC_FRER_STAT1_A
Offset: 0x18C8
Reset: 0x00000000
Property: Read/Write

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
							VEC_REC_ROGUE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VEC_REC_ROGUE[7:0]							
Access	R/W	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
							LATENT_ERRS[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LATENT_ERRS[7:0]							
Access	R/W	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 – VEC_REC_ROGUE[9:0] Number of Dropped Frames (Clear on read)
Returns the number of frames dropped by the vector recovery algorithm for being out of range.

Bits 9:0 – LATENT_ERRS[9:0] Number of Sequence Numbers Seen Without a Duplicate (Clear on read)
Returns the number of sequence numbers seen without a duplicate.
The count is updated when a frame is dropped from the history vector. So the update only happens after a new frame is received.

62.8.306.GMAC Express MAC Frame Elimination Statistics1 B Register

Name: GMAC_EMAC_FRER_STAT1_B
Offset: 0x18CC
Reset: 0x00000000
Property: Read/Write

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SEQRST_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
							OUT_OF_ORDER[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	OUT_OF_ORDER[7:0]							
Access	R/W	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 – SEQRST_COUNT[7:0] Number of Times the Sequence Recovery Reset Timer Decrements to Zero (Clear on read)
Returns the number of times the sequence recovery reset timer decrements to zero.

Bits 9:0 – OUT_OF_ORDER[9:0] Out of Order Sequence Numbers Received (Clear on read)
Returns of out of order sequence numbers received. Incremented when a frame is accepted but the sequence number is not +1 of the highest stored value.

62.8.307.GMAC Express MAC Frame Elimination Control2 A Register

Name: GMAC_EMAC_FRER_CTRL2_A
Offset: 0x18D0
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	EN_ELIMINATION	EN_VECTOR_REC_ALG	EN_SEQRECRST_TIMER	USE_R_TAG				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
								OFFSET_VALUE[8]
Access								R/W
Reset								0

Bit	15	14	13	12	11	10	9	8
	OFFSET_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

Bit	7	6	5	4	3	2	1	0
	MEMBER_STREAM_2[3:0]				MEMBER_STREAM_1[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 – EN_ELIMINATION 802.1CB Elimination of Received Frames Enable

Value	Description
0	Disables the elimination of received frames.
1	Enables the elimination of received frames.

Bit 30 – EN_VECTOR_REC_ALG 802.1CB Vector Recovery Algorithm Enable

Value	Description
0	Enables the match recovery algorithm.
1	Enables the vector recovery algorithm.

Bit 29 – EN_SEQRECRST_TIMER 802.1CB Sequence Recovery Reset Timer Enable

Value	Description
0	Disables the sequence recovery reset timer.
1	Enables the sequence recovery reset timer.

Bit 28 – USE_R_TAG Redundancy Tag Enable

Value	Description
0	Identifies bottom of sequence number with OFFSET_VALUE.
1	Identifies sequence number with redundancy tag.

Bits 16:8 – OFFSET_VALUE[8:0] Offset in Bytes from Start Packet Delimiter to MSB for 802.1CB Sequence Number

Defines the offset value in bytes from the start packet delimiter to the most significant byte of the 802.1CB sequence number. 9-bit width allows a TCP sequence number to be used with IPv6. This value must only be changed when EN_ELIMINATION=0.

Bits 7:4 – MEMBER_STREAM_2[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

Bits 3:0 – MEMBER_STREAM_1[3:0] Pointer to Screener Type 2 Register

Used for member stream identification. The member stream 1 and 2 values can be programmed to identical values.

62.8.308.GMAC Express MAC Frame Elimination Control2 B Register

Name: GMAC_EMAC_FRER_CTRL2_B
Offset: 0x18D4
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
				SEQ_NUM_LENGTH[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
			SEQ_REC_WINDOW[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 12:8 – SEQ_NUM_LENGTH[4:0] Number of Significant Bits of the 802.1CB Sequence Number
Defines the number of LSBs to consider in the 802.1CB sequence number.
The vector recovery algorithm does not work for values lower than 4. The value 0 and values greater than 16 are equivalent to 16.

Bits 5:0 – SEQ_REC_WINDOW[5:0] Vector Recovery Window
Defines the window size used by the vector recovery algorithm to determine whether to reject a packet. A value of zero means the entire history vector is used.

62.8.309.GMAC Express MAC Frame Elimination Statistics2 A Register

Name: GMAC_EMAC_FRER_STAT2_A
Offset: 0x18D8
Reset: 0x00000000
Property: Read/Write

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
							VEC_REC_ROGUE[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	VEC_REC_ROGUE[7:0]							
Access	R/W	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
							LATENT_ERRS[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LATENT_ERRS[7:0]							
Access	R/W	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 – VEC_REC_ROGUE[9:0] Number of Dropped Frames (Clear on read)
Returns the number of frames dropped by the vector recovery algorithm for being out of range.

Bits 9:0 – LATENT_ERRS[9:0] Number of Sequence Numbers Seen Without a Duplicate (Clear on read)
Returns the number of sequence numbers seen without a duplicate.
The count is updated when a frame is dropped from the history vector. So the update only happens after a new frame is received.

62.8.310.GMAC Express MAC Frame Elimination Statistics2 B Register

Name: GMAC_EMAC_FRER_STAT2_B
Offset: 0x18DC
Reset: 0x00000000
Property: Read/Write

The counters do not roll over.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	SEQRST_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
							OUT_OF_ORDER[9:8]	
Access							R/W	R/W
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	OUT_OF_ORDER[7:0]							
Access	R/W	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 – SEQRST_COUNT[7:0] Number of Times the Sequence Recovery Reset Timer Decrements to Zero (Clear on read)
Returns the number of times the sequence recovery reset timer decrements to zero.

Bits 9:0 – OUT_OF_ORDER[9:0] Out of Order Sequence Numbers Received (Clear on read)
Returns of out of order sequence numbers received. Incremented when a frame is accepted but the sequence number is not +1 of the highest stored value.

62.8.311.GMAC Express MAC Receive Queue Flush Register

Name: GMAC_EMAC_RX_FLUSH_Q
Offset: 0x1B00
Reset: 0x00000000
Property: Read/Write

This register defines the traffic policing mode of operation. Each mode can be set simultaneously with the exception of bits 2 and 3, which are exclusive. If bits 2 and 3 are both set then only bit 3 is treated as active

Bit	31	30	29	28	27	26	25	24
	MAX_VAL[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
	MAX_VAL[7:0]							
Access	R/W	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
					LIMIT_FRAME_SIZE	LIMIT_NUM_BYTES	DROP_ON_RESRC_ERR	DROP_ALL
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 31:16 – MAX_VAL[15:0] Maximum Value for the Received Frame Size or Number of 128-Byte Chunk
Defines the maximum value for the received frame size when the bit LIMIT_FRAME_SIZE=1 or the number of 128 byte chunks of data received for this queue and already stored in the memory of the queue awaiting DMA memory writes when LIMIT_NUM_BYTES=1 and LIMIT_FRAME_SIZE=0.

Bit 3 – LIMIT_FRAME_SIZE Maximum Frame-length Received

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	When set, MAX_VAL indicates the maximum frame-length in bytes that may be received. Frames exceeding this length will be dropped. This traffic policing function is relevant to the 802.1Qci standard which specifies stream filtering based on a maximum service data unit (SDU) size.

Bit 2 – LIMIT_NUM_BYTES Limitation of the Number of 128-Byte Chunk of Data Stored in the Memory of this Queue

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	Limits the number of 128 byte chunks of data received for this queue and already stored in the memory of the queue awaiting DMA memory writes to the value defined in the field MAX_VAL.

Bit 1 – DROP_ON_RESRC_ERR Drop on Resource Error

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	If a free DMA descriptor for this queue cannot be obtained (also referred to as lack of descriptor resource and occurs when the software either cannot free up descriptors quickly enough to meet the receive traffic rate or has deliberately decided not to free any descriptors), all new frames received on this queue will be automatically discarded.

Bit 0 – DROP_ALL Drop All Frames

Value	Name	Description
0	DISABLED	No effect.
1	ENABLED	Drops all frames of this queue.

62.8.312.GMAC Express MAC Screening Type 2 Rate Limit Register 0

Name: GMAC_EMAC_SCR2_RATE_LIMIT0
Offset: 0x1B40
Reset: 0x00000000
Property: Read/Write

This register manages the traffic policing function relevant to the 802.1Qci standard and associated to type 2 screener

Bit	31	30	29	28	27	26	25	24
	MAX_RATE_VAL[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
	MAX_RATE_VAL[7:0]							
Access	R/W	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
	INTERVAL_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
	INTERVAL_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 31:16 – MAX_RATE_VAL[15:0] Maximum Rate Value for the Interval Time
Defines the maximum number of bytes the type 2 screener is permitted to match in the programmed interval time.

Bits 15:0 – INTERVAL_TIME[15:0] Interval Time for Maximum Rate Checking
Defines the period for which the total number of bytes of received frames matched by the screener are accumulated and compared with the value configured in MAX_RATE_VAL. When the value exceeds MAX_RATE_VAL, then the current frame and frames subsequently matched will dropped until an interval time passes where MAX_RATE_VAL is not exceeded. When MAX_RATE_VAL=0, then no rate limiting will be performed. The interval time is specified in units of 64 receive clock periods.

62.8.313.GMAC Express MAC Screening Type 2 Rate Limit Register 1

Name: GMAC_EMAC_SCR2_RATE_LIMIT1
Offset: 0x1B44
Reset: 0x00000000
Property: Read/Write

This register manages the traffic policing function relevant to the 802.1Qci standard and associated to type 2 screener

Bit	31	30	29	28	27	26	25	24
	MAX_RATE_VAL[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
	MAX_RATE_VAL[7:0]							
Access	R/W	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
	INTERVAL_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
	INTERVAL_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 31:16 – MAX_RATE_VAL[15:0] Maximum Rate Value for the Interval Time
Defines the maximum number of bytes the type 2 screener is permitted to match in the programmed interval time.

Bits 15:0 – INTERVAL_TIME[15:0] Interval Time for Maximum Rate Checking
Defines the period for which the total number of bytes of received frames matched by the screener are accumulated and compared with the value configured in MAX_RATE_VAL. When the value exceeds MAX_RATE_VAL, then the current frame and frames subsequently matched will dropped until an interval time passes where MAX_RATE_VAL is not exceeded. When MAX_RATE_VAL=0, then no rate limiting will be performed. The interval time is specified in units of 64 receive clock periods.

62.8.314.GMAC Express MAC Screening Type 2 Rate Status Register

Name: GMAC_EMAC_SCR2_RATE_STATUS
Offset: 0x1B80
Reset: 0x00000000
Property: Read-only

This register manages the traffic policing function relevant to the 802.1Qci standard and associated to type 2 screener.

Bit	31	30	29	28	27	26	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
							EXCESS_RATE_R1	EXCESS_RATE_R0
Access							R	R
Reset							0	0

Bits 0, 1 – EXCESS_RATE_Rx Excessive Screener Rate Register

Value	Description
0	No excessive rate in screener since the last read of GMAC_EMAC_SCR2_RATE_STATUS.
1	A screener rate limiting mechanism has been triggered since the last read of GMAC_EMAC_SCR2_RATE_STATUS.

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, LON, , 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 provides an additional logic for CRC calculation and checking and an optimized interface to get an easy link with Two-Pin mode devices such as 24-bit ADC MCP3910.

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

- 16-data Transmit and Receive FIFOs
- Programmable Baud Rate Generator
- Baud Rate can be Independent of the Processor/Peripheral Clock
- 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
- LON Mode
  - Compliant with CEA-709 specification
  - Full-layer 2 implementation
  - Differential Manchester encoding/decoding (CDP)
  - Preamble generation including bit- and byte-sync fields
  - LON timings handling (beta1, beta2, IDT, etc.)
  - CRC generation and checking
  - Automated random number generation
  - Backlog calculation and update
  - Collision detection support
  - Supports both comm_type = 1 and comm_type = 2 modes
  - Clock drift tolerance up to 16%
  - Optimal for node-to-node communication (no embedded digital line filter)
- 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

- 16-data Transmit and Receive FIFOs
- Host or Client Serial Peripheral Bus Interface

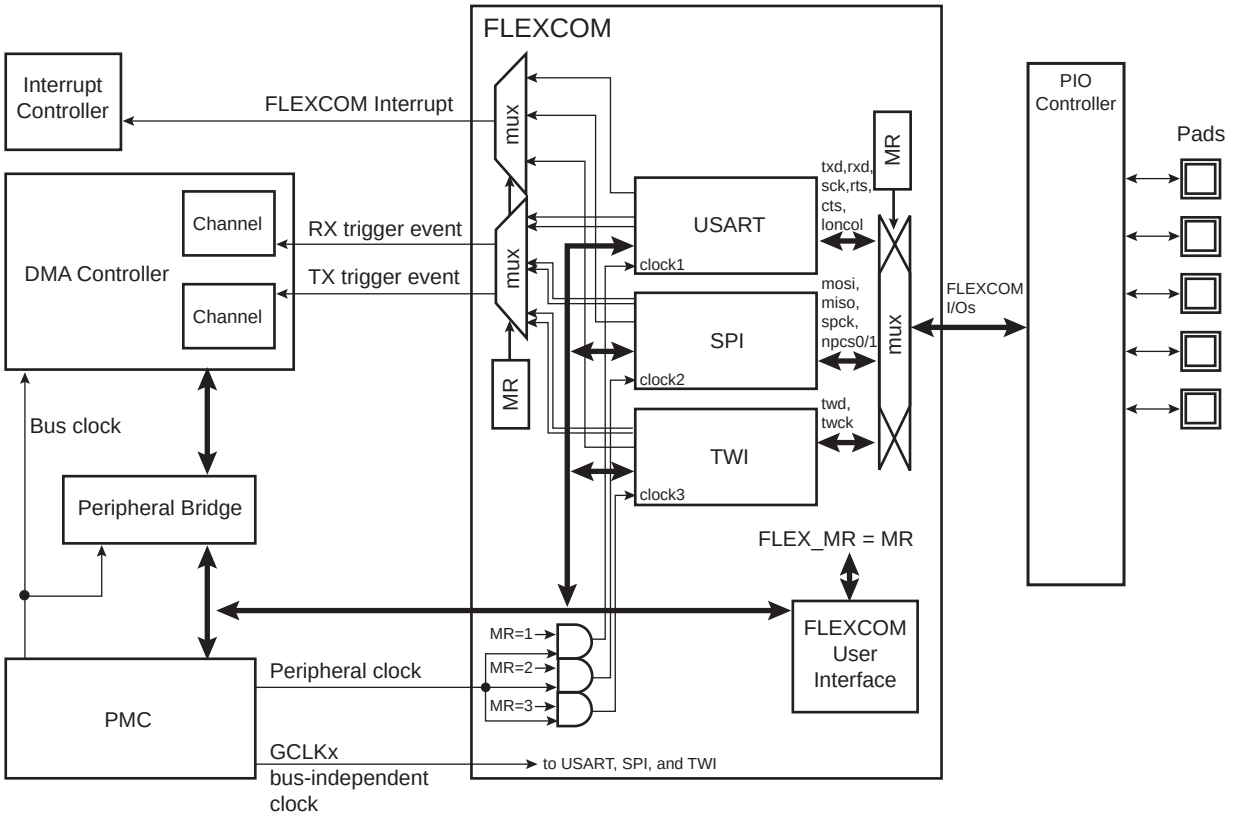
  - 8-bit to 16-bit programmable data length per chip select
  - Programmable phase and polarity per chip select
  - Programmable transfer delay between consecutive transfers and delay before SPI clock per chip select
  - Programmable delay between chip selects
- Selectable Mode Fault Detection
- Host 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
- CRC Generation and Checking
- Two-Pin Mode for Easy Link with MCP3910 ADC
- Functional Safety: Protection, Monitors and Reports
  - Register Write protection
  - Reports any write-protected access

63.2.3. TWI/SMBus Characteristics

- 16-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:**
  a. 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	Function	Signal Name by Mode					
		SPI	TWI	2-Wire UART	4-Wire UART	USART	ISO7816
FLEXCOM_IO0	Transmit Data	MOSI	TWD	TXD	TXD	TXD	TXD
FLEXCOM_IO1	Receive Data	MISO	TWCK	RXD	RXD	RXD	
FLEXCOM_IO2	Serial Clock	SPCK	–	–	–	SCK	SCK
FLEXCOM_IO3	Clear to send/Chip Select	NPCS0/NSS	–	–	CTS	–	–
FLEXCOM_IO4	Request to send/Chip Select	NPCS1	–	–	RTS	–	–
FLEXCOM_IO5	Chip Select	NPCS2	–	–	–	–	–
FLEXCOM_IO6	Chip Select	NPCS3	–	–	–	–	–
FLEXCOM_IO7	LON Collision	–	–	–	–	LONCOL	–

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.

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 (for example, 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 division of the peripheral clock, the divider being product dependent, but generally set to 8
- a fully programmable generic clock (GCLK) provided by PMC and independent of processor/peripheral clock
- the external clock, available on the SCK pin

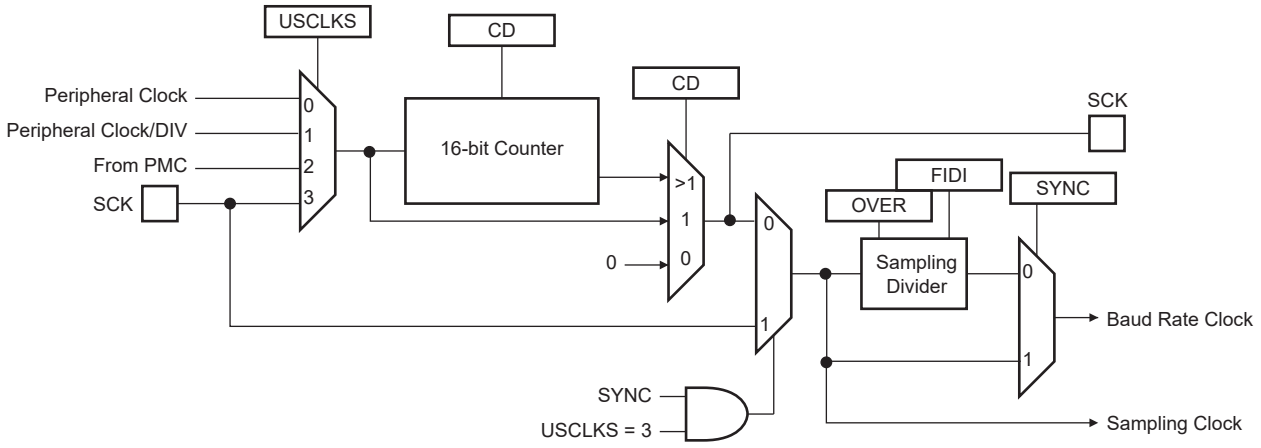
The baud rate generator is based upon a 16-bit divider, which is programmed with the CD field of the Baud Rate Generator register (FLEX_US_BRGR). If a zero is written to CD, the baud rate generator does not generate any clock. If a one is written to CD, the divider is bypassed and becomes inactive.

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 follow constraints depending on the operating mode (see Baud Rate in Synchronous Mode).

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%
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%

Table 63.3. Baud Rate Example (OVER = 0) (continued)

Source Clock (MHz)	Expected Baud Rate (bit/s)	Calculation Result	CD	Actual Baud Rate (bit/s)	Error
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%.

$$\text{Error} = 1 - \left(\frac{\text{Expected Baud Rate}}{\text{Actual Baud Rate}}\right)$$

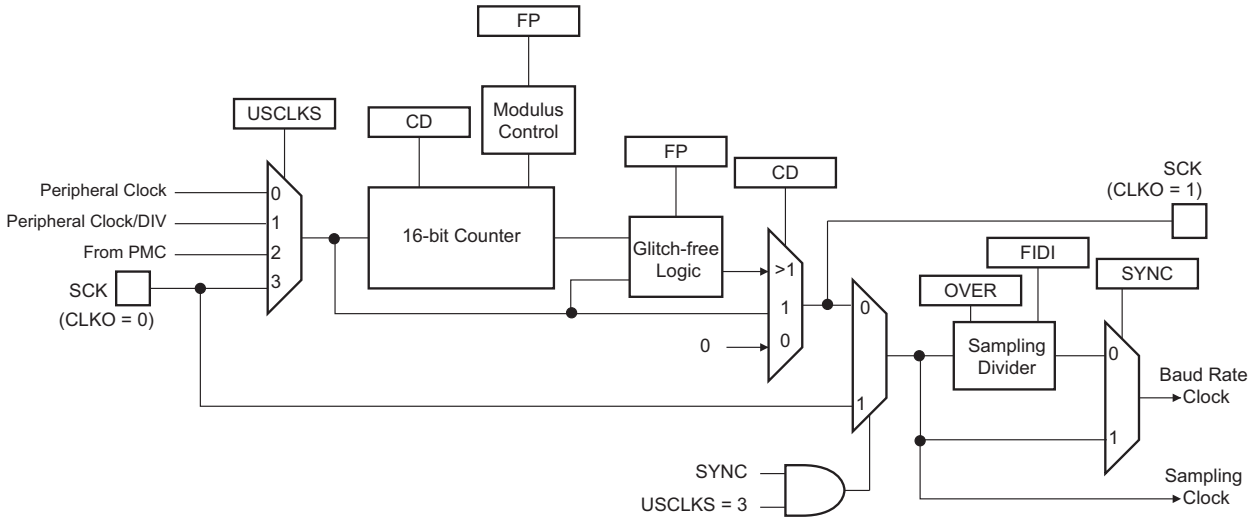
63.7.1.2. Fractional Baud Rate in Asynchronous Mode

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. When operating in asynchronous modes and the SCK pin is selected for baud rate generation, the external clock frequency must be at least three times lower than the system clock.

In Synchronous mode, SCK must be lower than $f_{\text{peripheral clock}}/6$.

Note: In Transmit only Synchronous mode (FLEX_US_CR.TXEN=1, FLEX_US_CR.RXDIS=1, FLEX_US_MR.SYNC=1, USCLKS=3), the transmit path (TXD) can be operated at $f_{\text{peripheral clock}}/3$.

When either the external clock SCK or the internal clock divided (peripheral clock/DIV or 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

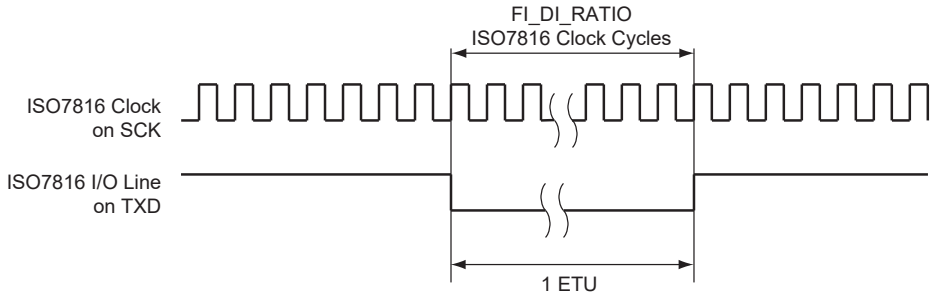
If the USART is configured in ISO7816 mode, the clock selected by the USCLKS field in FLEX_US_MR is first divided by the value programmed in field CD field in FLEX_US_BRGR. The resulting clock can be provided to the SCK pin to feed the smart card clock inputs. This means that FLEX_US_MR.CLKO can be set.

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 ISO7816 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.

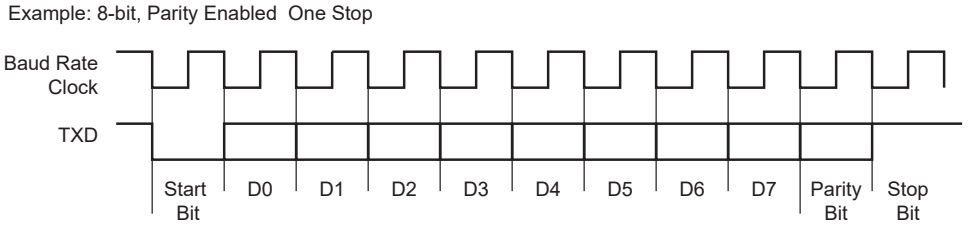
63.7.3. Synchronous and Asynchronous Modes

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.

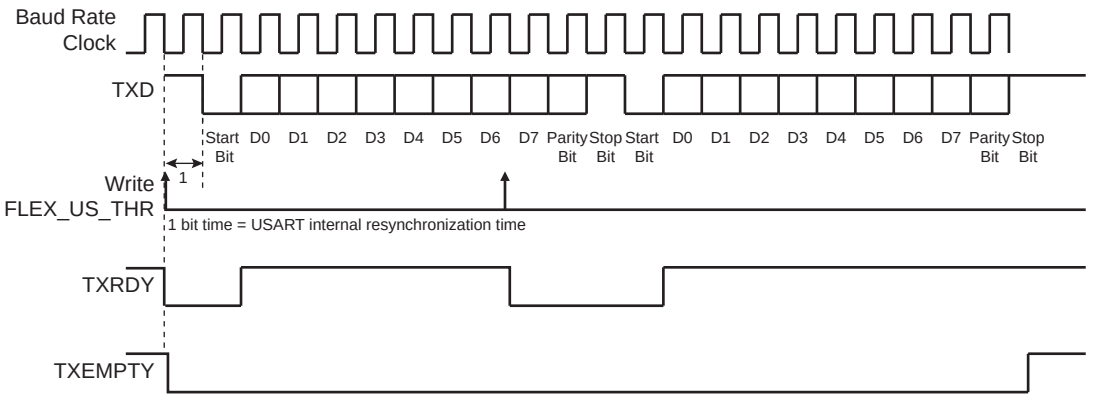
Figure 63.5. Character Transmit



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

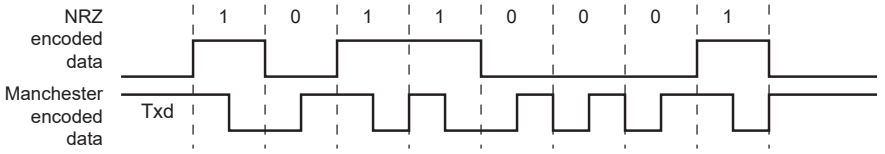


63.7.3.2. Manchester Encoder

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

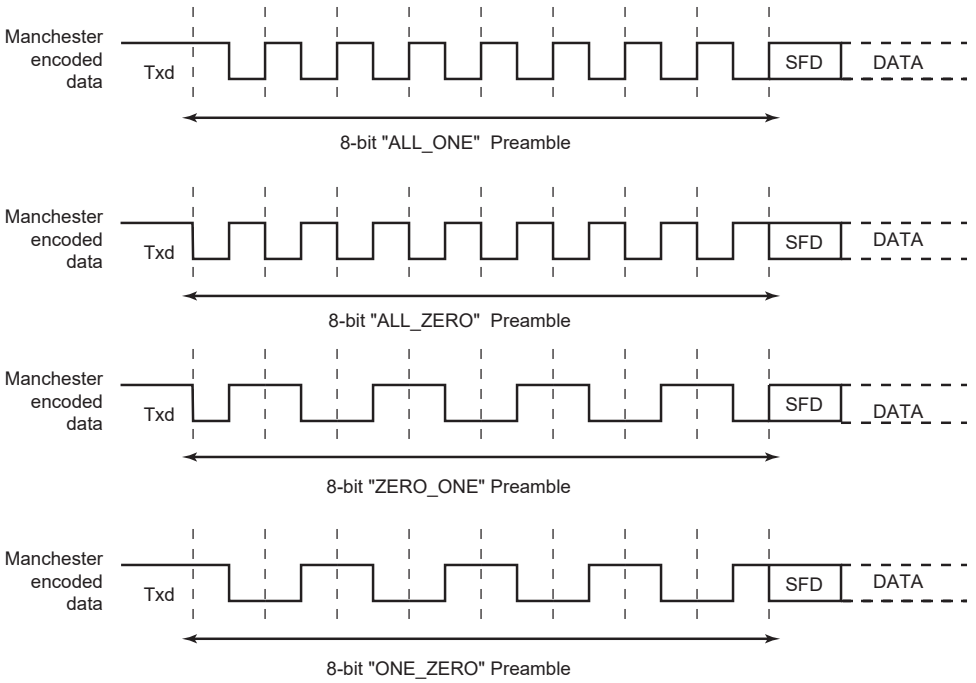
encoded sequence is: the byte 0xB1 or 10110001 encodes to 10 01 10 10 01 01 01 10, assuming the default polarity of the encoder. The following figure illustrates this coding scheme.

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.

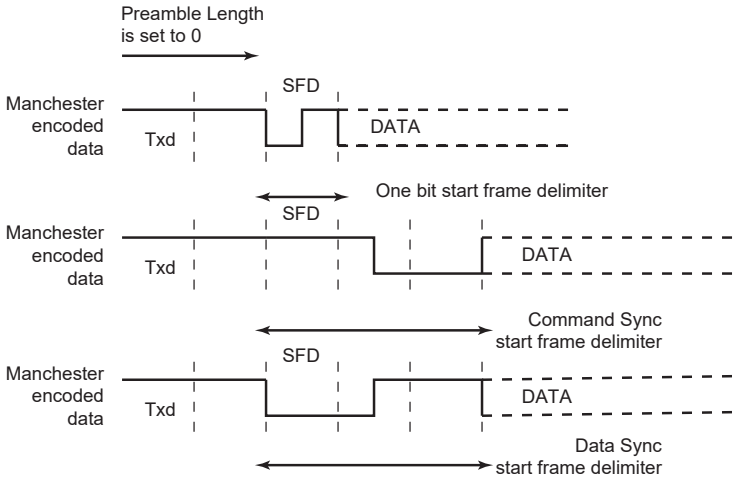
Figure 63.8. Preamble Patterns, Default Polarity Assumed



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.

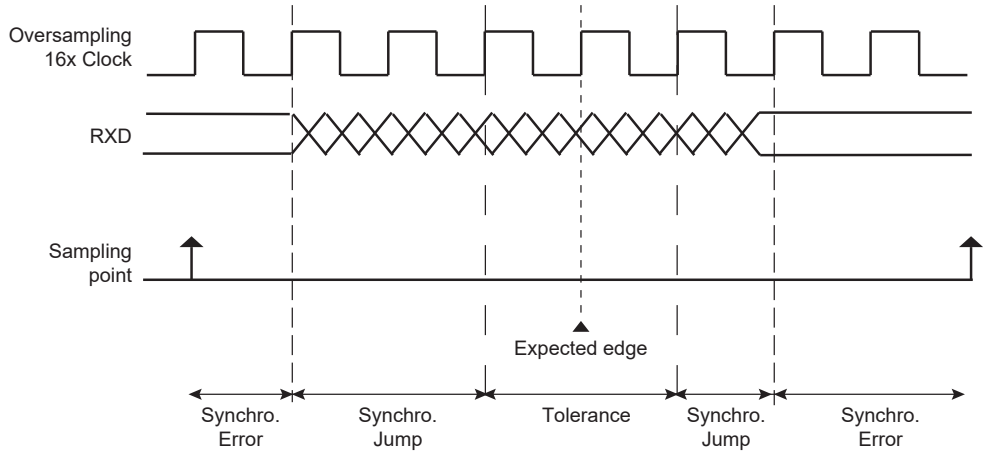
Figure 63.9. Start Frame Delimiter



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

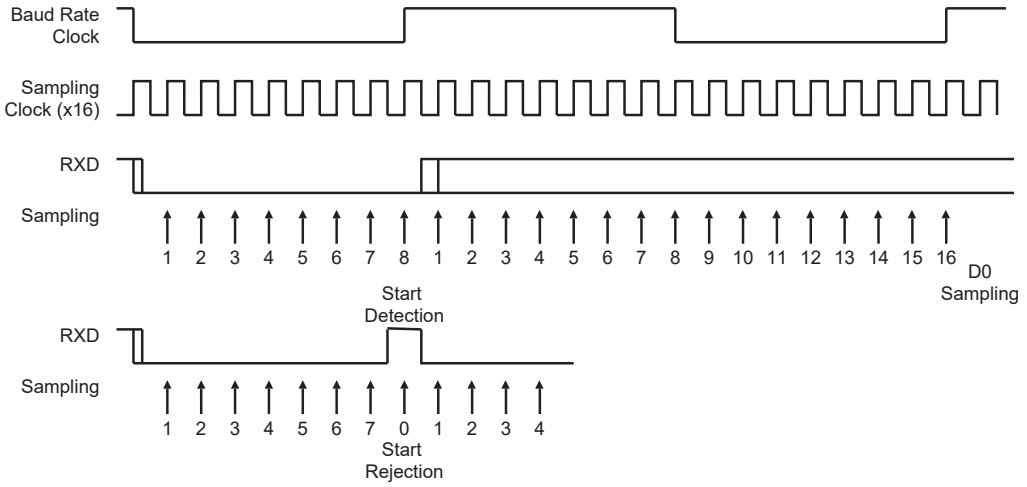
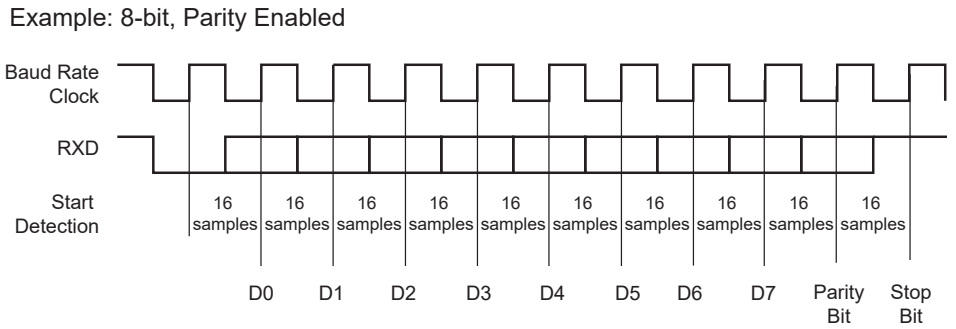


Figure 63.12. Asynchronous Character Reception



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.

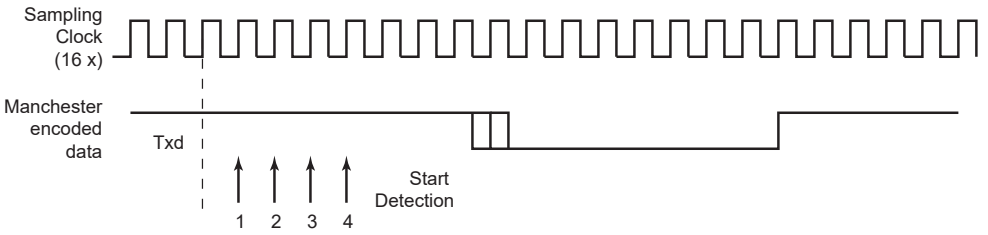
An optional preamble sequence can be defined. Its length is user-defined and totally independent of the transmitter side. Use the FLEX_US_MAN.RX_PL field to configure the length of the preamble

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

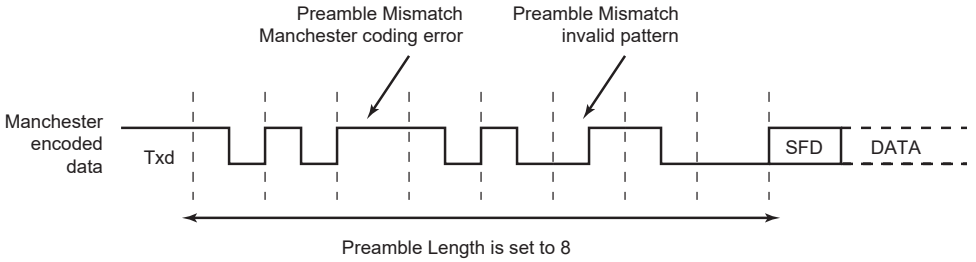
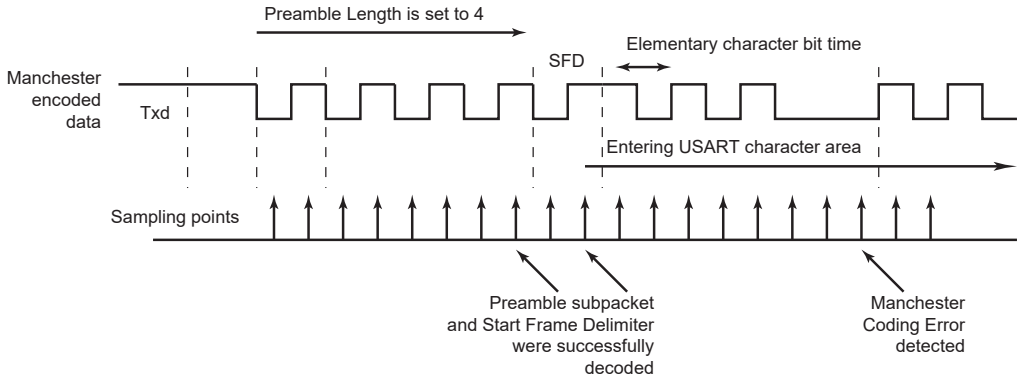


Figure 63.15. Manchester Error Flag



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.

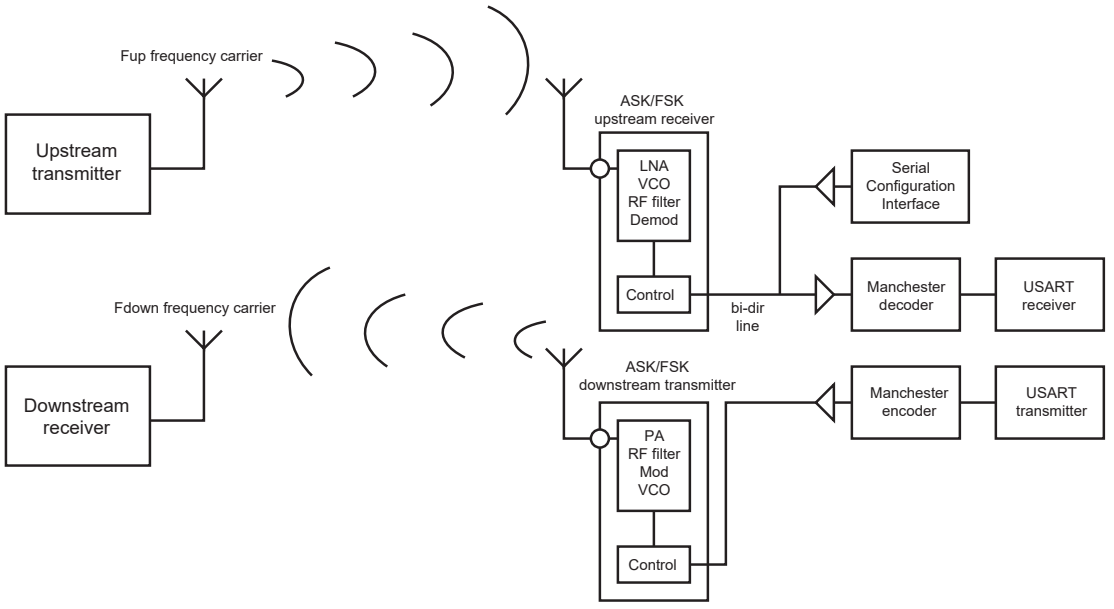
As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

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

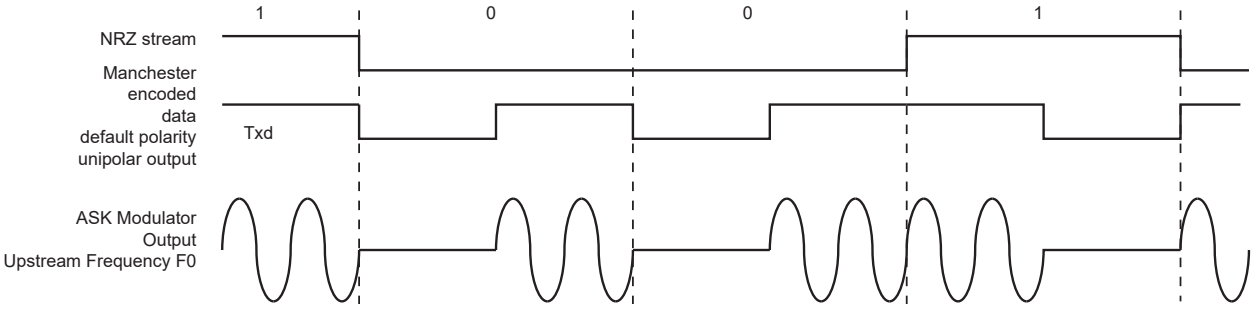
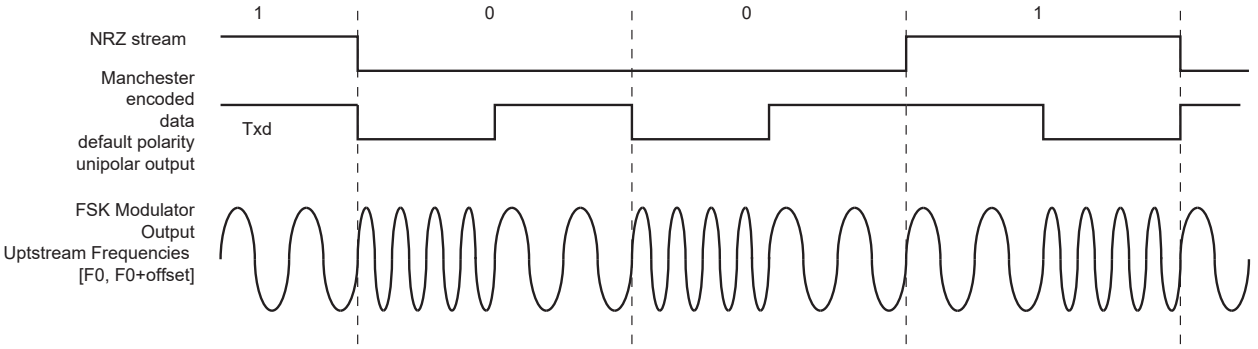


Figure 63.18. FSK Modulator Output



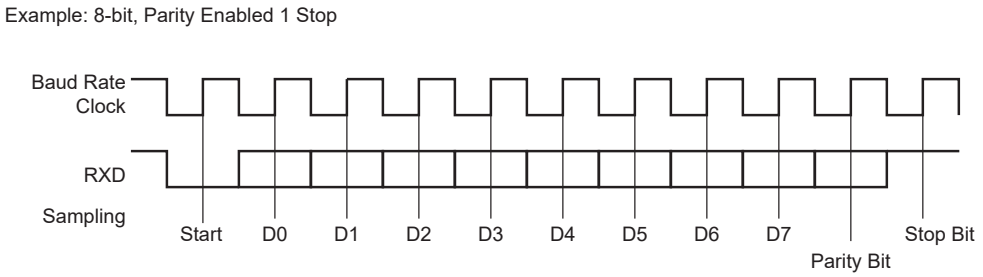
63.7.3.6.Synchronous Receiver

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.

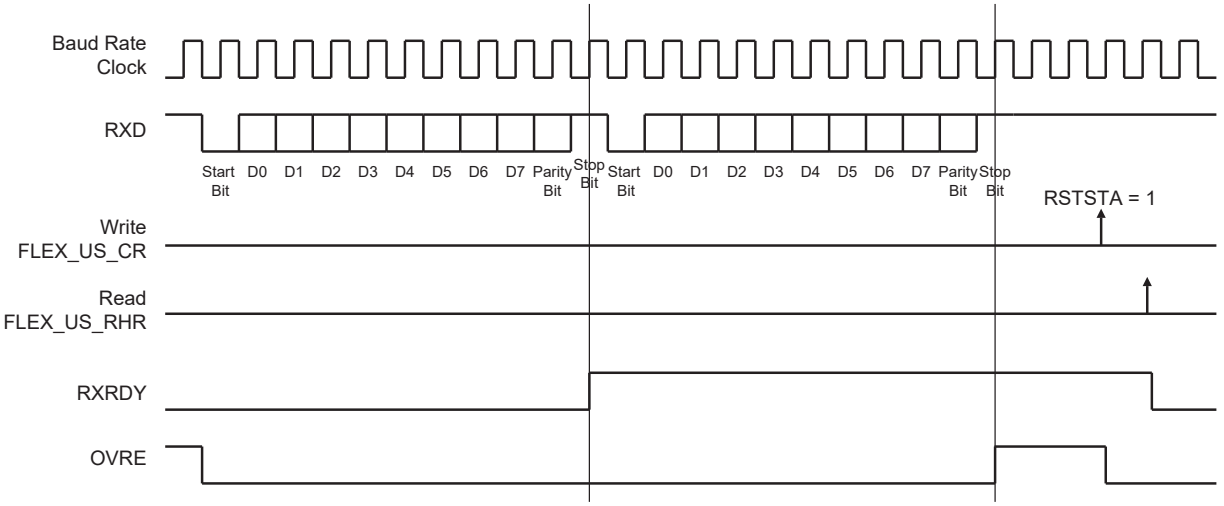
Figure 63.19. Synchronous Mode Character Reception



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.

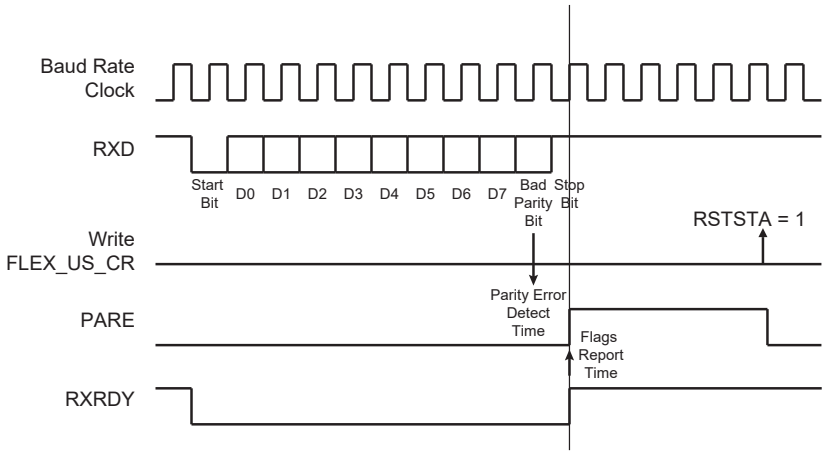
The following table shows an example of the parity bit for the character 0x41 (character ASCII “A”) depending on the configuration of the USART. Because there are two bits set to 1 in the character value, the parity bit is set to 1 when the parity is odd, or configured to 0 when the parity is even.

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.

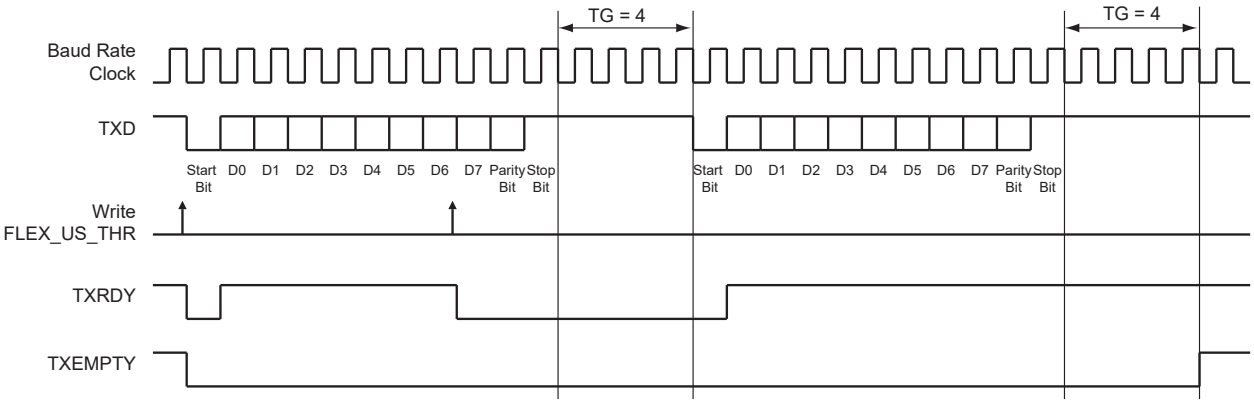
The timeguard function enables the transmitter to insert an idle state on the TXD line between two characters. This idle state actually acts as a long stop bit.

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.

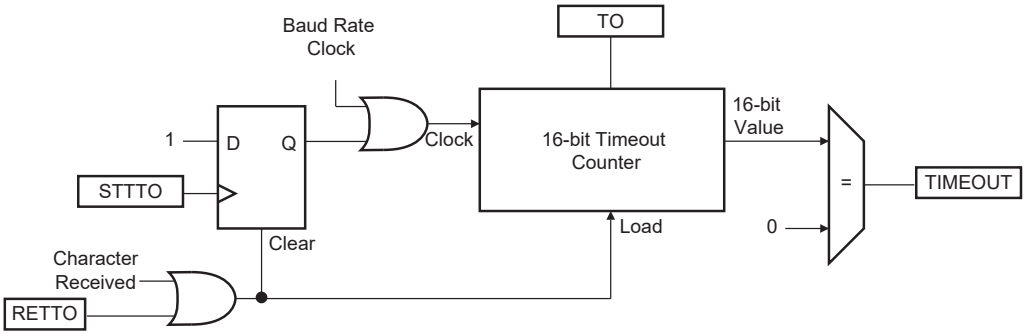
The timeout delay period (during which the receiver waits for a new character) is programmed in the TO field of the Receiver Timeout register (FLEX_US_RTOR). If the TO field is written to 0, the Receiver Timeout is disabled and no timeout is detected. The FLEX_US_CSR.TIMEOUT bit remains at 0. Otherwise, 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

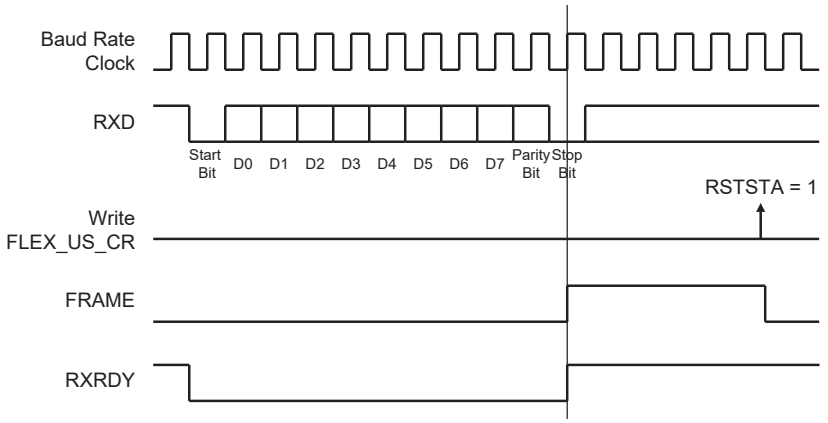
Baud Rate (bit/s)	Bit Time (µs)	Timeout (ms)
600	1,667	109,225
1,200	833	54,613
2,400	417	27,306
4,800	208	13,653
9,600	104	6,827
14,400	69	4,551
19,200	52	3,413
28,800	35	2,276
38,400	26	1,704
56,000	18	1,170
57,600	17	1,138
200,000	5	328

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.

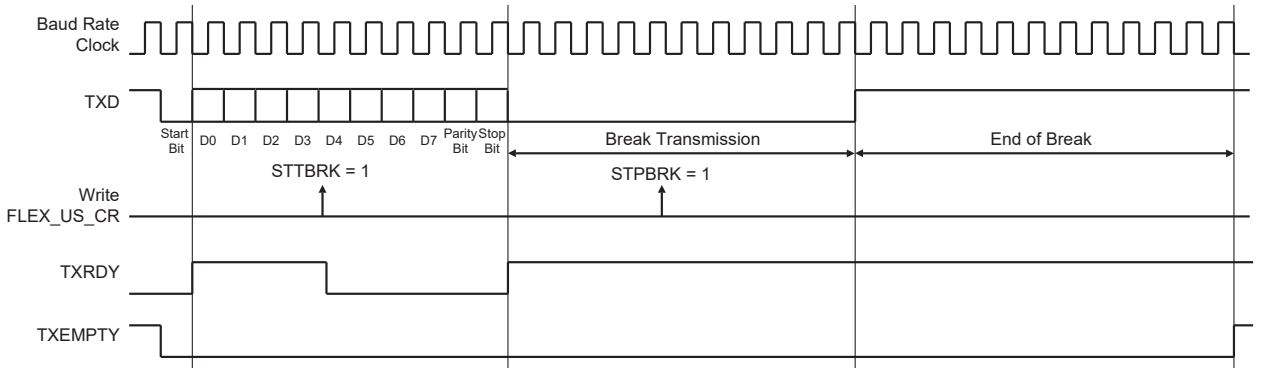
Setting both the FLEX_US_CR.STTBRK and FLEX_US_CR.STPBRK bits can lead to an unpredictable result. All Stop Break commands requested without a previous Start Break command are ignored. A byte written into the Transmit Holding register while a break is pending, but not started, is ignored.

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.

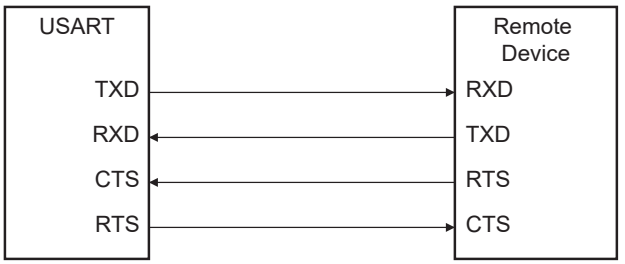
When the low stop bit is detected, the receiver asserts the FLEX_US_CSR.RXBRK bit. FLEX_US_CSR.RXBRK may be cleared by setting the FLEX_US_CR.RSTSTA bit.

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.

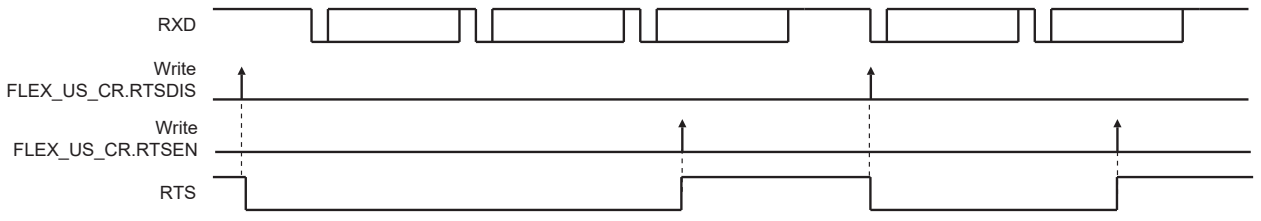
Figure 63.26. Connection with a Remote Device for Hardware Handshaking



Setting the USART to operate with hardware handshaking is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x2.

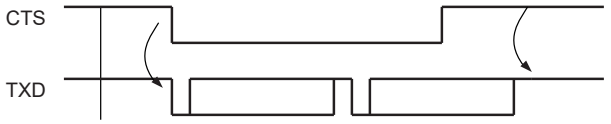
The USART behavior when hardware handshaking is enabled is the same as the behavior in standard Synchronous or Asynchronous mode, except that the receiver drives the RTS pin as described below and the level on the CTS pin modifies the behavior of the transmitter as described below. Using this mode requires 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



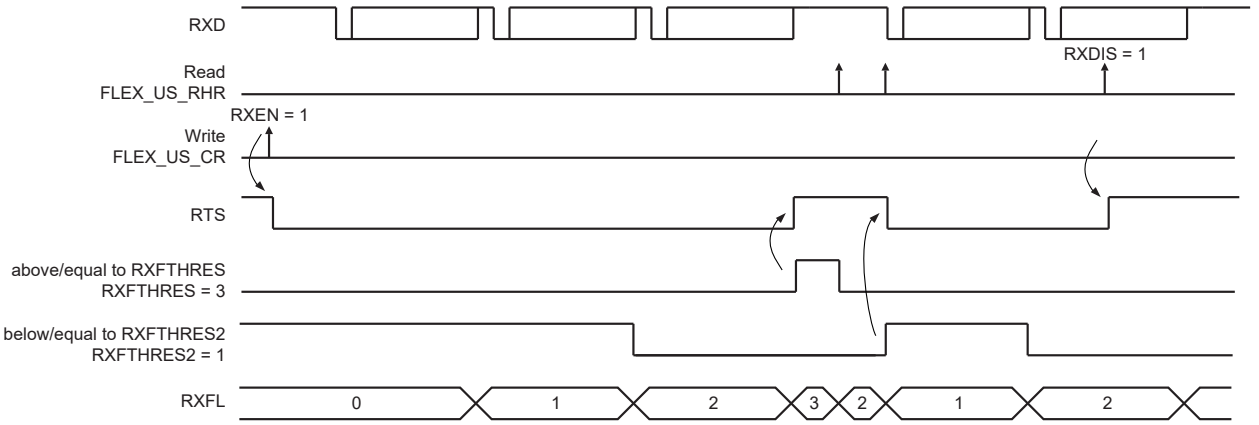
The following figure shows how the transmitter operates if hardware handshaking is enabled. The CTS pin disables the transmitter. If a character is being processed, the transmitter is disabled only after the completion of the current character and transmission of the next character happens as soon as 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.

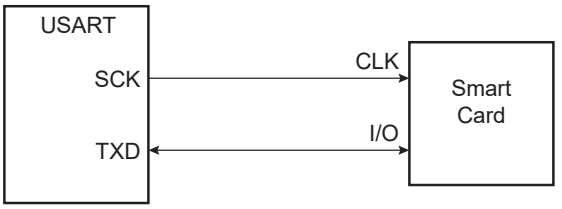
Setting the USART in ISO7816 mode is performed by writing the FLEX_US_MR.USART_MODE field to the value 0x4 for protocol T = 0 and to the value 0x6 for protocol T = 1.

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

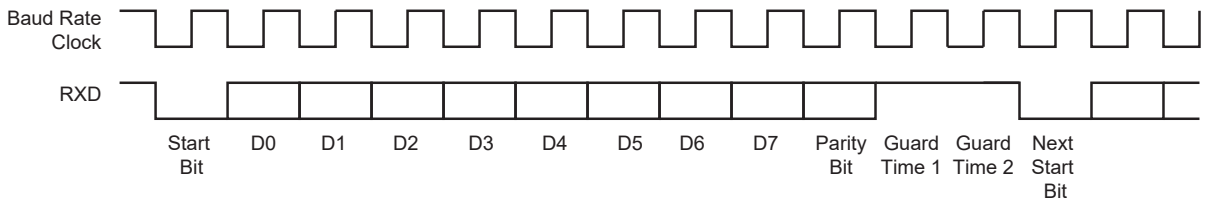
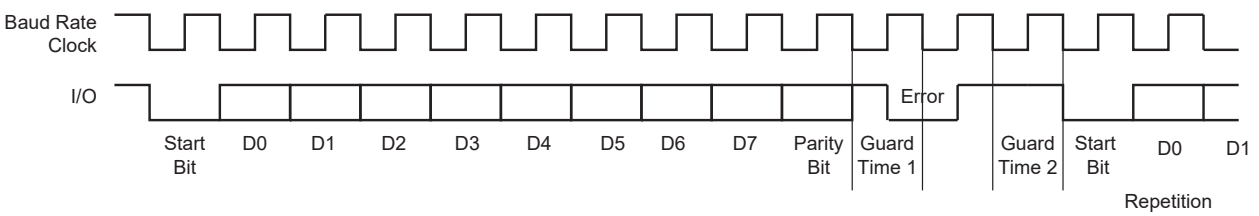


Figure 63.32. T = 0 Protocol with Parity Error



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

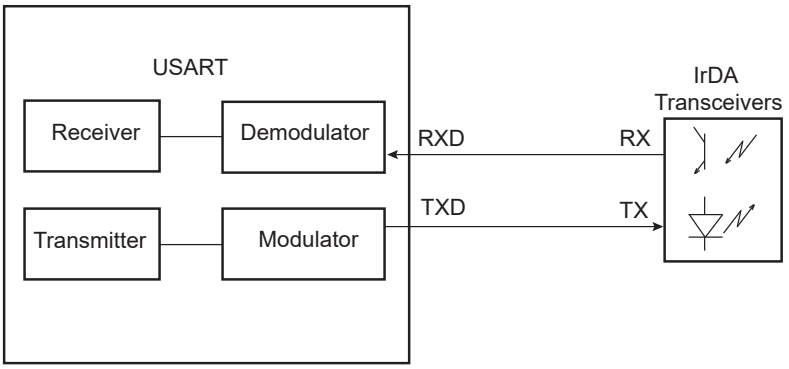
When operating in ISO7816 protocol T = 1, the transmission is similar to an asynchronous format with only one stop bit. The parity is generated when transmitting and checked when receiving. Parity error detection sets the FLEX_US_CSR.PARE bit.

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

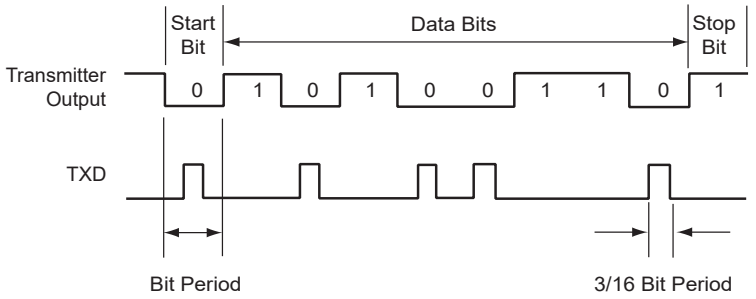
For baud rates up to and including 115.2 kbit/s, the RZI modulation scheme is used. “0” is represented by a light pulse of 3/16th of a bit time. Some examples of signal pulse duration are shown in the following table.

Table 63.10. IrDA Pulse Duration

Baud Rate	Pulse Duration (3/16)
2.4 kbit/s	78.13 µs
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

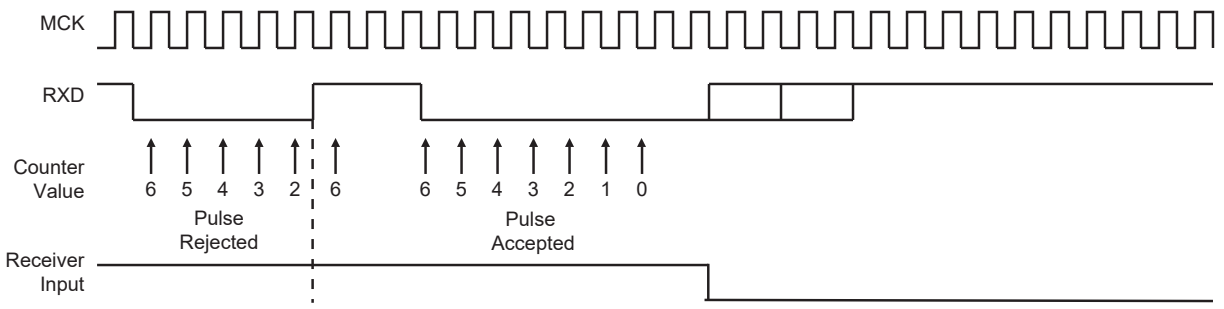
Peripheral Clock	Baud Rate (bit/s)	CD	Baud Rate Error	Pulse Time (µs)
3,686,400	115,200	2	0.00%	1.63
20,000,000	115,200	11	1.38%	1.63
32,768,000	115,200	18	1.25%	1.63
40,000,000	115,200	22	1.38%	1.63
3,686,400	57,600	4	0.00%	3.26
20,000,000	57,600	22	1.38%	3.26
32,768,000	57,600	36	1.25%	3.26
40,000,000	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
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:

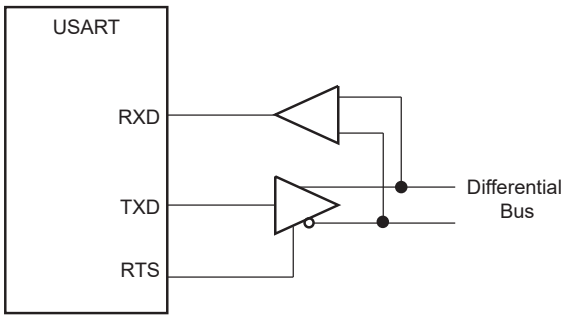
$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$

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.

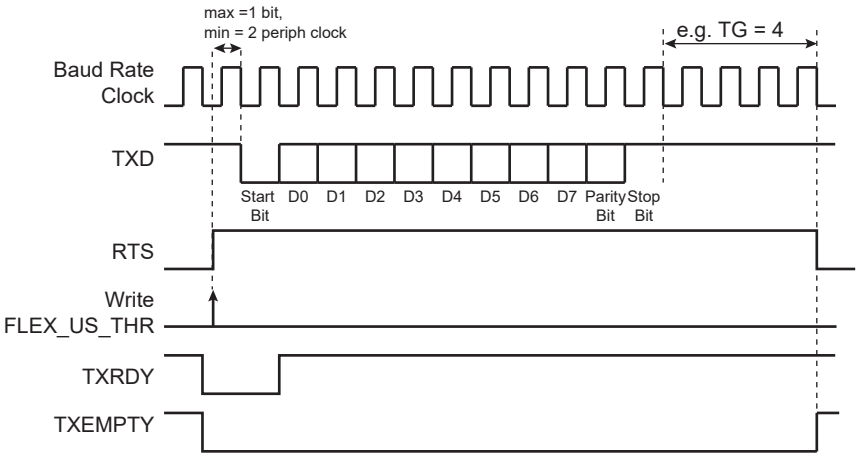
Figure 63.36. Typical Connection to an RS485 Bus



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 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.

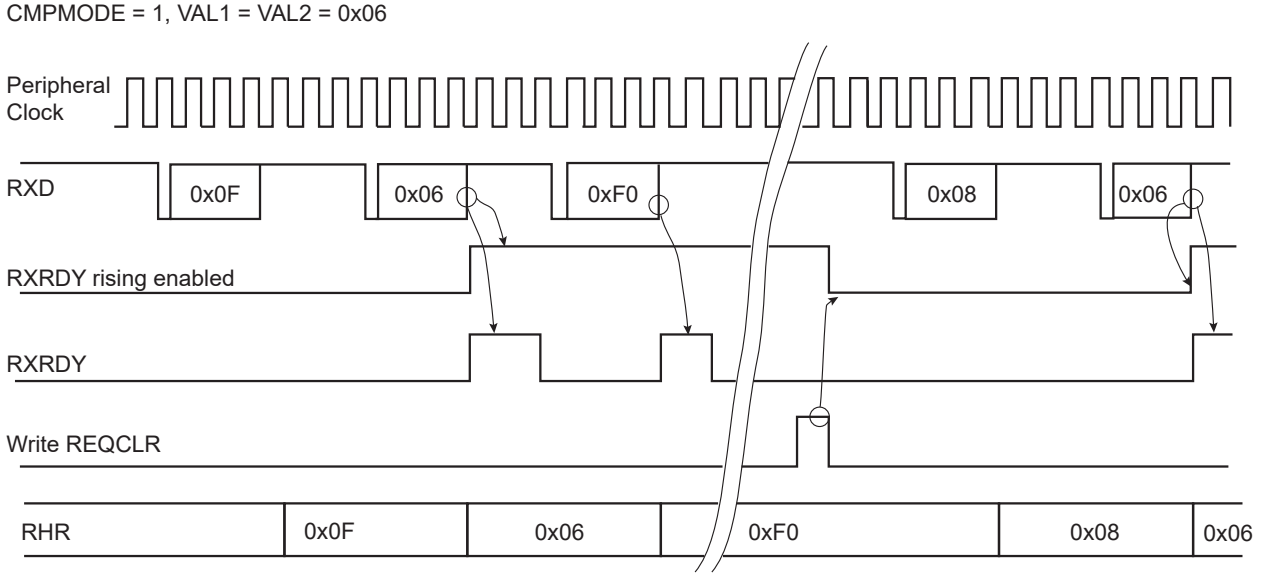
When the FLEX_US_CMPR.CMPMODE bit is set to FLAG_ONLY (value 0), all received data are loaded in FLEX_US_RHR and the CMP flag provides the status of the comparison result.

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



63.7.8. 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.

63.7.8.1. 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.

63.7.8.2. 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.

63.7.8.3. Receiver and Transmitter Control

See Receiver and Transmitter Control.

63.7.8.4. Character Transmission

See Transmitter Operations.

63.7.8.5. Character Reception

See Receiver Operations.

63.7.8.6. 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.

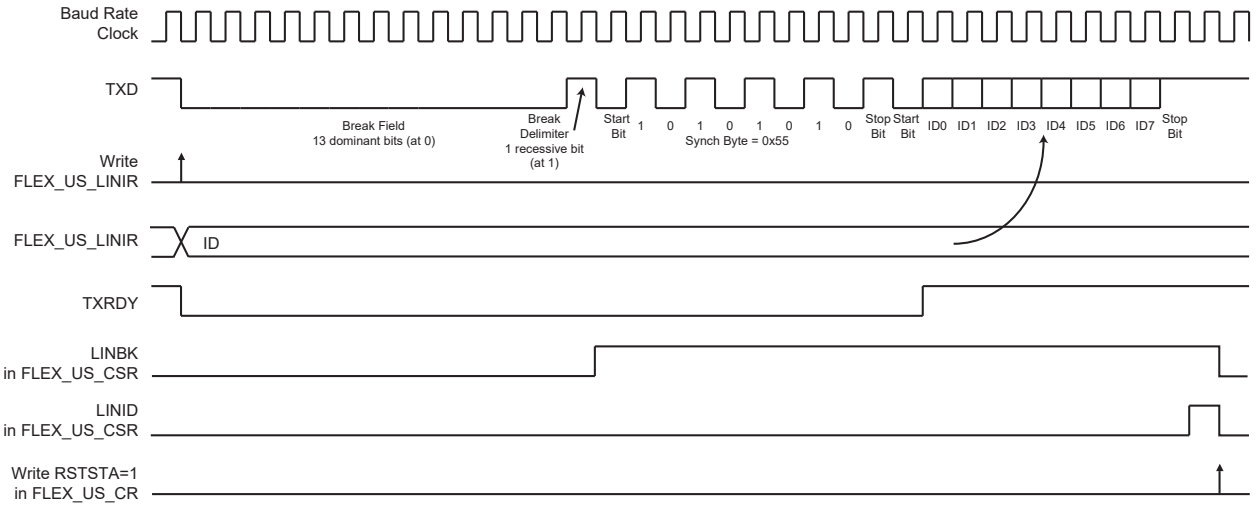
The Break Field, the Synch Field and the Identifier Field are sent automatically one after the other.

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 63.39. Header Transmission



63.7.8.7. 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).

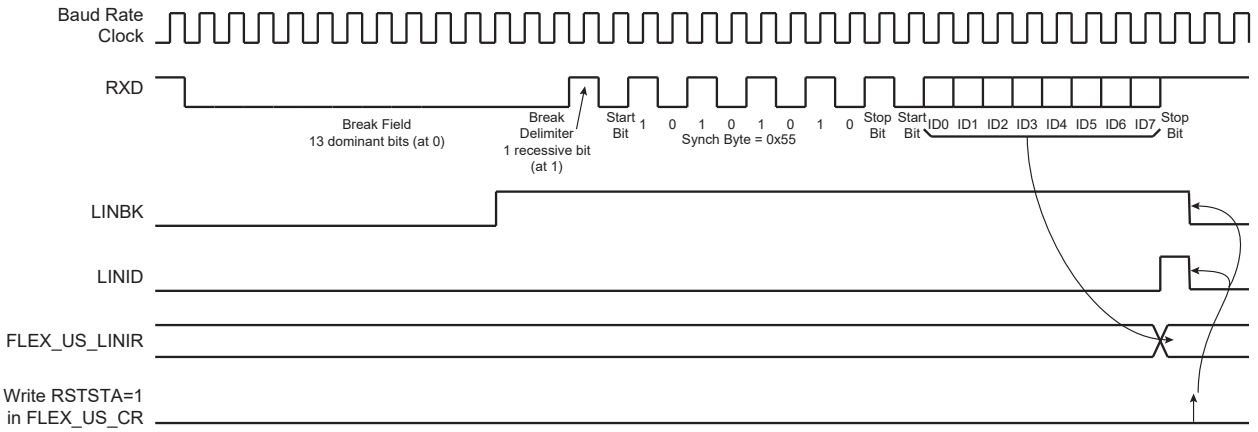
After receiving the Synch Field, the USART expects to receive the Identifier Field.

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.

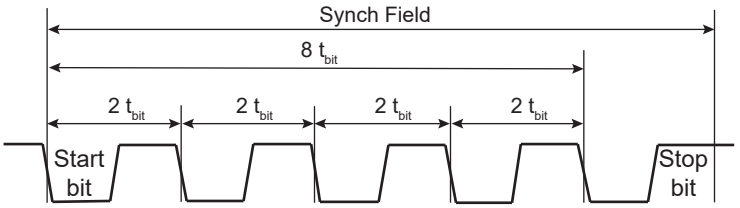
Figure 63.40. Header Reception



63.7.8.8. 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 63.41. 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).

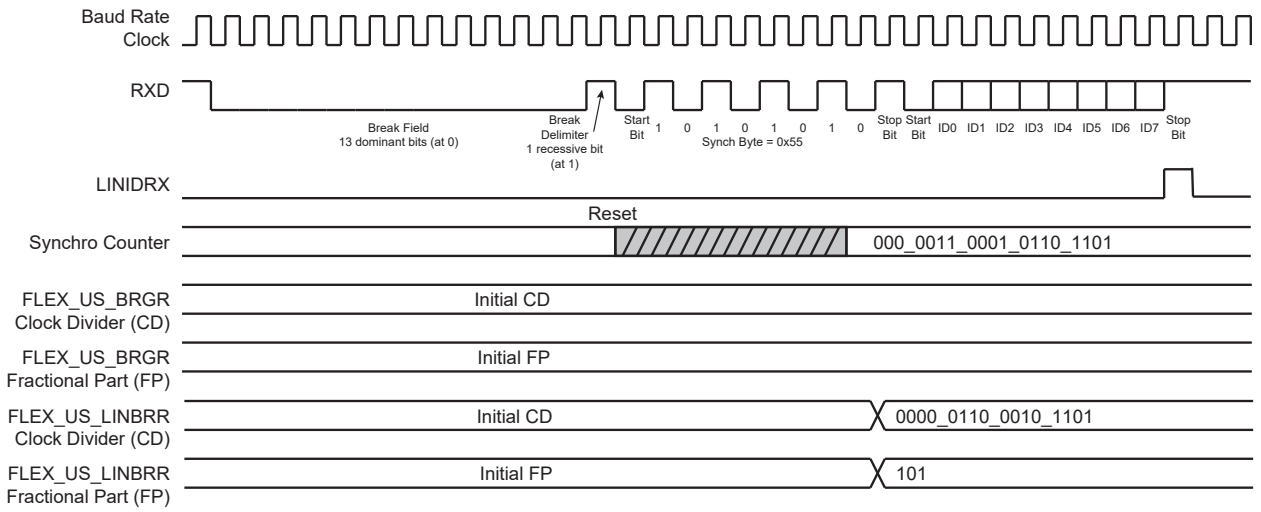
Once the Synch Field has been entirely received, the clock divider (LINCD) and the fractional part (LINFP) are updated in the LIN Baud Rate register (FLEX_US_LINBRR) with the computed values, if the Synchronization is not disabled by the SYNCDIS bit in the LIN Mode register (FLEX_US_LINMR).

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 63.42. 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).

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times f_{CLIENT}}\right)\%$$

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{Over}) + \beta] \times \text{Baud rate}}{8 \times \left(\frac{f_{TOL\_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

$-0.5 \le \alpha \le +0.5 \quad -1 < \beta < +1$

$f_{TOL\_UNSYNCH}$ 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:

$$f_{Nom}(\text{min}) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{Over}) + 1] \times \text{Baud rate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

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

63.7.8.9.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.

63.7.8.10.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

63.7.8.11.Response Data Length

The LIN response data length is the number of data fields (bytes) of the response excluding the checksum.

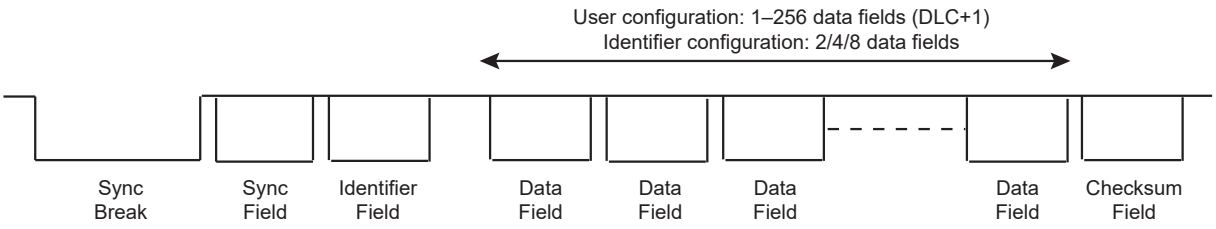
The response data length can either be configured by the user or be defined automatically by bits 4 and 5 of the Identifier (compatibility to LIN Specification 1.1). The user can choose between these two modes by the FLEX_US_LINMR.DLM bit:

- 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 63.43. Response Data Length



63.7.8.12.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).

63.7.8.13.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)
- $t_{Frame\_Maximum} = 1.4 \times (34 + 10 \times (DLC + 1 + 1) + 1) \times t_{bit}$
- $t_{Frame\_Maximum} = (77 + 14 \times DLC) \times t_{bit}$

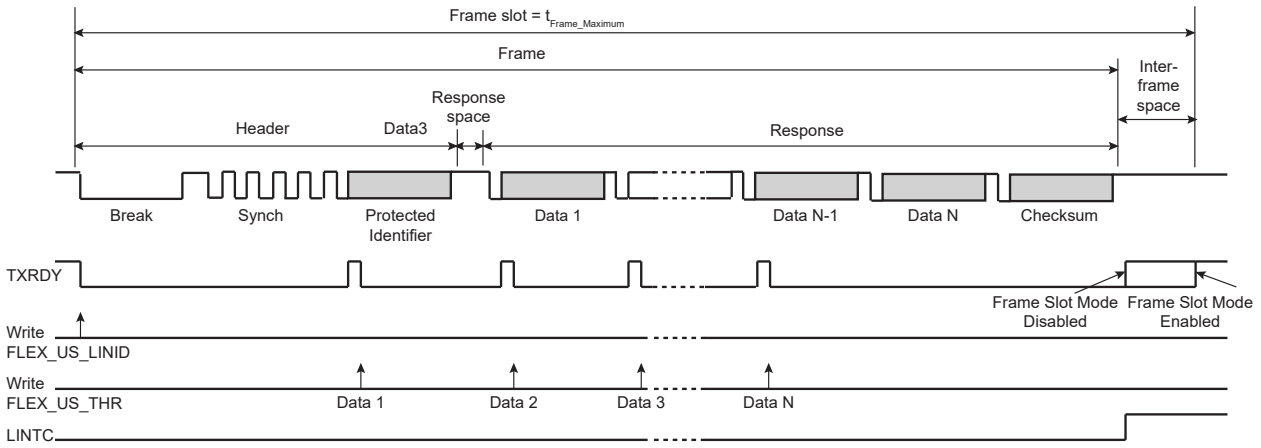
If the Checksum is not sent (CHKDIS = 1):

- $t_{Header\_Nominal} = 34 \times t_{bit}$
- $t_{Response\_Nominal} = 10 \times NData \times t_{bit}$
- $t_{Frame\_Maximum} = 1.4 \times (t_{Header\_Nominal} + t_{Response\_Nominal} + 1)$(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 63.44. Frame Slot Mode



63.7.8.14.LIN Errors

63.7.8.14.1.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.

63.7.8.14.2.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.

63.7.8.14.3.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.

63.7.8.14.4.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.

63.7.8.14.5.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.

63.7.8.14.6.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.

63.7.8.14.7.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.

63.7.8.15.LIN Frame Handling

63.7.8.15.1.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 63.45. Host Node Configuration, NACT = PUBLISH

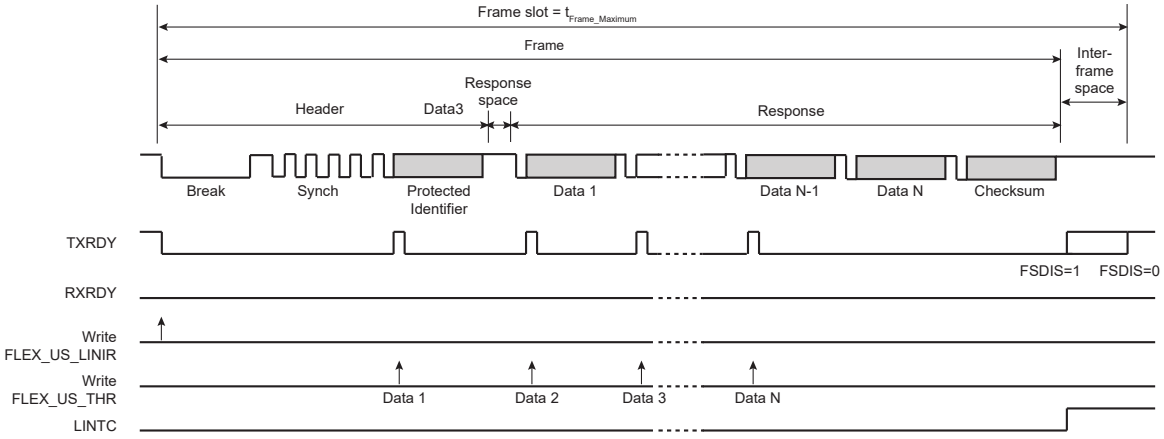


Figure 63.46. Host Node Configuration, NACT = SUBSCRIBE

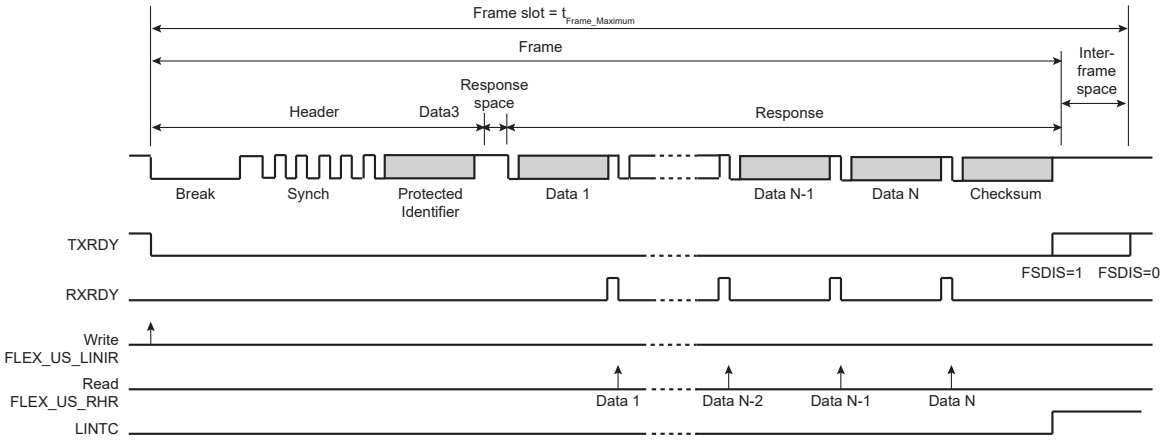
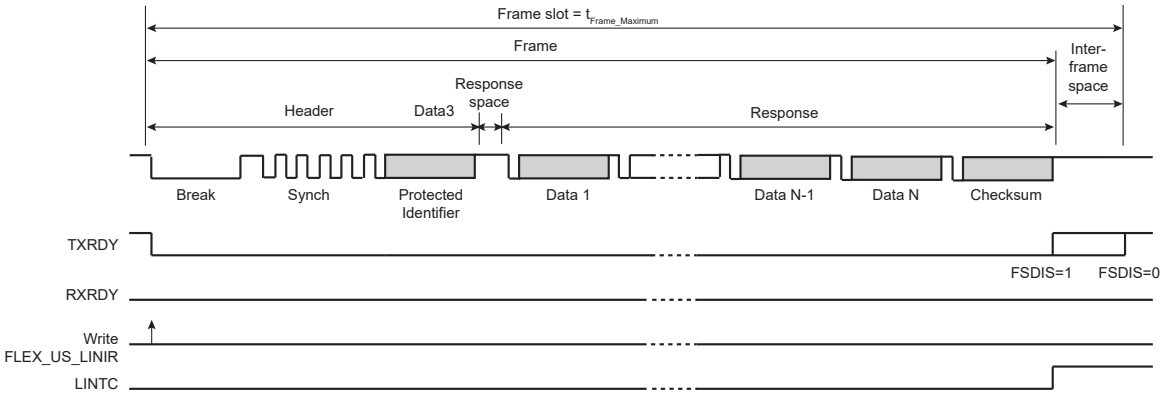


Figure 63.47. Host Node Configuration, NACT = IGNORE



63.7.8.15.2.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 63.48. Client Node Configuration, NACT = PUBLISH

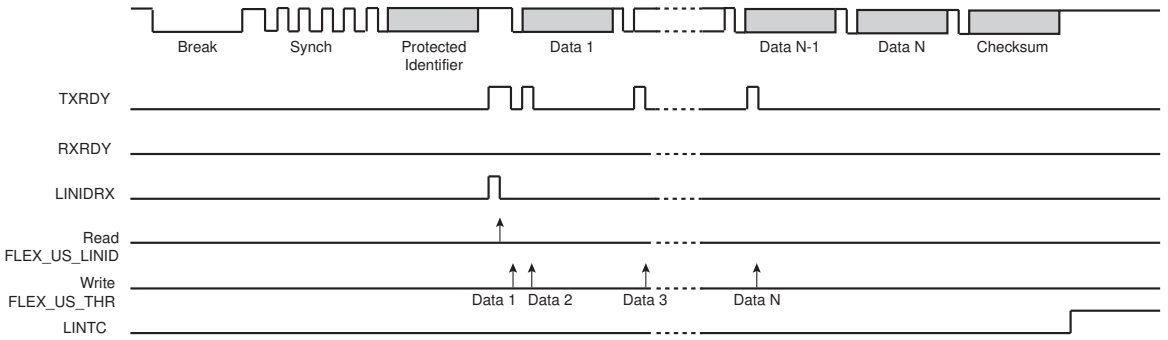


Figure 63.49. Client Node Configuration, NACT = SUBSCRIBE

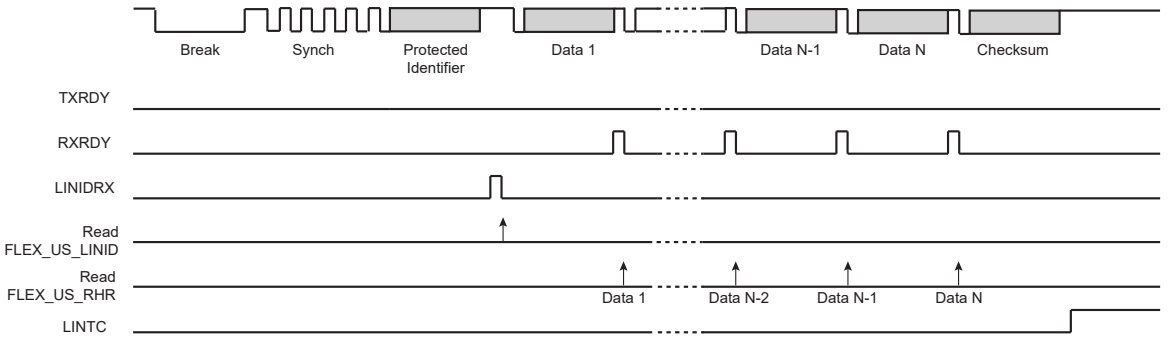
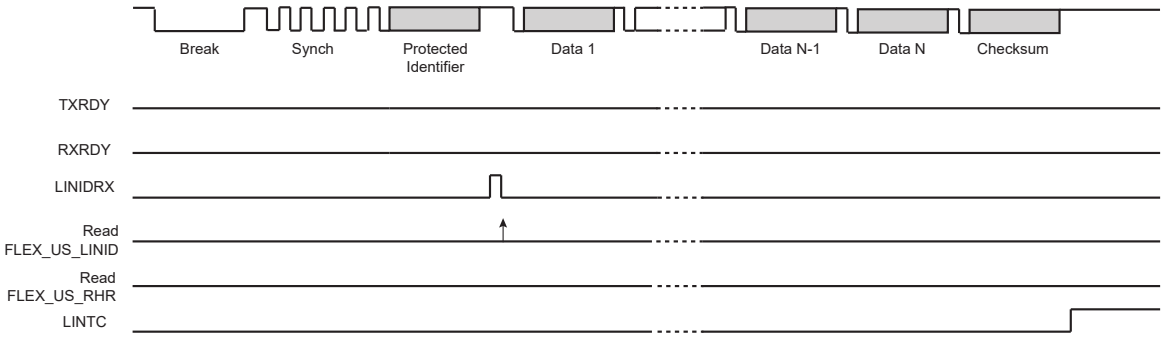


Figure 63.50. Client Node Configuration, NACT = IGNORE



63.7.8.16.LIN Frame Handling with the DMA

The USART can be used in association with the DMA in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The DMA uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The DMA 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 DMA in the USART is always a byte.

63.7.8.16.1.Host Node Configuration

The user can choose between two DMA 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 DMA in the Transmit Holding register FLEX_US_THR (instead of the LIN Mode register FLEX_US_LINMR). Because the DMA 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 63.51. Host Node with DMA (PDCM = 1)

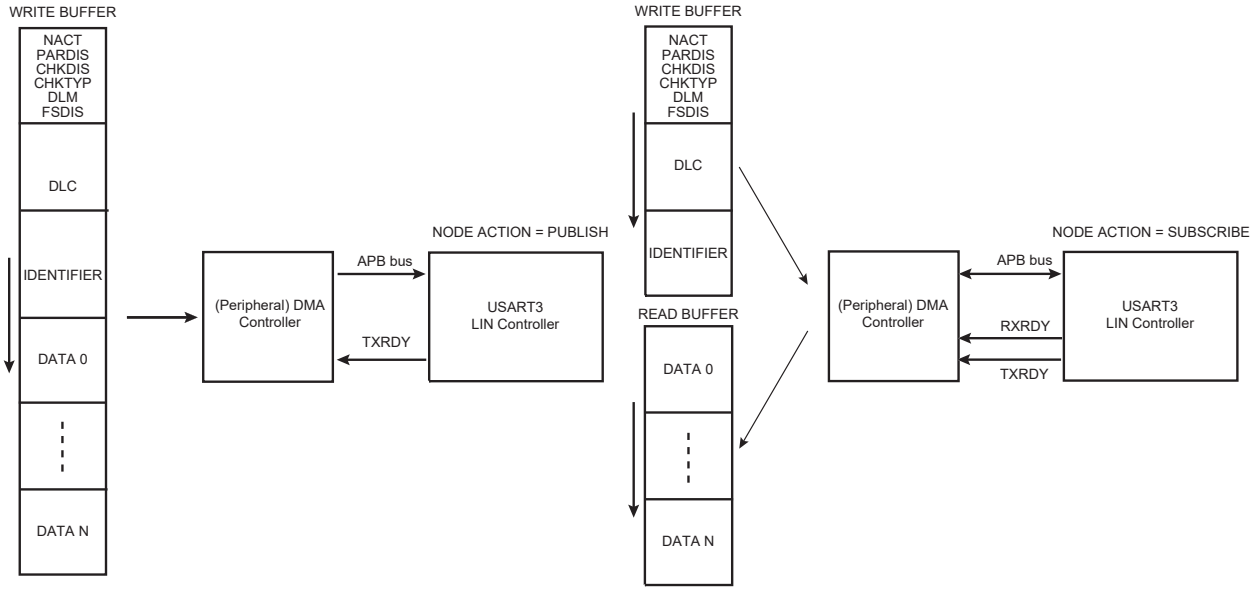
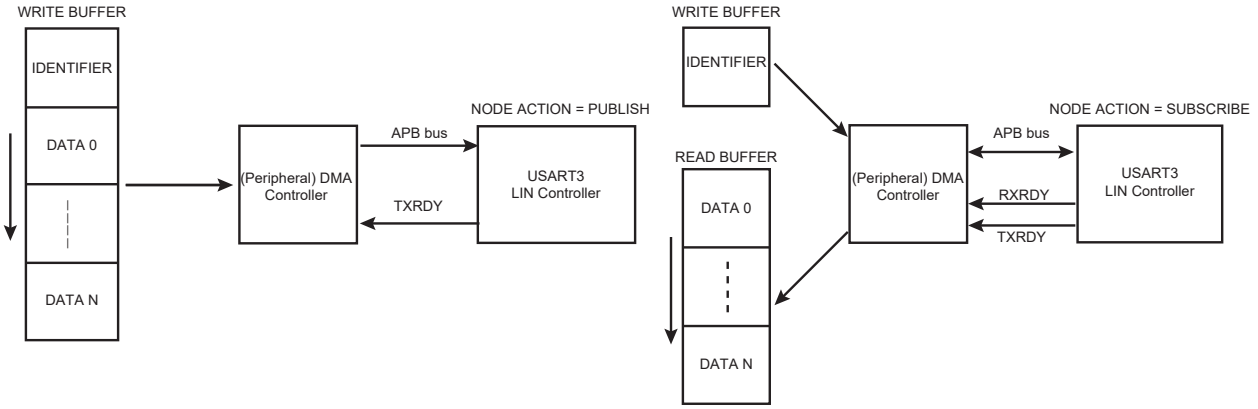


Figure 63.52. Host Node with DMA (PDCM = 0)



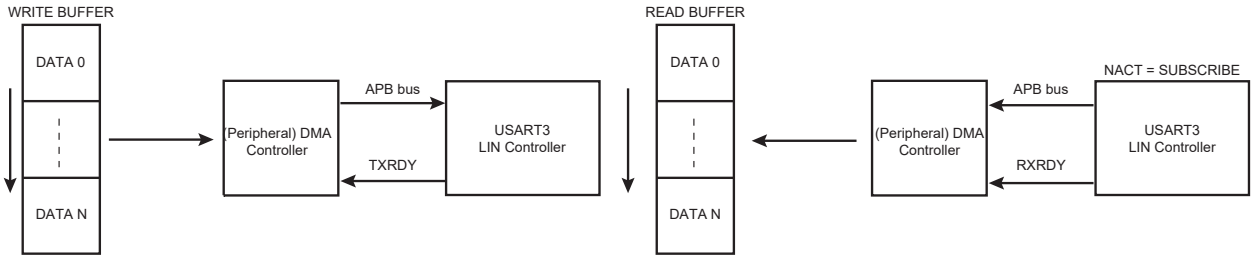
63.7.8.16.2.Client Node Configuration

In this configuration, the DMA 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 63.53. Client Node with DMA



63.7.8.17.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.

63.7.8.18.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

Table 63.13. Receiver Timeout Programming (continued)

LIN Specification	Baud Rate	Timeout period	TO
1.3	–	25,000 t_{bit}	25,000

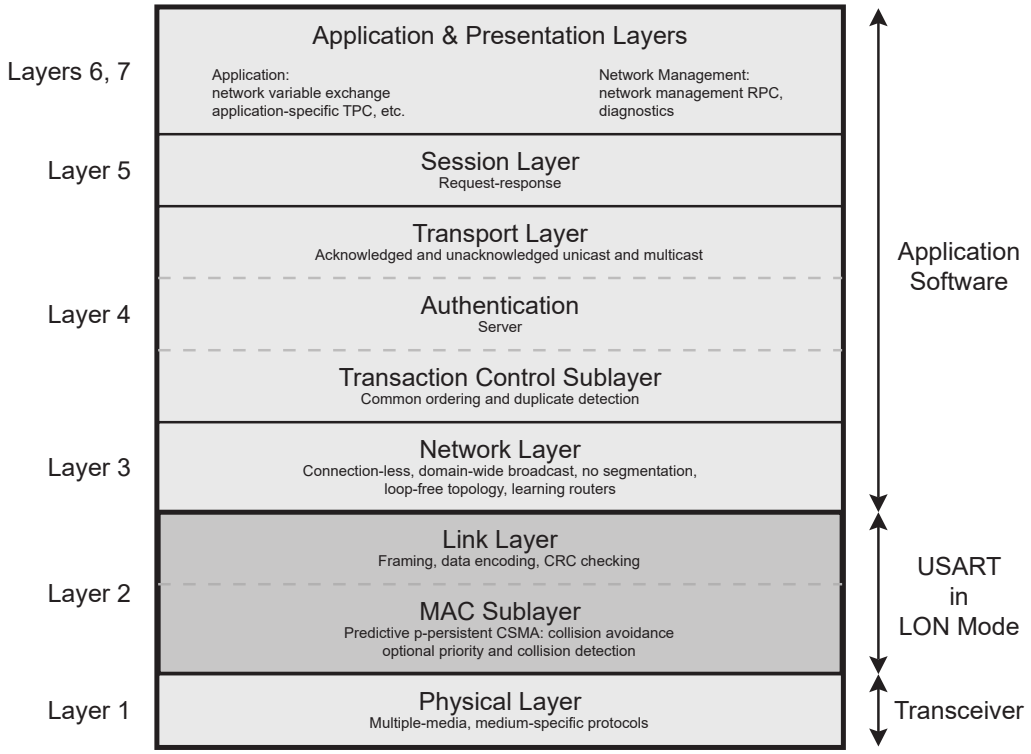
63.7.9. LON Mode

The LON mode provides connectivity to the local operating network (LON).

The LON standard covers all seven layers of the OSI (Open Systems Interconnect) reference model from the physical interfaces such as wired, power line, RF, and IP to the application layer and all layers in between.

The LON mode enables the transmission and reception of Physical Protocol Data Unit (PPDU) frames with minimum intervention from the microprocessor.

Figure 63.54. LON Protocol Layering



The USART configured in LON mode is a full-layer 2 implementation including standard timings handling, framing (transmit and receive PPDU frames), backlog estimation and other features. At the frame encoding/decoding level, differential Manchester encoding is used (also known as CDP). When configured in LON mode, there is no embedded digital line filter, thus the optimal usage is node-to-node communication.

63.7.9.1. Mode of Operation

To configure the USART to act as a LON node, the USART_MODE field of the USART Mode register (FLEX_US_MR) must be set to 0x9.

To avoid unpredictable behavior, any change of the LON node configuration must be preceded by a software reset of the transmitter and the receiver (except the initial node configuration after a hardware reset) and followed by a transmitter/receiver enable. See Receiver and Transmitter Control.

63.7.9.2. Receiver and Transmitter Control

See Receiver and Transmitter Control.

63.7.9.3.Character Transmission

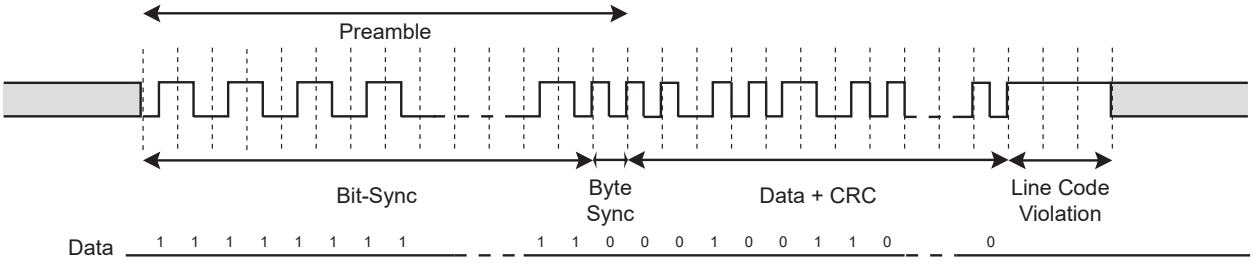
A LON frame is made up of a preamble, a data field (up to 256 bytes) and a 16-bit CRC field. The preamble and CRC fields are automatically generated and the LON node starts the transmission algorithm on a write to LON_L2HDR. See Sending a Frame.

63.7.9.4.Character Reception

When receiving a LON frame, the Receive Holding register (FLEX_US_RHR) is updated upon completed character reception and the RXRDY bit in the Status register rises. If a character is completed while the RXRDY bit is set, the OVRE (Overrun Error) bit is set. The LON preamble field is only used for synchronization, therefore only the Data and CRC fields are transmitted to the Receive Holding register (FLEX_US_RHR). See Sending a Frame.

63.7.9.5.LON Frame

Figure 63.55. LON Framing



63.7.9.5.1.Encoding / Decoding

The USART configured in LON mode encodes transmitted data and decodes received data using differential Manchester encoding. In differential Manchester encoding, a '1' bit is indicated by making the first half of the signal equal to the last half of the previous bit's signal (no transition at the start of the bit-time). A '0' bit is indicated by making the first half of the signal opposite to the last half of the previous bit's signal (a zero bit is indicated by a transition at the beginning of the bit-time). As is the case with normal Manchester encoding, missing transition at the middle of bit-time represents a Manchester code violation.

The FLEX_US_MAN.RXIDLEV bit informs the USART of the receiver line idle state value (receiver line inactive) thus ensuring higher reliability of preamble synchronization. By default, RXIDLEV is set (receiver line is at level 1 when there is no activity).

Differential Manchester encoding is polarity-insensitive.

Figure 63.56. LON PPDU

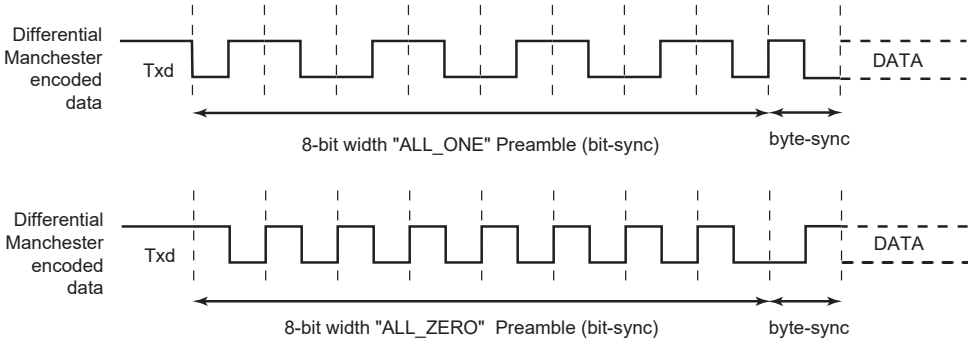
63.7.9.5.2.Preamble Transmission

Each LON frame begins with a preamble of variable length which consists of a bit-sync field and a byte-sync field. The LONPL field of the USART LON Preamble register (FLEX_US_LONPR) defines the preamble length. Note that a preamble length of '0' is not allowed.

The LON implementation allows two different preamble patterns ALL_ONE and ALL_ZERO which can be configured through the TX_PL field of the USART Manchester Configuration register (FLEX_US_MAN). The following figure illustrates and defines the valid patterns.

Other preamble patterns are not supported.

Figure 63.57. Preamble Patterns



63.7.9.5.3.Preamble Reception

LON received frames begin with a preamble of variable length. The receiving algorithm does not check the preamble length, although a minimum of length of 4 bits is required for the receiving algorithm to consider the received preamble as valid.

As is the case with LON preamble transmission, two preamble patterns (ALL_ONE and ALL_ZERO) are allowed and can be configured through the RX_PL field of the USART Manchester Configuration register (FLEX_US_MAN). The above figure illustrates and defines the valid patterns.

Other preamble patterns are not supported.

63.7.9.5.4.Header Transmission

Each LON frame, after sending the preamble, starts with the frame header also called L2HDR according to CEA-709 specification. This header consist of the priority bit, the alternative path bit and the backlog increment. It is the first data to be sent.

In LON mode, the transmitting algorithm starts when FLEX_US_LONL2HDR is written (it is the first data to send).

63.7.9.5.5.Header Reception

Each LON frame, after receiving the preamble, receives the frame header also called L2HDR according to CEA-709 specification. This header consists of the priority bit, the alternative path bit, and the backlog increment.

The frame header is the first received data and the RXRDY bit rises as soon as the frame header as been received and stored in the Receive Holding register (FLEX_US_RHR).

63.7.9.5.6.Data

Data are sent/received serially after the preamble transmission/reception. Data can be either sent/received MSB first or LSB first depending on the MSBF bit value in the USART Mode register (FLEX_US_MR).

63.7.9.5.7.CRC

The two last bytes of LON frames are dedicated to CRC.

When transmitting, the CRC of the frame is automatically generated and sent when expected.

When receiving frames, the CRC is automatically checked and the FLEX_US_CSR.LCRCE flag is set if the calculated CRC does not match the received one. Note that the two received CRC bytes are seen as two additional data from the user point of view.

63.7.9.5.8.End Of Frame

The USART configured in LON mode terminates the frame with a three t_{bit} long Manchester code violation. After sending the last CRC bit, it maintains the data transitionless during three bit periods.

63.7.9.6.LON Operating Modes

63.7.9.6.1.Transmitting/Receiving Modules

According to the LON node configuration and LON network state, the transmitting module is activated if a transmission request has been made and access to the LON bus granted. It returns to idle state once the transmission ends.

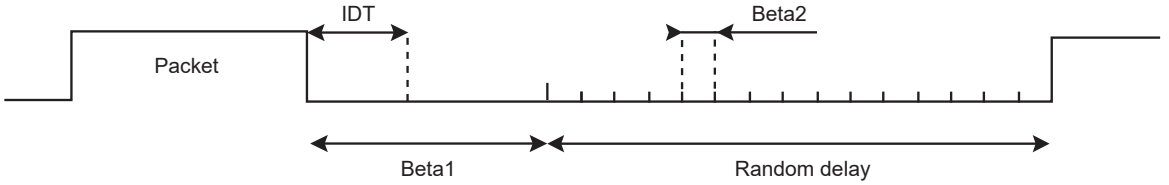
According to the LON node configuration and LON network state, the receiving module is activated if a valid preamble is detected and the transmitting module is not activated.

63.7.9.6.2.comm_type

In CEA-709 standard 2, communication configurations are defined and configurable through the comm_type variable. The comm_type variable value can be set in the USART LON Mode register (FLEX_US_LONMR) through the COMMT bit. The selection of the comm_type determines the MAC behavior in the following ways:

- comm_type = 1:
  - An indeterminate time is defined during the Beta 1 period in which all transitions on the channel are ignored (see figure below).
  - The MAC sublayer ignores collisions occurring during the first 25% of the transmitted preamble. Optionally (according to the FLEX_US_LONMR.CDTAIL bit), it ignores collisions reported following the transmission of the CRC but prior to the end of transmission.
  - If a collision is detected during preamble transmission, the MAC sublayer can terminate the packet if so configured according to the FLEX_US_LONMR.TCOL bit. Collisions detected after the preamble have been sent do not terminate transmission.
- comm_type = 2:
  - No indeterminate time is defined at the MAC sublayer.
  - The MAC sublayer always terminates the packet upon notification of a collision.

Figure 63.58. LON Indeterminate Time



63.7.9.6.3.Collision Detection

As an option of the CEA-709 standard collision detection is supported through an active low Collision Detect (CD) input from the transceiver.

The Collision Detection source can be either external (see I/O Lines Description) or internal. The collision detection source selection is defined through the LCDS bit in the USART LON Mode register (FLEX_US_LONMR).

The Collision Detection feature can be activated through the COLDET bit of the USART LON Mode register. If the collision detection feature is enabled and CD signal goes low for at least half t_{bit} period then a collision is detected and reported as defined in comm_type.

63.7.9.6.4.Collision Detection Mode

As defined in comm_type, if comm_type = 1 the LON node can be configured to either not terminate transmission upon collision notification during preamble transmission or terminate transmission.

The FLEX_US_LONMR.TCOL bit allows to decide whether to terminate transmission or not upon collision notification during preamble transmission.

63.7.9.6.5.Collision Detection after CRC

As defined in comm_type, if comm_type = 1 the LON node can be configured to ignore collisions reported after the CRC has been sent but prior to the end of the frame.

The FLEX_US_LONMR.CDTAIL bit can be used to decide whether such collision notifications must be considered or not.

63.7.9.6.6.Random Number Generation

The Predictive p-persistent CSMA algorithm defined in the CEA-709.1 Standard is based on a random number generation.

This random number is automatically generated by an internal algorithm.

In addition, a USART IC DIFF register (FLEX_US_ICDIFF) is available to avoid that two same chips with the same software generate the same random number after reset. The value of this register is used by the internal algorithm to generate the random number. Therefore, putting a different value here for each chip ensures that the random number generated after a reset at the same time will not be the same. It is recommended to put the chip ID code here.

63.7.9.7.LON Node Backlog Estimation

As defined in CEA-709, the LON node maintains its own backlog estimation. The node backlog estimation is initially set to one, will always be greater than 1 and will never exceed 63. If the node backlog estimation exceeds the maximum backlog value, the backlog value is set to 63 and a backlog overflow error flag is set (LBLOVFE flag).

The node backlog estimation is incremented each time a frame is sent or received successfully. The increment to the backlog is encoded into the link layer header, and represents the number of messages that the packet shall cause to be generated upon reception.

The backlog decrements under one of the following conditions:

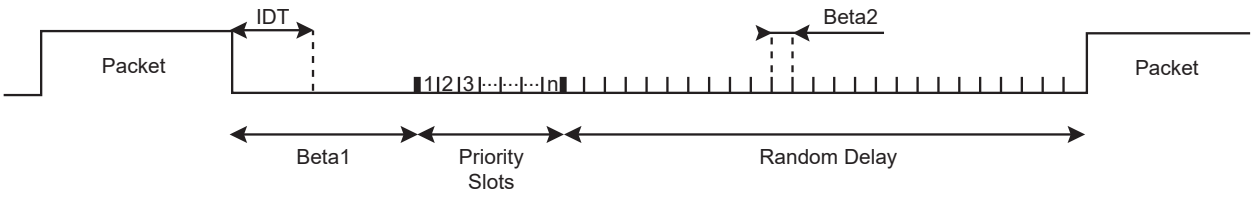
- On waiting to transmit: If Wbase randomizing slots go by without channel activity.
- On receive: If a packet is received with a backlog increment of '0'.
- On transmit: If a packet is transmitted with a backlog increment of '0'.
- On idle: If a packet cycle time expires without channel activity.

63.7.9.7.1.Optional Collision Detection Feature And Backlog Estimation

Each time a frame is transmitted and a collision occurred, the backlog is incremented by 1. In this case, the backlog increment encoded in the link layer is ignored.

63.7.9.8.LON Timings

Figure 63.59. LON Timings



63.7.9.8.1.Beta2

A node wishing to transmit generates a random delay T. This delay is an integer number of randomizing slots of duration Beta2.

The beta2 length (in t_{bit}) is configurable through FLEX_US_FIDI. Note that a length of '0' is not allowed.

63.7.9.8.2.Beta1 Tx/Rx

Beta1 is the period immediately following the end of a packet cycle (see the above figure). A node attempting to transmit monitors the state of the channel, and if it detects no transmission during the Beta1 period, it determines the channel to be idle.

The Beta1 value is different depending on the previous packet type (received packet or transmitted packet).

Beta1Rx and Beta1Tx length can be configured respectively through the USART LON Beta1 Rx register (FLEX_US_LONB1RX) and the USART LON Beta1 Tx register (FLEX_US_LONB1TX). Note that a length of '0' is not allowed.

63.7.9.8.3.Pcycle Timer

The packet cycle timer is reset to its initial value whenever the backlog is changed. It is started (begins counting down at its current value) whenever the MAC layer becomes idle. An idle MAC layer is defined as:

- Not receiving
- Not transmitting,
- Not waiting to transmit,
- Not timing Beta1,
- Not waiting for priority slots, and not waiting for the first Wbase randomizing window to complete.

On transition from idle to either transmit or receive, the packet cycle timer is halted.

The pcycle timer value can be configured in FLEX_US_TTGR. Note that '0' value is not allowed.

63.7.9.8.4.Wbase

The wbase timer represents the base windows size. Its duration, derived from Beta2, equals 16 Beta2 slots.

63.7.9.8.5.Priority Slots

On a channel by channel basis, the protocol supports optional priority. Priority slots, if any, follow immediately after the Beta1 period that follows the transmission of a packet (see the above figure). The number of priority slots per channel ranges from 0 to 127.

The number of priority slots in the LON network configuration is defined through the PSNB field of the USART LON Priority register (FLEX_US_LONPRIO). And the priority slot affected to the LON node, if any, is defined through the FLEX_US_LONPRIO.NPS field.

63.7.9.8.6.Indeterminate Time

See comm_type.

Like Beta1, the IDT value is different depending on whether the previous frame was transmitted or received.

IDTRx and IDTTx can be configured respectively through the USART LON IDT Rx register (FLEX_US_LONIDTRX) and the USART LON IDT Tx register (FLEX_US_LONIDTTX).

63.7.9.8.7.End of Frame Condition

The USART configured in LON mode terminates the frame with a three t_{bit} long Manchester code violation. After sending the last CRC bit, it maintains the data transitionless during three bit periods.

While receiving data, the USART configured in LON mode detects an end of frame condition after a t_{eof} transitionless Manchester code violation. The EOFS field in the USART LON Mode Register can configure t_{eof}.

63.7.9.9.LON Errors

All these flags can be read in the LON Channel Status register (FLEX_US_CSR) and generate interrupts if configured in the LON Interrupt Enable register (FLEX_US_IER).

These flags can be reset by writing a one to the FLEX_US_CR.RSTSTA bit.

63.7.9.9.1.Underrun Error

If the USART is in LON mode and if a character is sent while the Transmit Holding register (FLEX_US_THR) is empty, the UNRE bit flag is set.

63.7.9.9.2.Collision Detection

The LCOL flag is set whenever a valid collision has been detected and the LON node is configured to report it (see Collision Detection).

63.7.9.9.3.LON Frame Early Termination

The LFET flag is set whenever a LON frame has been terminated early due to collision detection.

63.7.9.9.4.Reception Error

The LCRCE flag is set if the received frame has an erroneous CRC and the flag LSFE is set if the received frame is too short (LON frames must be at least 8 bytes long).

These flags can be read in FLEX_US_CSR.

63.7.9.9.5.Backlog Overflow

The LBLOVFE flag is set if the LON node backlog estimation goes over 63, which is the maximum backlog value.

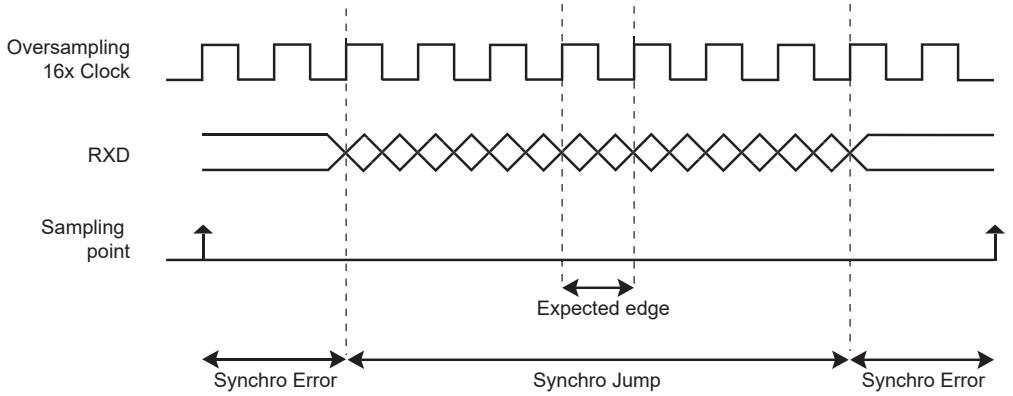
63.7.9.10.Drift Compensation

It may happen that while receiving a frame, the baud rate used by the sender is not exactly the one expected, due to sender clock drifting for instance. In such case, the hardware drift compensation algorithm is used to recover up to 16% clock drift (expected baud rate ±16% is supported).

Drift compensation is available only in 16X Oversampling mode. To enable the hardware system, the DRIFT bit of the FLEX_US_MAN register must be set. If the RXD edge is between one and three 16X clock cycles away from the expected edge, then the period is shortened or lengthened accordingly to center the RXD edge.

Drift compensation hardware feature allows up to 16% clock drift to be handled, provided system clock is fast enough compared to the selected baud rate.

Figure 63.60. Bit Resynchronization



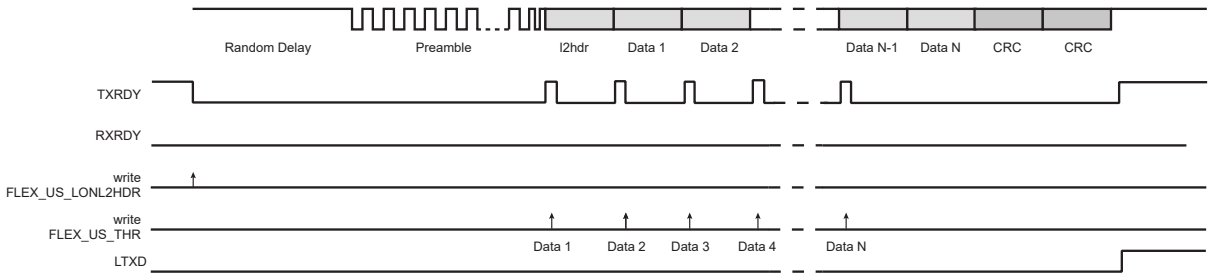
63.7.9.11.LON Frame Handling

63.7.9.11.1.Sending a Frame

1. Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.

2. Write FLEX_US_MR.USART_MODE to select the LON mode configuration.
3. Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
4. Write COMMT, COLDET, TCOL, CDTAIL, RDMNBM and DMAM in FLEX_US_LONMR to configure the LON operating mode.
5. Write BETA2, BETA1TX, BETA1RX, PCYCLE, PSNB, NPS, IDTTX and ITDRX respectively in FLEX_US_FIDI, FLEX_US_LONB1TX, FLEX_US_LONB1RX, FLEX_US_TTGR, FLEX_US_LONPRIO, FLEX_US_LONIDTTX and FLEX_US_LONIDTRX to set the LON network configuration.
6. Write FLEX_US_MAN.TX_PL to select the preamble pattern to use.
7. Write LONPL and LONDL in FLEX_US_LONPR and FLEX_US_LONDL to set the frame transfer.
8. Check that FLEX_US_CSR.TXRDY is set to 1.
9. Write FLEX_US_LONL2HDR to send the header.
10. Wait until FLEX_US_CSR.TXRDY rises.
11. Write FLEX_US_THR.TCHR to send a byte.
12. If all the data have not been written, repeat the two previous steps.
13. Wait until FLEX_US_CSR.LTXD rises.
14. Check the LON errors.

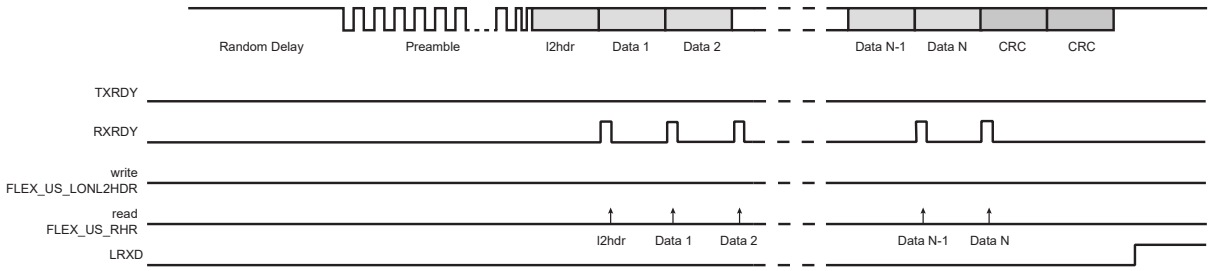
Figure 63.61. Tx Frame



63.7.9.11.2.Receiving a Frame

1. Write FLEX_US_CR.TXEN and FLEX_US_CR.RXEN to enable both the transmitter and the receiver.
2. Write FLEX_US_MR.USART_MODE to select the LON mode configuration.
3. Write FLEX_US_BRGR.CD and FLEX_US_BRGR.FP to configure the baud rate.
4. Write COMMT, COLDET, TCOL, CDTAIL, RDMNBM and DMAM in FLEX_US_LONMR to configure the LON operating mode.
5. Write BETA2, BETA1TX, BETA1RX, PCYCLE, PSNB, NPS, IDTTX and ITDRX respectively in FLEX_US_FIDI, FLEX_US_LONB1TX, FLEX_US_LONB1RX, FLEX_US_TTGR, FLEX_US_LONPRIO, FLEX_US_LONIDTTX and FLEX_US_LONIDTRX to set the LON network configuration.
6. Write FLEX_US_MAN.RXIDLEV and FLEX_US_MAN.RX_PL to indicate the receiver line value and select the preamble pattern to use.
7. Wait until FLEX_US_CSR.RXRDY rises.
8. Read FLEX_US_RHR.RCHR.
9. If all the data and the two CRC bytes have not been read, repeat the two previous steps.
10. Wait until FLEX_US_CSR.LRXD rises.
11. Check the LON errors.

Figure 63.62. Rx Frame



63.7.9.12.LON Frame Handling with the Peripheral DMA Controller

The USART can be used in association with the DMA Controller in order to transfer data directly into/from the on- and off-chip memories without any processor intervention.

The DMA uses the trigger flags, TXRDY and RXRDY, to write or read into the USART. The DMA 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 DMA in the USART is always a byte.

63.7.9.12.1.Configuration

The user can choose between two DMA modes by the DMAM bit in the LON Mode register

(FLEX_US_LONMR):

- DMAM = 1: The LON frame data length (DATAL) is stored in the WRITE buffer and it is written by the DMA in the Transmit Holding register FLEX_US_THR (instead of the LON Data Length register FLEX_US_LONDL).
- DMAM = 0: The LON frame data length (DATAL) is not stored in the WRITE buffer and it must be written by the user in the LON Data Length Register (FLEX_US_LONDL).

In both DMA modes, L2HDR is considered as a data and its value must be stored in the WRITE buffer as the first data to write.

Figure 63.63. DMAM = 1

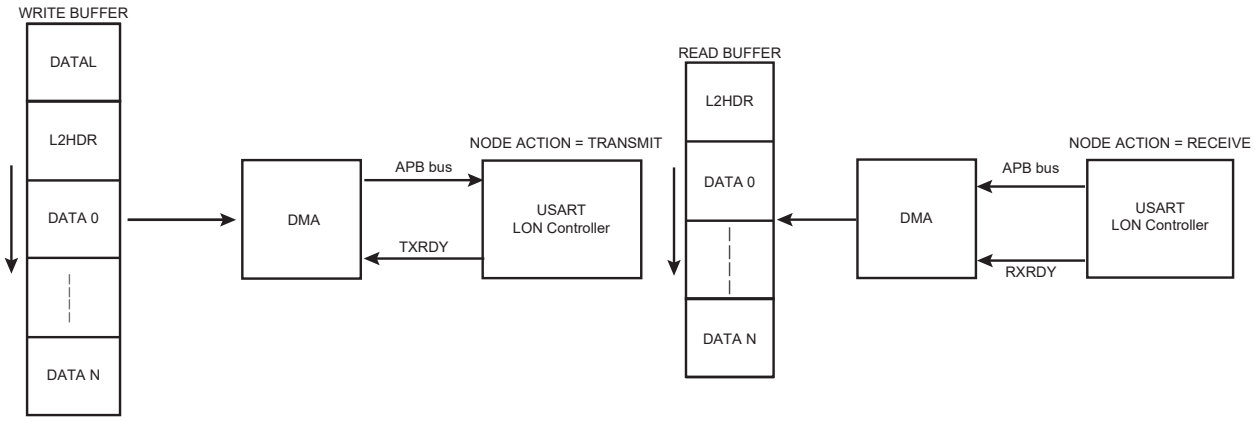
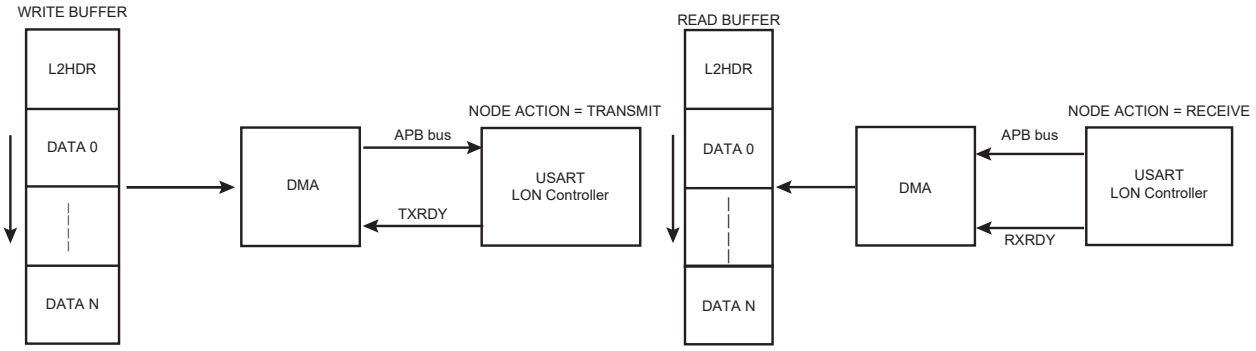


Figure 63.64. DMAM = 0



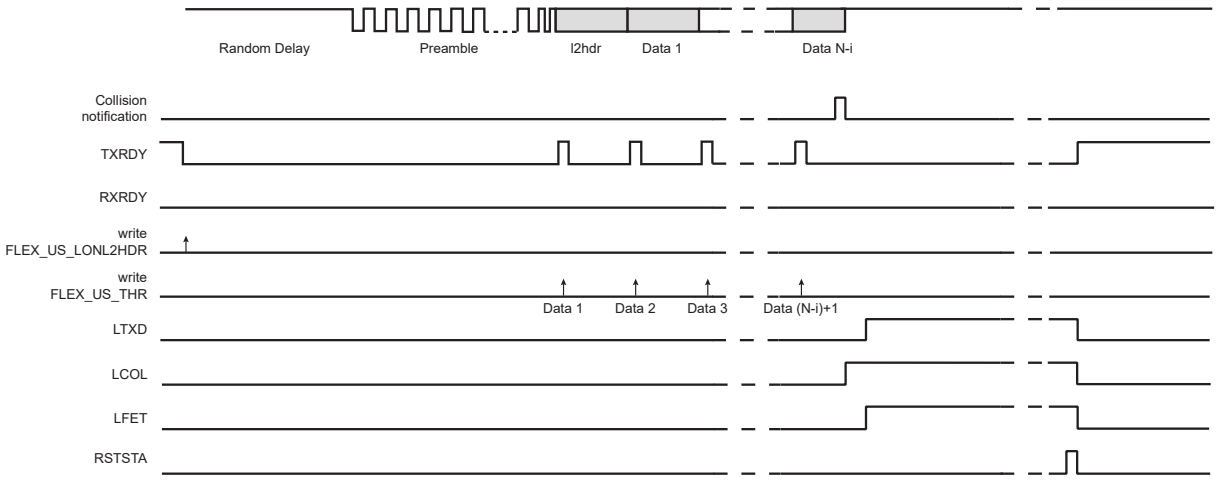
63.7.9.12.2.DMA and Collision Detection

As explained in section comm_type, depending on LON configuration the transmission may be terminated early upon collision notification which means that the DMA transfer may be stopped before its end.

In case of early end of transmission due to collision detection, the USART in LON mode acts as follows:

- Send the end of frame trigger.
- Hold down TXRDY, thus avoiding any additional DMA transfer.
- Set the LTXD, LCOL and LFET flags in FLEX_US_CSR.
- Wait for the application to reconfigure the DMA.
- Wait until the LCOL and LFET flags are cleared through the FLEX_US_CR.RSTSTA bit (it will release the TXRDY signal).

Figure 63.65. DMA, Collision and Early Frame Termination



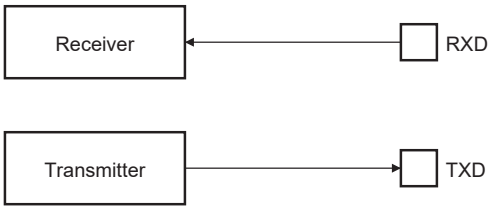
63.7.10. Test Modes

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.

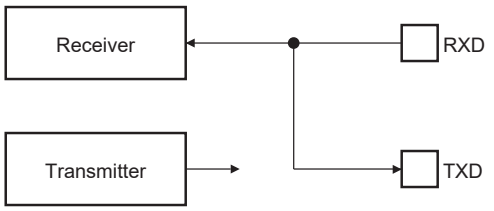
Figure 63.66. Normal Mode Configuration



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.

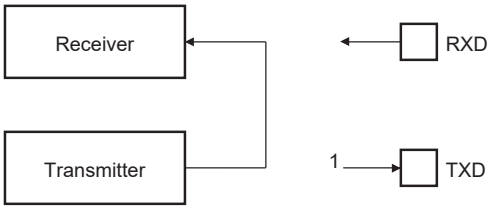
Figure 63.67. Automatic Echo Mode Configuration



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.

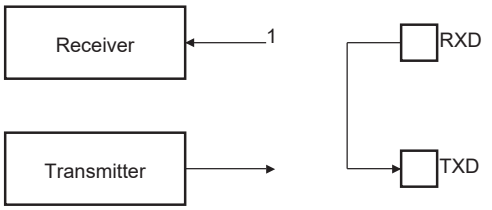
Figure 63.68. Local Loopback Mode Configuration



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 63.69. Remote Loopback Mode Configuration



63.7.11. USART FIFOs

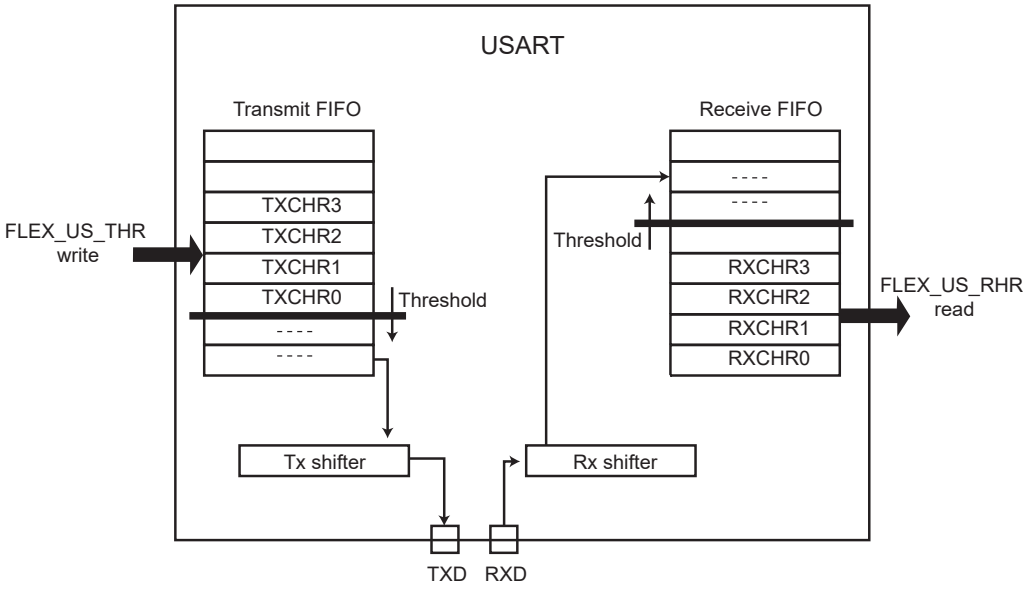
63.7.11.1.Overview

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 16-data Transmit FIFO and a 16-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 63.70. USART FIFOs Block Diagram



63.7.11.2.Sending Data with FIFO Enabled

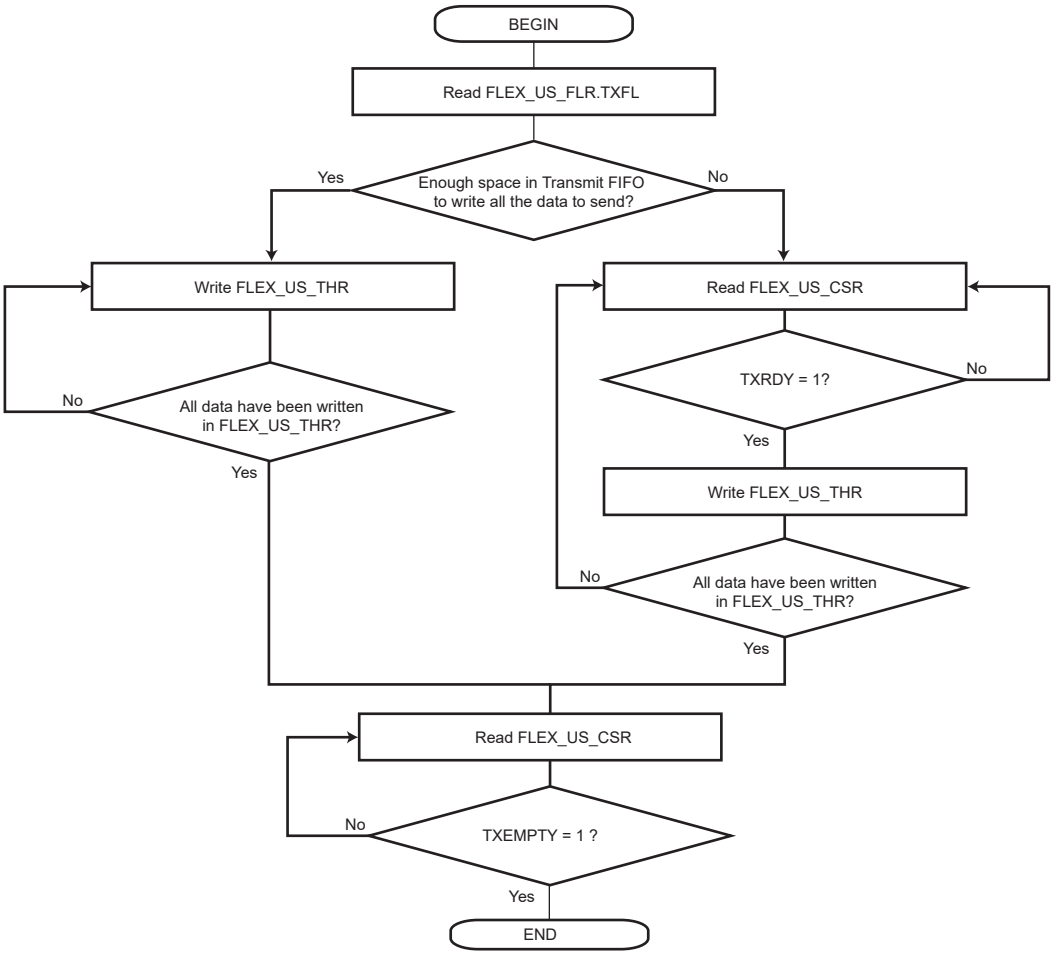
When the Transmit FIFO is enabled, write access to FLEX_US_THR loads the Transmit FIFO.

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 63.71. Sending Data with FIFO Enabled

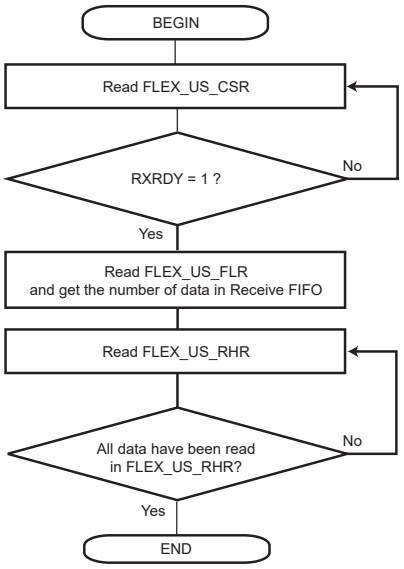


63.7.11.3.Receiving Data with FIFO Enabled

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 63.72. 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 63.73. TXRDY Behavior for Single Data Access and TXRDYM = 0

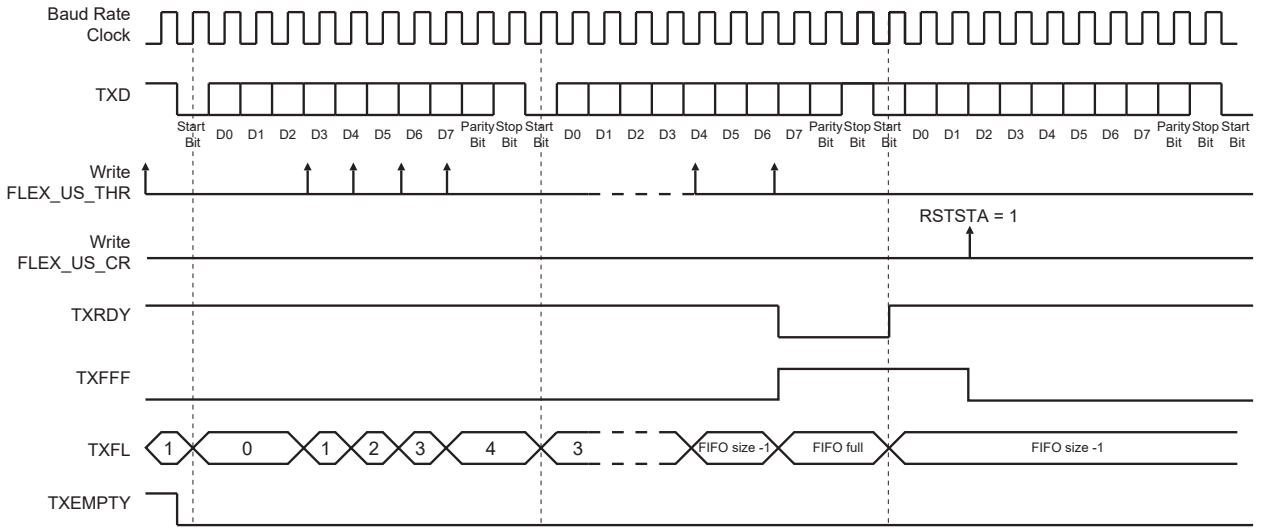
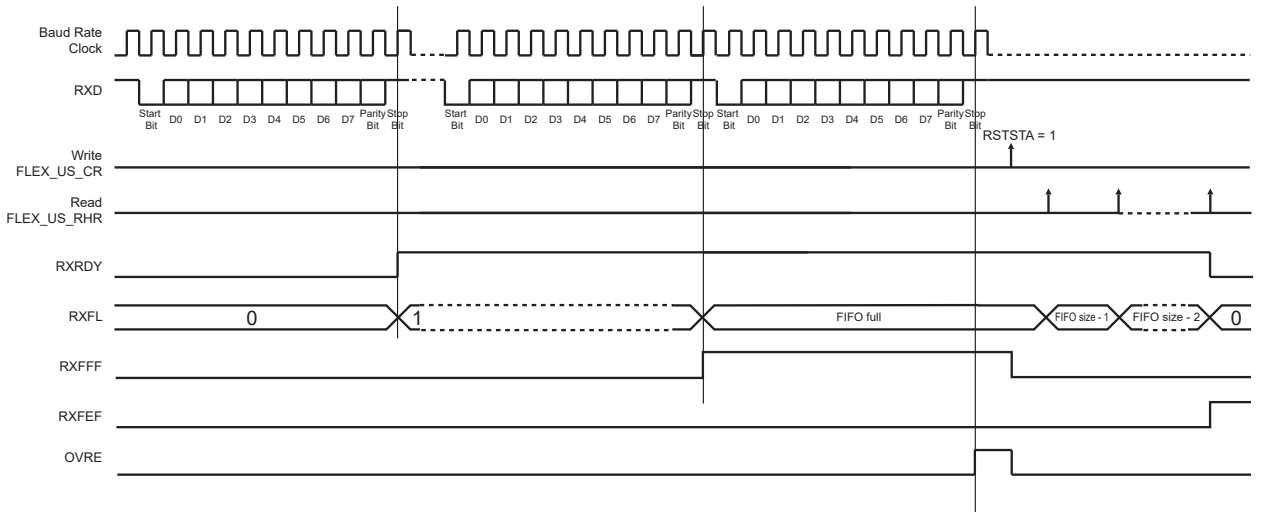


Figure 63.74. RXRDY Behavior for Single Data Access and RXRDYM = 0



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 LON mode, 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.DMA

The DMA 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 LON mode, 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.DMA

The DMA 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 DMA channels, etc., without any risk.

- LON Mode:

If a frame is terminated early due to collision, 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 DMA 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 Overflow/Underflow Error

If the Transmit FIFO is full and a write access is performed on FLEX_US_THR, it generates a Transmit FIFO overflow 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 overflow error is generated and FLEX_US_FESR.TXFPTEF is set.

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 underflow error is generated and FLEX_US_FESR.RXFPTEF is set.

No 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 such error occurs, other FIFO flags may not behave as expected; their states must be ignored.

If a Transmit FIFO overflow error occurs, a transmitter reset must be performed using FLEX_US_CR.RSTTX. If a Receive underflow 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 LON Mode Register
- USART LON Beta1 Tx Register
- USART LON Beta1 Rx Register
- USART LON Priority Register
- USART LON IDT Tx Register
- USART LON IDT Rx Register
- USART IC DIFF 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.

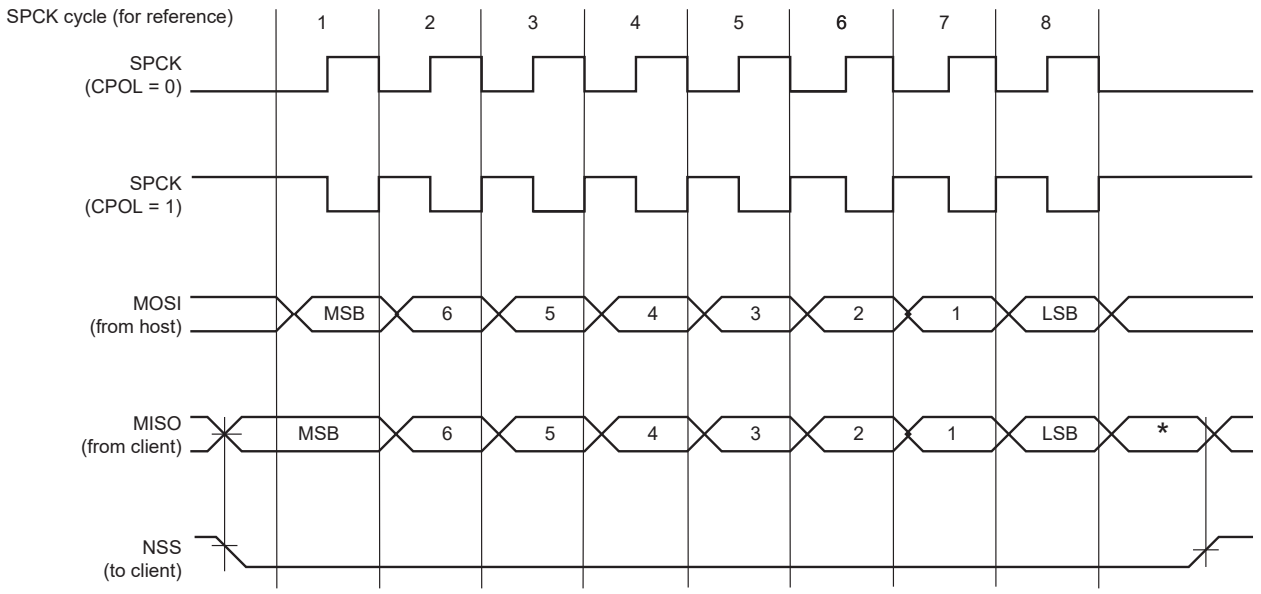
The following table shows the four modes and corresponding parameter settings.

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

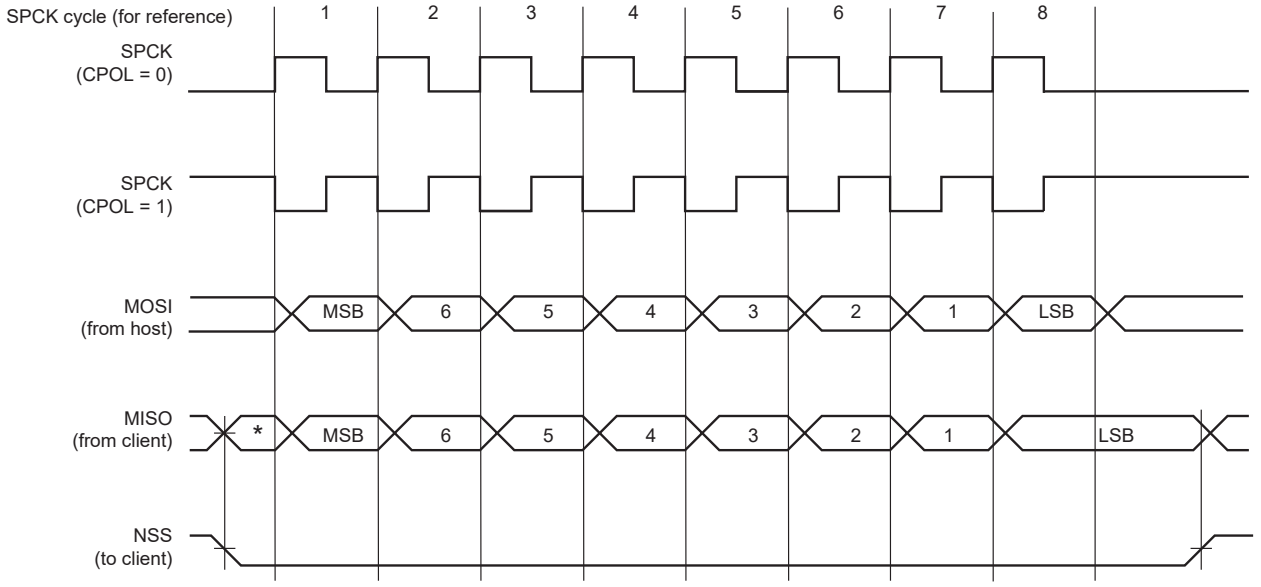
The following figures show examples of data transfers.

Figure 63.75. SPI Transfer Format (NCPHA = 1, 8 bits per transfer) Modes 0 and 2



* Not defined.

Figure 63.76. SPI Transfer Format (NCPHA = 0, 8 bits per transfer) Modes 1 and 3



* Not defined.

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.

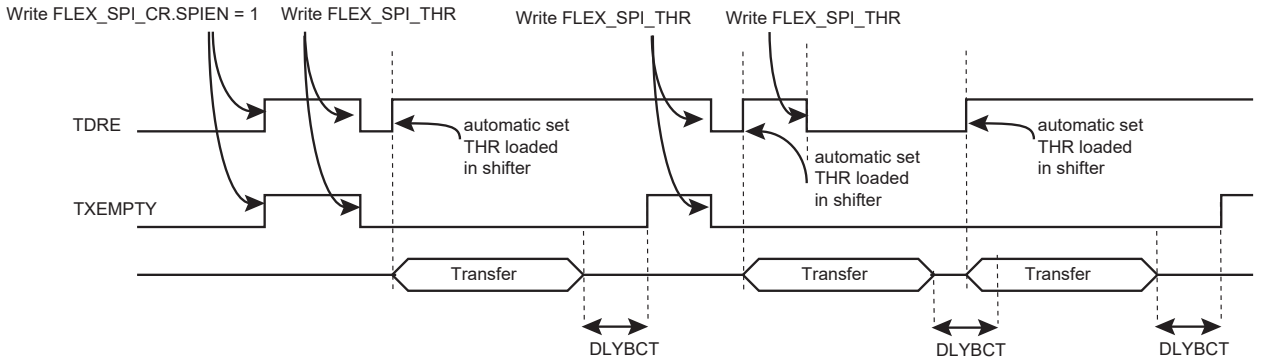
As soon as the FLEX_SPI_TDR is written, the Transmit Data Register Empty (TDRE) flag in FLEX_SPI_SR is cleared. When the data written in FLEX_SPI_TDR is loaded into the shift register, the FLEX_SPI_SR.TDRE flag is set. The TDRE bit is used to trigger the Transmit DMA channel (see figure below).

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 63.77. TDRE and TXEMPTY Flag Behavior



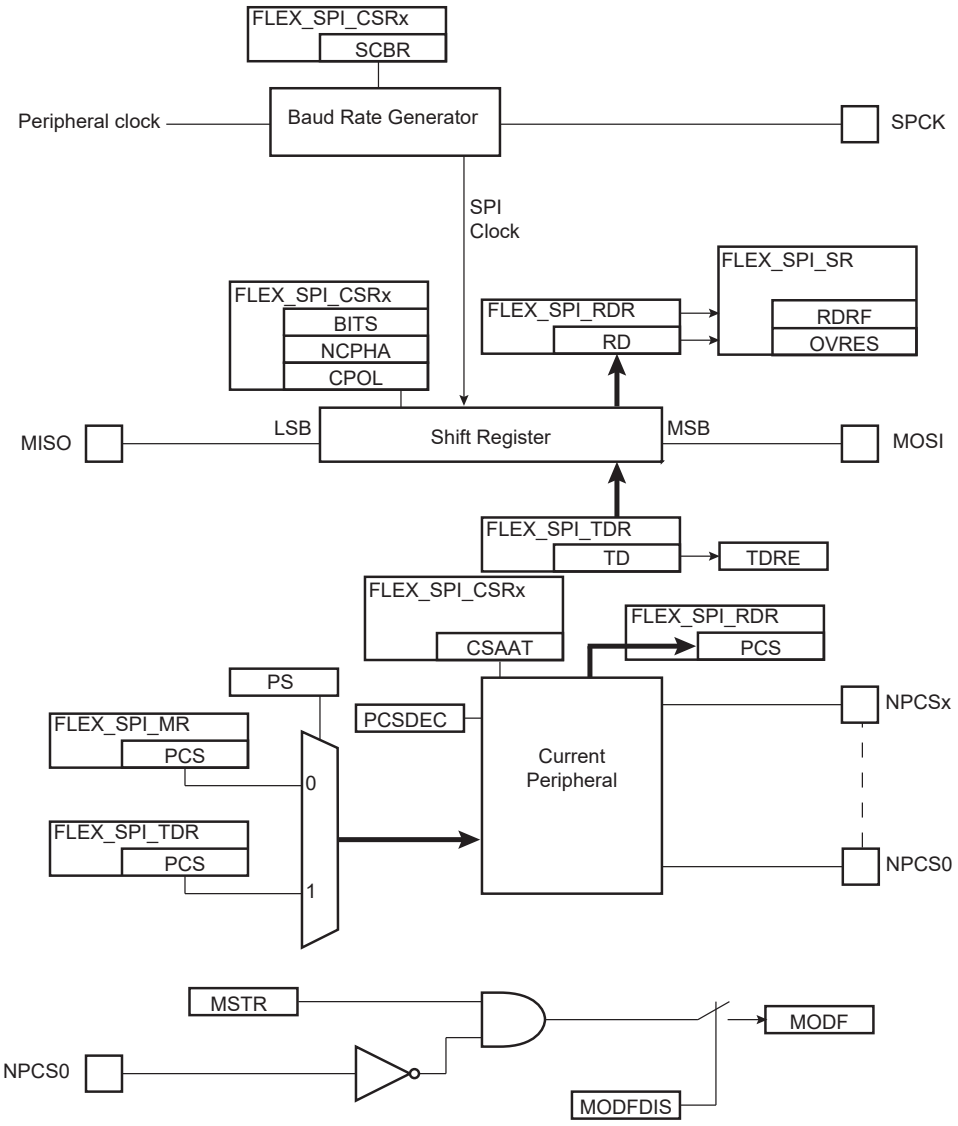
The transfer of received data from the shift register to FLEX_SPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in FLEX_SPI_SR. When the received data are read, the RDRF bit is cleared.

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.

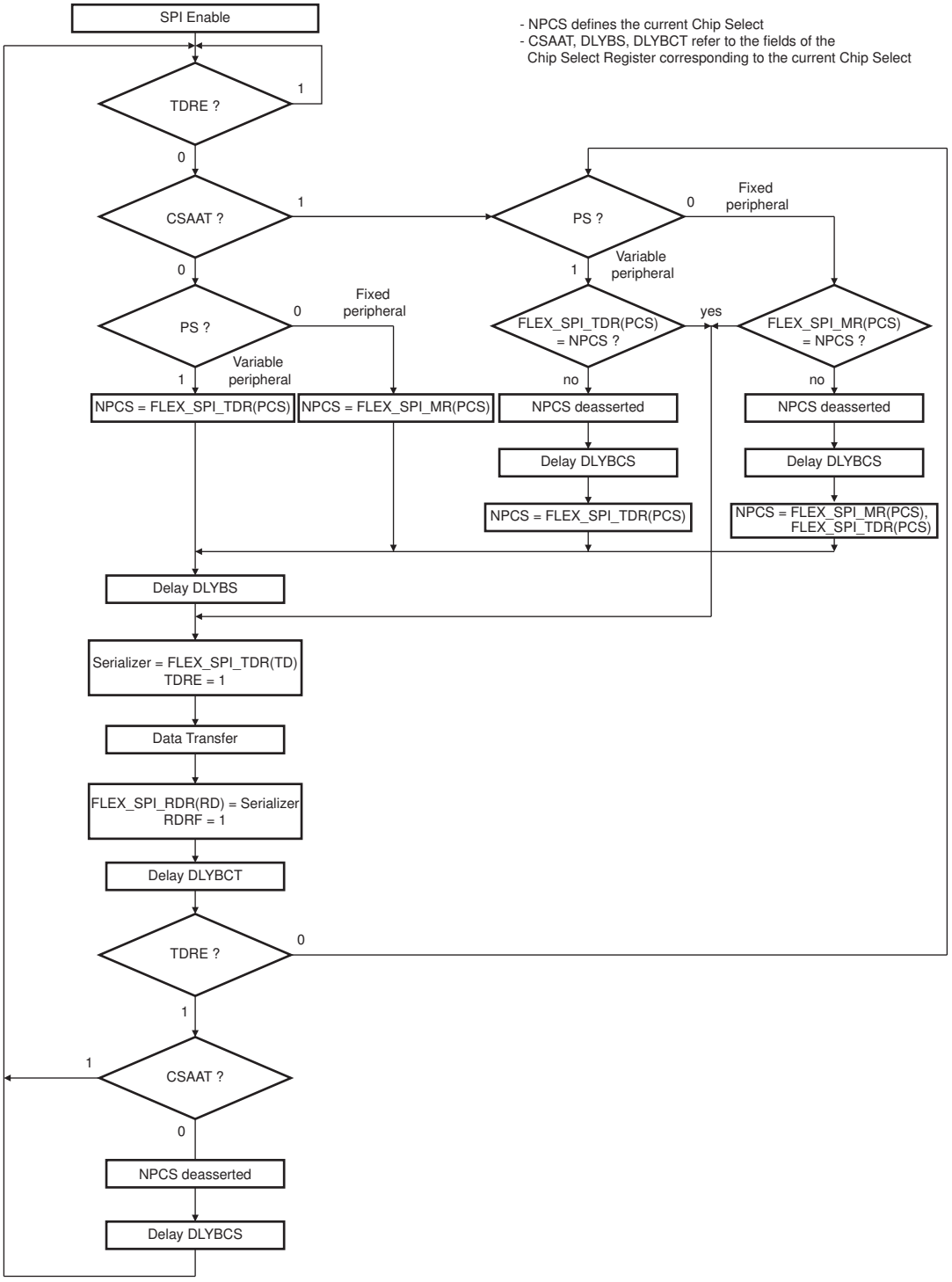
63.8.3.1.Host Mode Block Diagram

Figure 63.78. Host Mode Block Diagram



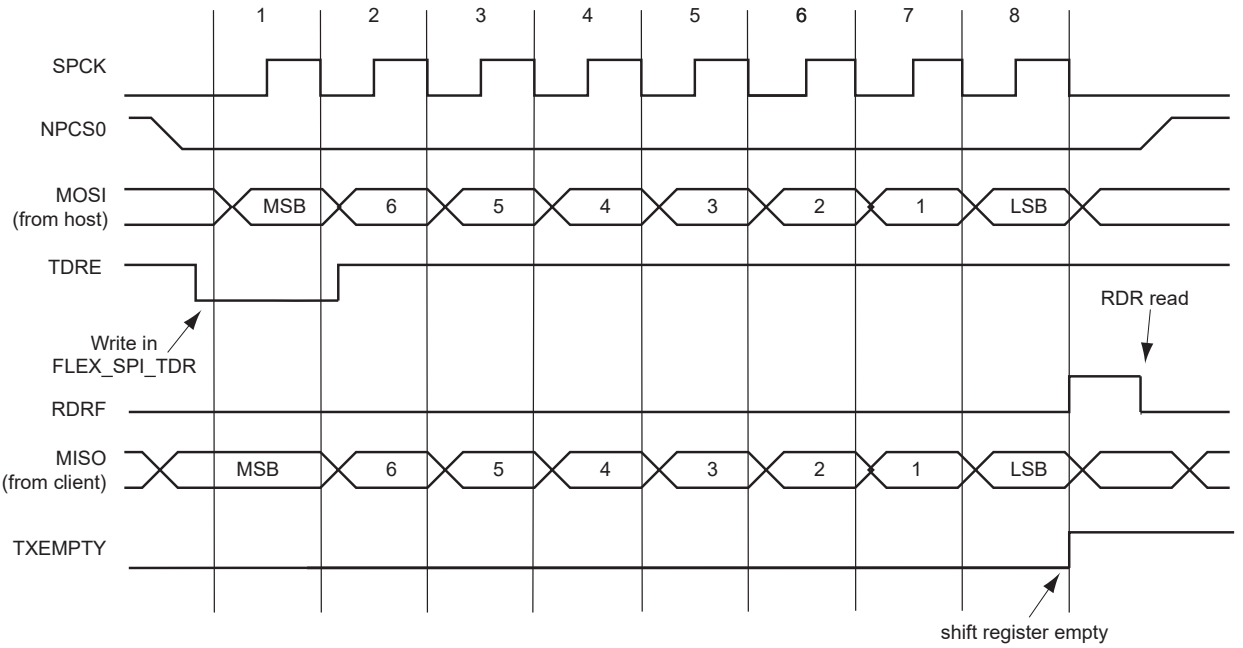
63.8.3.2.Host Mode Flowchart

Figure 63.79. 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 63.80. 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

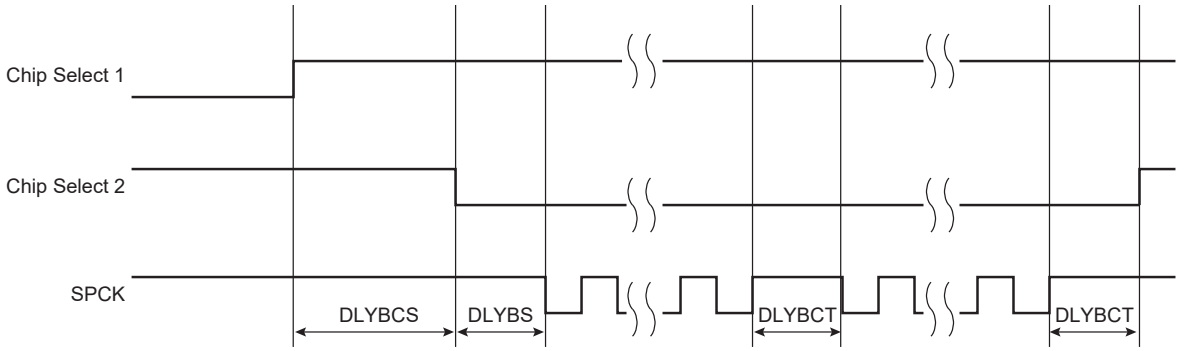
The figure below shows a chip select transfer change and consecutive transfers on the same chip select. Three delays can be programmed to modify the transfer waveforms:

- 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 “SPI Timings” in the section “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 “SPI Timings” in the section “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 63.81. 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.

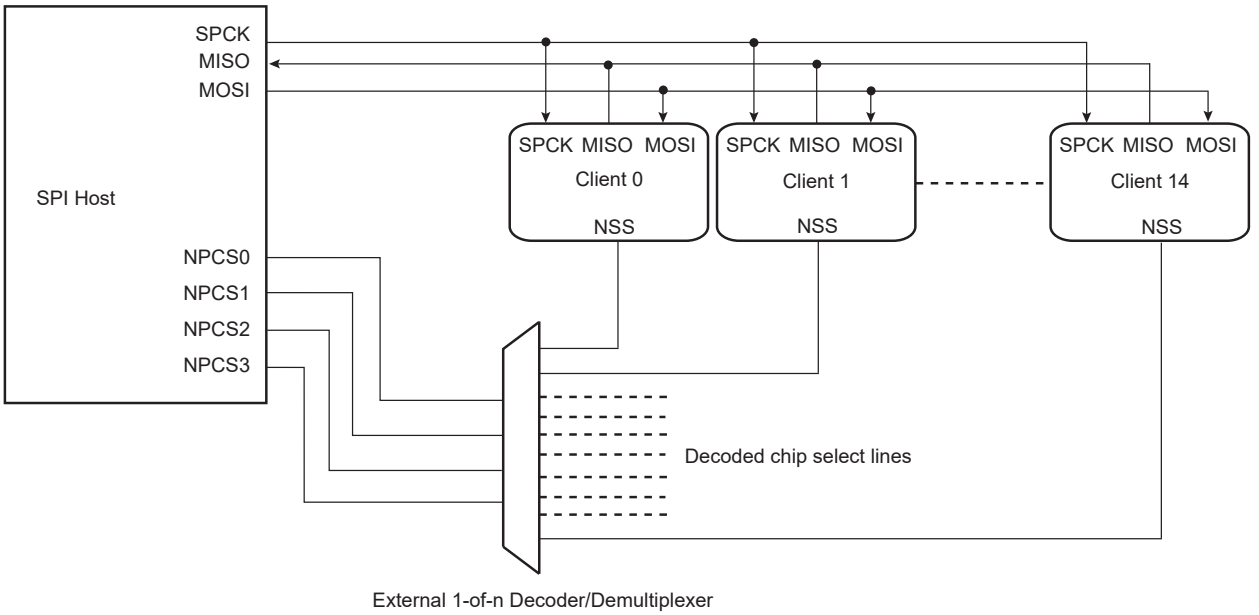
When operating with decoding, the SPI directly outputs the value defined by the PCS field on the NPCS lines of either FLEX_SPI_MR or FLEX_SPI_TDR (depending on PS).

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 63.82. 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

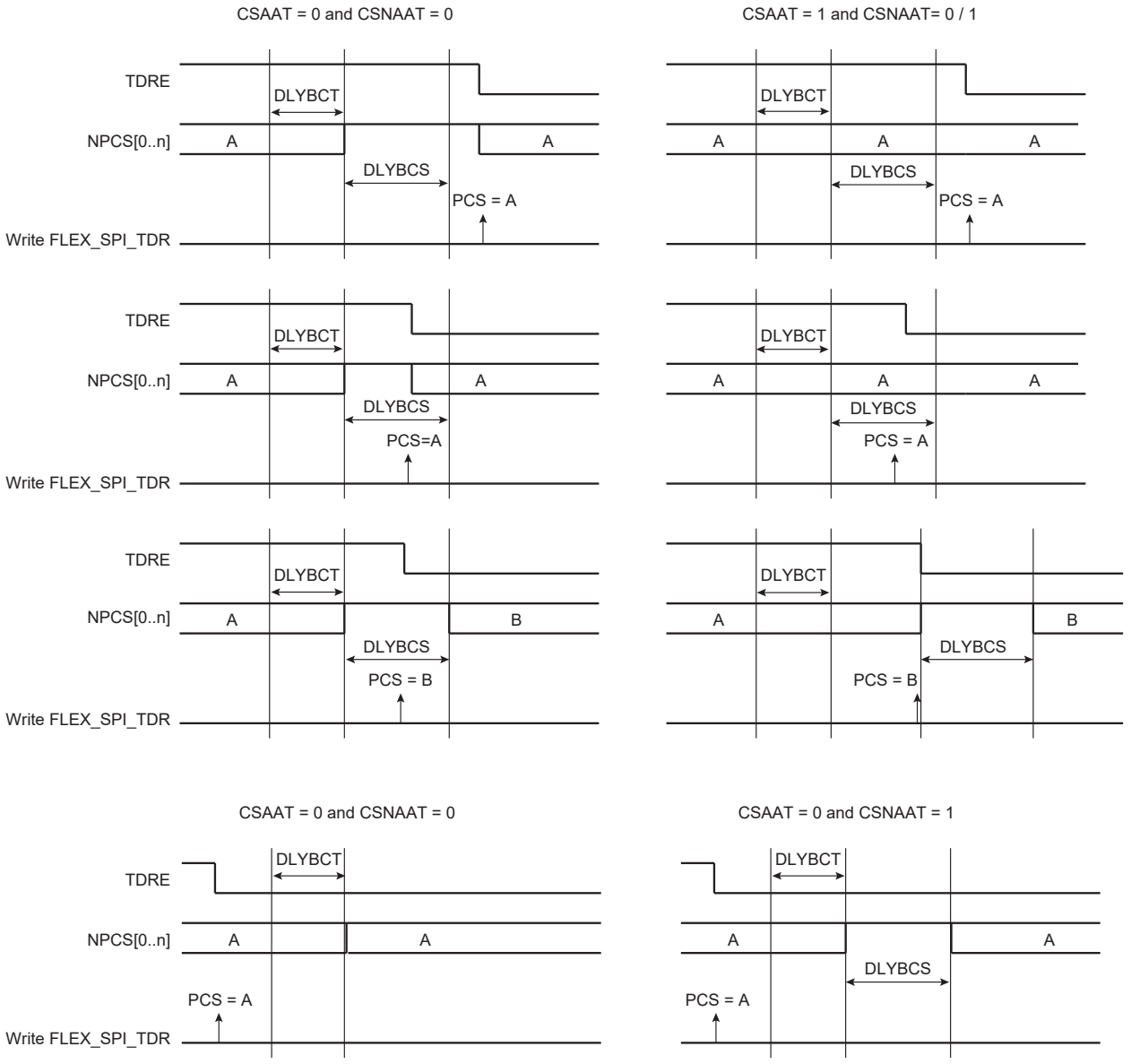
DMA provides faster reloads of FLEX_SPI_TDR compared to software. However, depending on the system activity, it is never sure that FLEX_SPI_TDR is written with the next data before the end of the current transfer. Consequently, a data can be lost by the deassertion of the NPCS line for SPI client peripherals 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 63.83. 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.

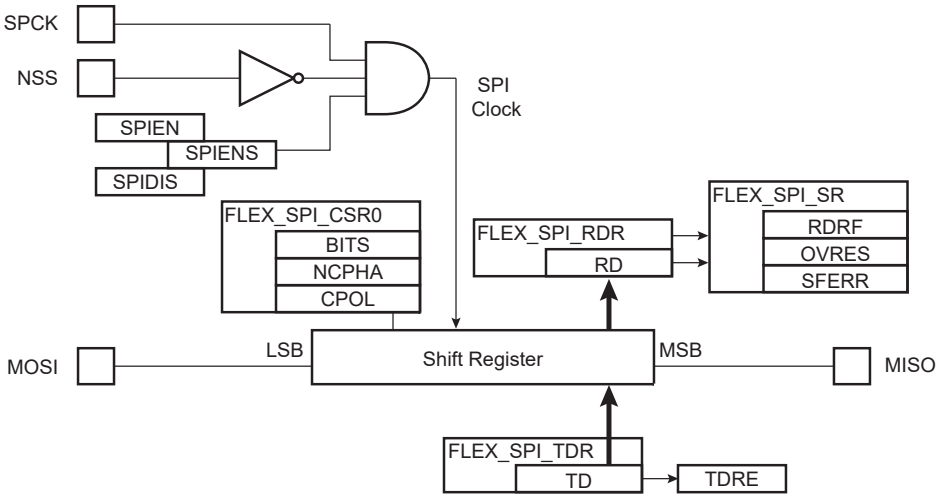
Then, a new data is loaded in the shift register from FLEX_SPI_TDR. If no character is ready to be transmitted, i.e., no character has been written in FLEX_SPI_TDR since the last load from FLEX_SPI_TDR to the shift register, FLEX_SPI_TDR is retransmitted. In this case the Underrun Error Status flag (UNDES) 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 63.84. 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).

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:

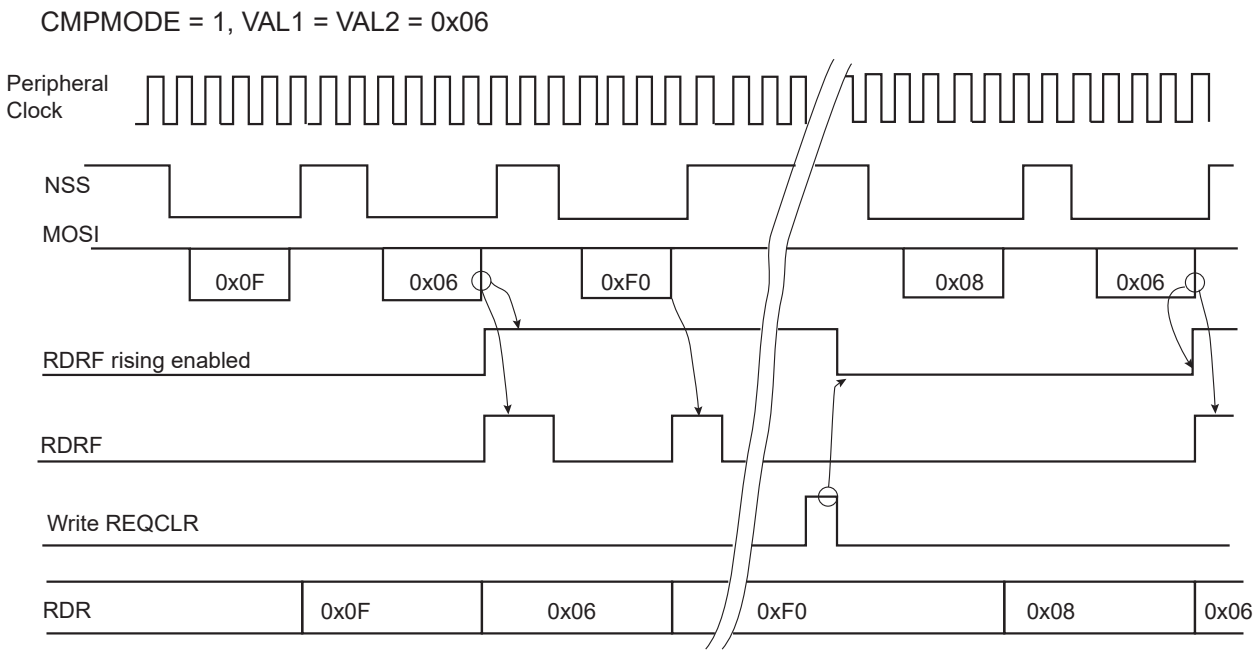
- If VAL1 equals VAL2, then the comparison is performed on a single value and the flag is set to 1 if the received character equals VAL1.
- If VAL1 is strictly lower than VAL2, then any value between VAL1 and VAL2 sets the CMP flag.
- If VAL1 is strictly higher than VAL2, then the flag CMP is set to 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 63.85. Receive Data Register Management



63.8.6. SPI FIFOs

63.8.6.1. Overview

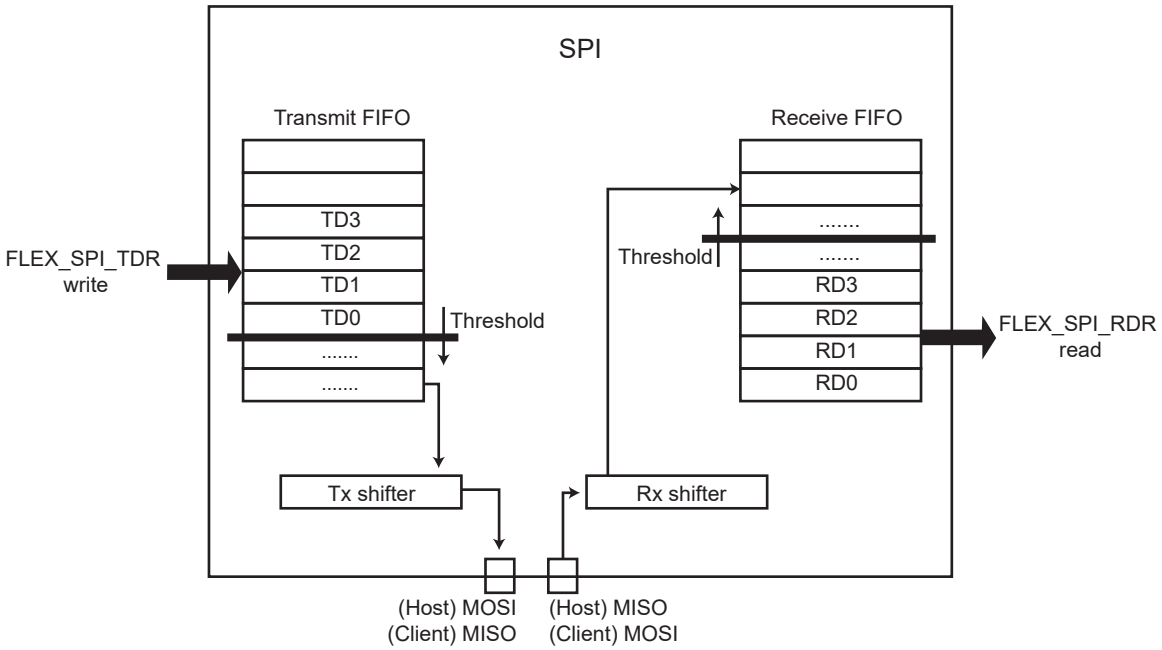
The SPI includes two FIFOs which can be enabled/disabled using the FLEX_SPI_CR.FIFOEN/FIFODIS. The SPI module must be disabled before enabling or disabling the SPI FIFOs (FLEX_SPI_CR.SPIDIS).

Writing FLEX_SPI_CR.FIFOEN to '1' enables a 16-data Transmit FIFO and a 16-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 63.86. SPI FIFOs Block Diagram



63.8.6.2.Sending Data with FIFO Enabled

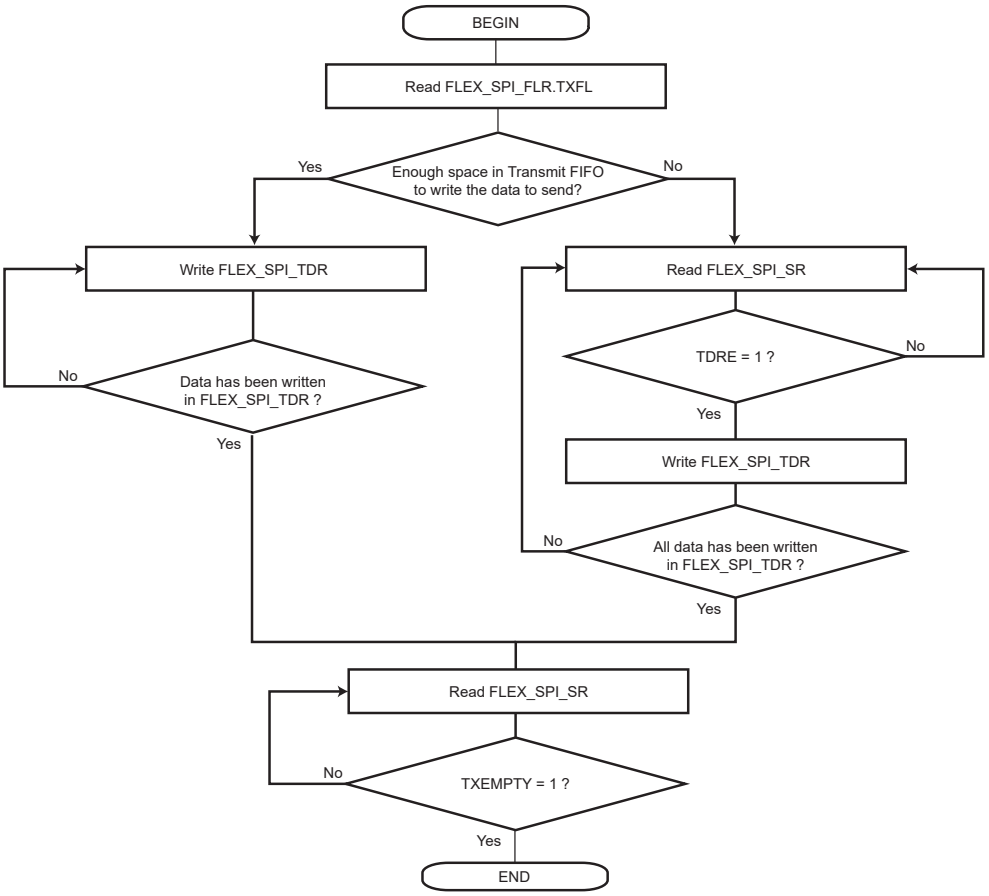
When the Transmit FIFO is enabled, write access to FLEX_SPI_TDR loads the Transmit FIFO.

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 63.87. Sending Data with FIFO Enabled

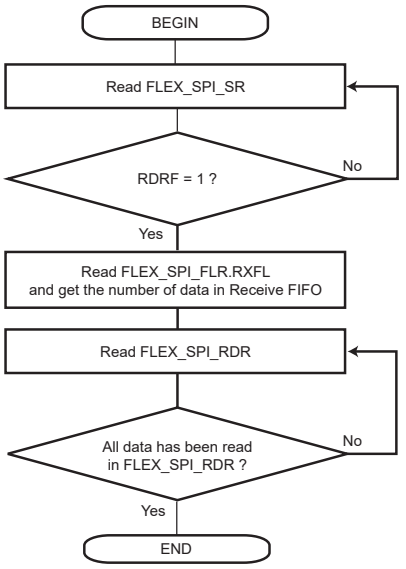


63.8.6.3.Receiving Data with FIFO Enabled

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 63.88. Receiving Data with FIFO Enabled



63.8.6.4.Clearing/Flushing FIFOs

Each FIFO can be cleared/flushed using FLEX_SPI_CR.TXFCLR/RXFCLR.

63.8.6.5.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 63.89. TDRE Behavior for Single Data Access and TXRDYM = 0

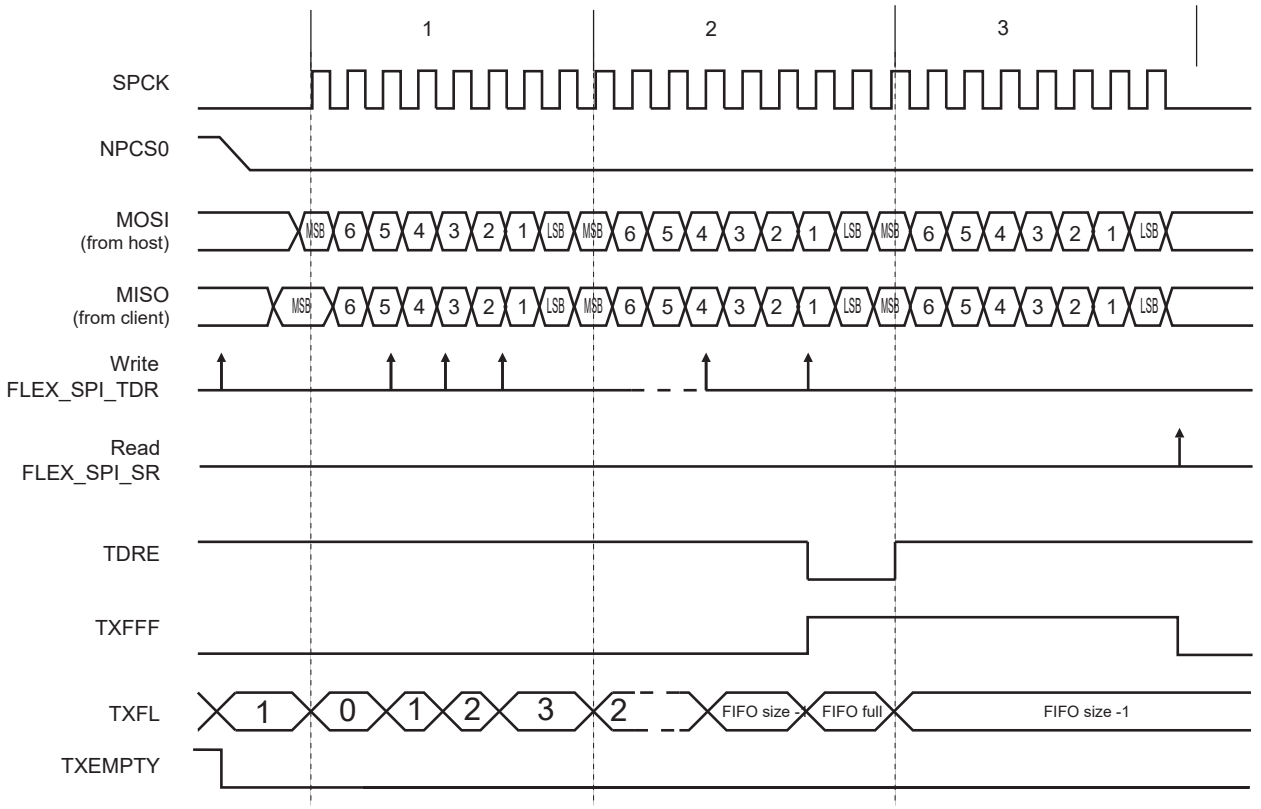
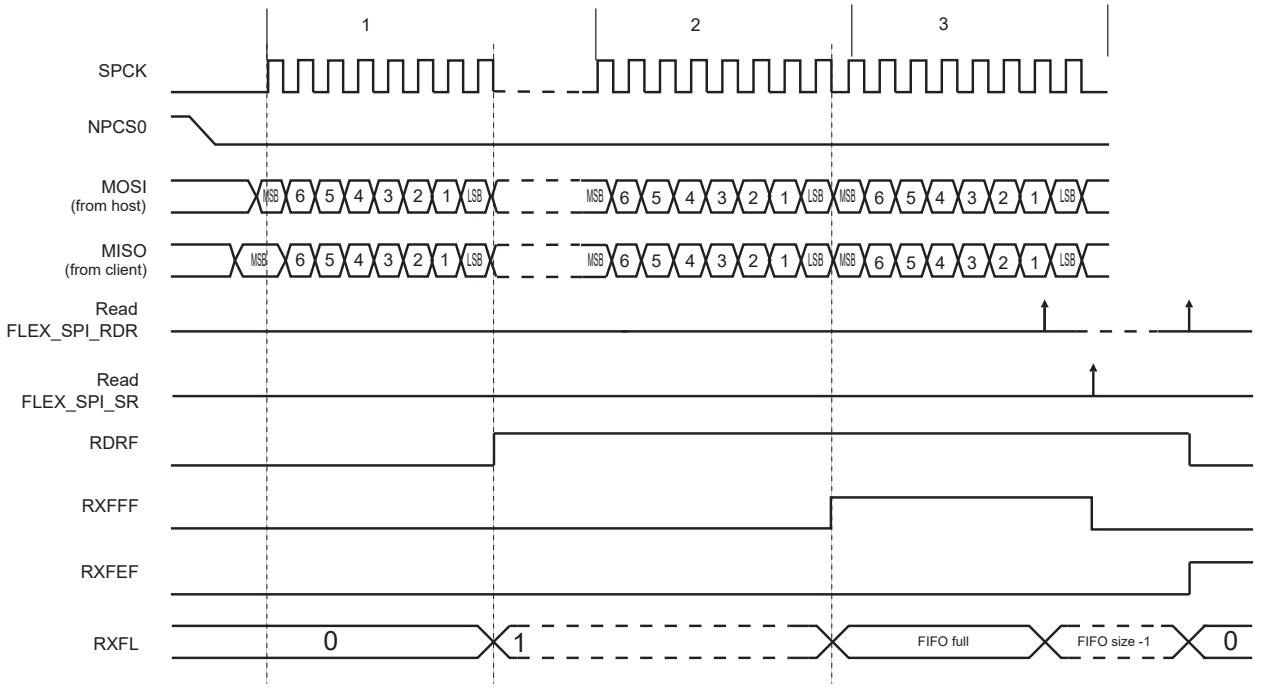


Figure 63.90. RDRF Behavior for Single Data Access and RXRDYM = 0



63.8.6.6.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.

63.8.6.6.1.DMA

When FIFOs operate in Single Data mode, the DMA 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.

63.8.6.7.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

63.8.6.7.1.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.

63.8.6.7.2.DMA

It is mandatory to configure DMA 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).

63.8.6.8.FIFO Overflow/Underflow Error

If the Transmit FIFO is full and a write access is performed on FLEX_SPI_TDR, it generates a Transmit FIFO overflow 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 overflow error is generated and FLEX_SPI_SR.TXFPTEF is set.

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 underflow error is generated and FLEX_SPI_SR.RXFPTEF is set.

No error occurs if the FIFO state/level is checked before writing/reading the required amount of data in FLEX_SPI_TDR/SPI_RDR. The FIFO state/level can be checked either with TXRDY, RXRDY, TXFL or RXFL. When such error occurs, other FIFO flags may not behave as expected; their states must be ignored. A software reset must be performed using FLEX_SPI_CR.SWRST (configuration will be lost).

63.8.6.9. 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.

63.8.6.10. 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.

63.8.7. SPI CRC Generation and Checking

The SPI offers the possibility to compute and check a CRC while receiving a frame. To enable the CRC check, the CRC Enable (CRCEN) bit must be set to '1' in FLEX_SPI_MR.

The SPI CRC register (FLEX_SPI_CRCR) configures the frame length using the Frame Length (FRL) field and the CRC size using the CRC Size (CRCS) bit.

The CRC checksum uses the 16-bit CRC-16 ANSI polynomial as defined in the IEEE 802.3 standard: x16 + x15 + x2 + 1.

It is possible to define a frame header length with the Frame Header Length (FRHL) field, and a frame header can be included or excluded from CRC calculation depending on the Frame Header Excluded (FHE) bit configuration. In case of continuous read frames, the Continuous Read Mode (CRM) bit defines if the frame header is sent only at the beginning of the first frame or at every frame. The CRC Error (CRCERR) flag in FLEX_SPI_SR indicates any failed CRC check.

It is possible to configure the SPI to receive or not CRC and/or Header as a classic data by configuring FLEX_SPI_CRCR.DCRX and FLEX_SPI_CRCR.DHRX. When CRC and/or Header is configured not to be received as data, upon receiving the Header and/or CRC, the RDRF flag does not rise and the FLEX_SPI_RDR register is not updated with the received value.

63.8.8. Two-Pin Mode

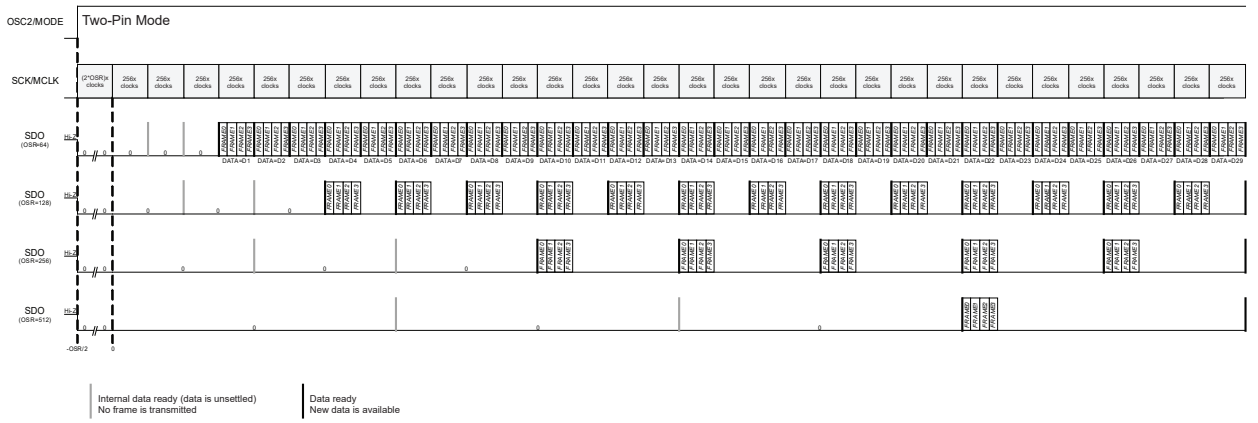
63.8.8.1. MCP3910 Protocol Overview

In Two-Pin mode, the SPI can easily interface to analog front-end ADC devices such as MCP3910. The MCP3910 device data output (SDO) is the host for the MOSI line, although the SPI generates the SPCK clock. FLEX_SPI_CSRx.SCBR defines the frequency of the SPCK output clock. The MISO line is not used.

Figure 63.91. Two-Pin MCP3910 Frame

SPI Two-Pin mode connects with up to four two-pin devices. Each MCP3910 product sends four times the same frame. See the following figure.

Figure 63.92. MCP3910 Communication Protocol



63.8.8.2.Two-Pin Mode Configuration

The Two-Pin mode is enabled when FLEX_SPI_MR.TPMEN=1 and FLEX_SPI_MR.MSTR=0.

When Two-Pin mode is enabled, SPI Mode 0 must be configured (FLEX_SPI_CSR0.CPOL=0 and FLEX_SPI_CSR0.NCPHA=0) and the data length must be 8 bits (FLEX_SPI_CSRx.BITS=0).

For proper CRC calculation and checking, FLEX_SPI_CRCR.FHE must be set to '0', FLEX_SPI_CRCR.CRM to '0', FLEX_SPI_CRCR.FRHL to '1' and FLEX_SPI_CRCR.CRCS to '0'. MCSPI_CRCR.FRL must be set according to the value used for FLEX_SPI_TPMR.OSR (refer to “Two-Wire Serial Interface Description” in the data sheet “MCP3910” available on www.microchip.com for more details on frame length and OSR).

See SPI CRC Generation and Checking for details on CRC check configuration.

To get the synchronization on SYNC BYTE, the Comparison mode must be enabled (FLEX_SPI_MR.CMPMODE=1) and the comparison value must be configured to SYNC BYTE (FLEX_SPI_CMPR.VAL1/2= 0xA5). The clock is generated on SPCK when FLEX_SPI_CR.SPIEN is set.

Once the SPI is configured in Two-Pin mode and Comparison mode is set, the synchronization byte (SYNC BYTE) is monitored to start receiving a frame. Upon SYNC BYTE reception, the previously received header byte is stored in the Two-Pin Header register (FLEX_SPI_TPHR) and the SYNC BYTE is stored in FLEX_SPI_RDR (the RDRF flag indicates when the data is available). Each frame byte received is written in FLEX_SPI_RDR until reception of the CRCCOM end field (CRC is received as a data).

The Oversampling Rate (OSR) field in the Two-Pin Mode register (FLEX_SPI_TPMR) defines the OSR frame configuration of the MCP3910, and thus the frame length.

Note: It is mandatory to configure FLEX_SPI_CRCR when Two-Pin mode is enabled.

63.8.8.3. Connection Example

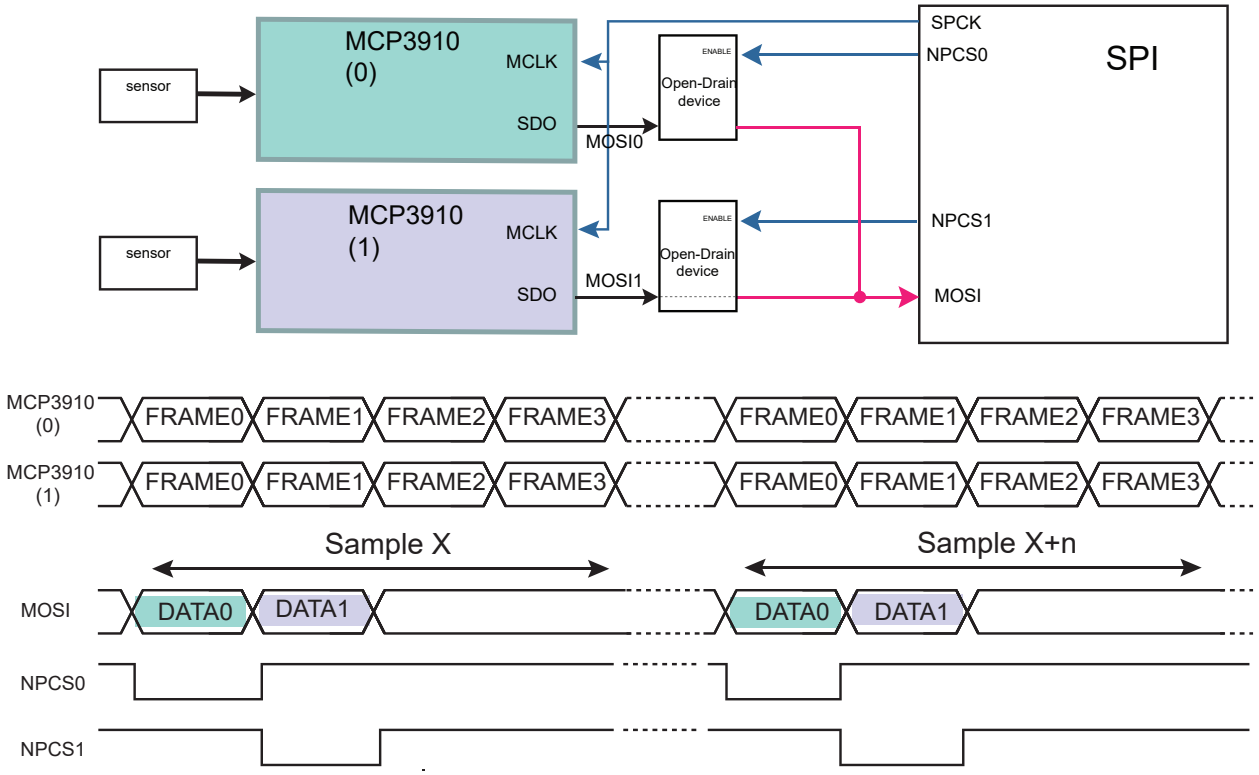
When more than 1 MCP3910 device is connected to SPI, the SDO data lines from the MCP3910 products must be multiplexed prior to driving the SPI MOSI line.

MOSI multiplexing must be performed by successively enabling each SDO line from the MCP3910 with external discrete components. See the following figure.

In this mode, the NPCS lines are activated successively after each frame reception.

The polarity of chip select output NPCS is defined by configuring the Chip Select Inversion Enable (CSIE) bit in FLEX_SPI_MR.

Figure 63.93. MCP3910 Connection Example with Two Devices



63.8.9. 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
- SPI CRC Register
- SPI Two-Pin Mode 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

63.8.10. Local Loopback Test Mode

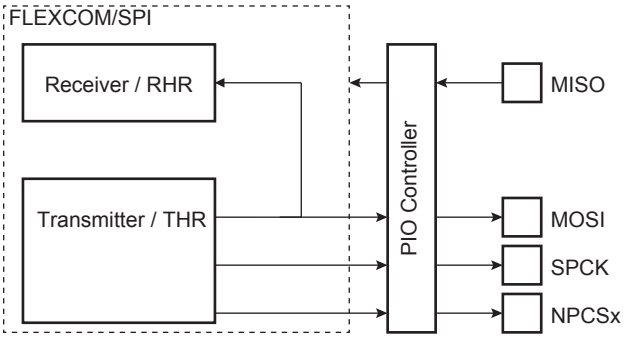
Local Loopback mode connects the output of the transmitter directly to the input of the receiver, as shown in the figure below. The MISO pin has no effect on the receiver and the MOSI pin is driven as in Normal mode.

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 63.94. Local Loopback Mode Configuration

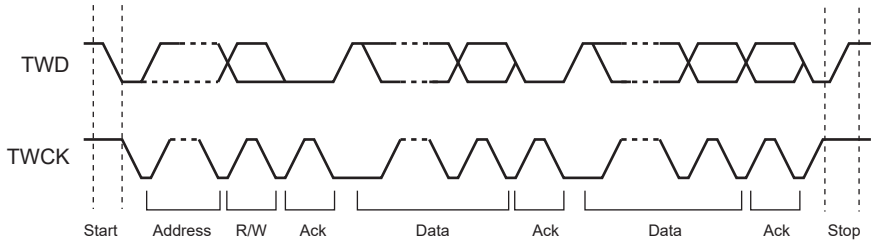


63.9. TWI Functional Description

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).

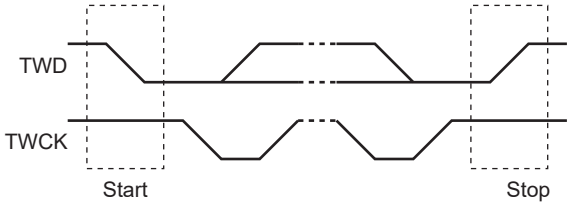
Figure 63.95. Transfer Format



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 63.96. 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 63.97. Host Write with One Data Byte

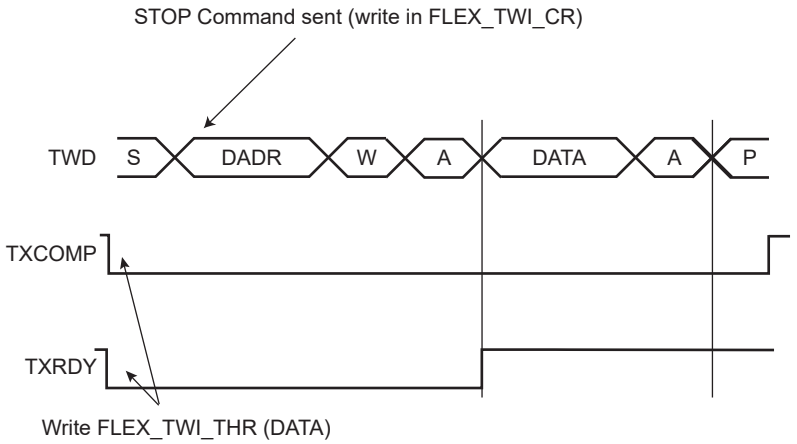


Figure 63.98. Host Write with Multiple Data Bytes

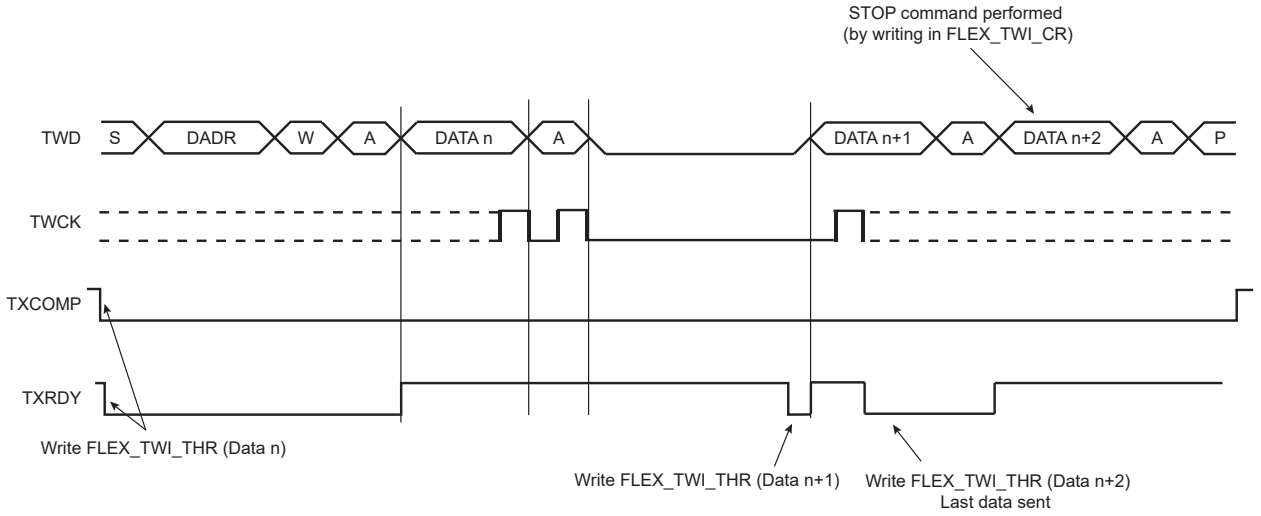
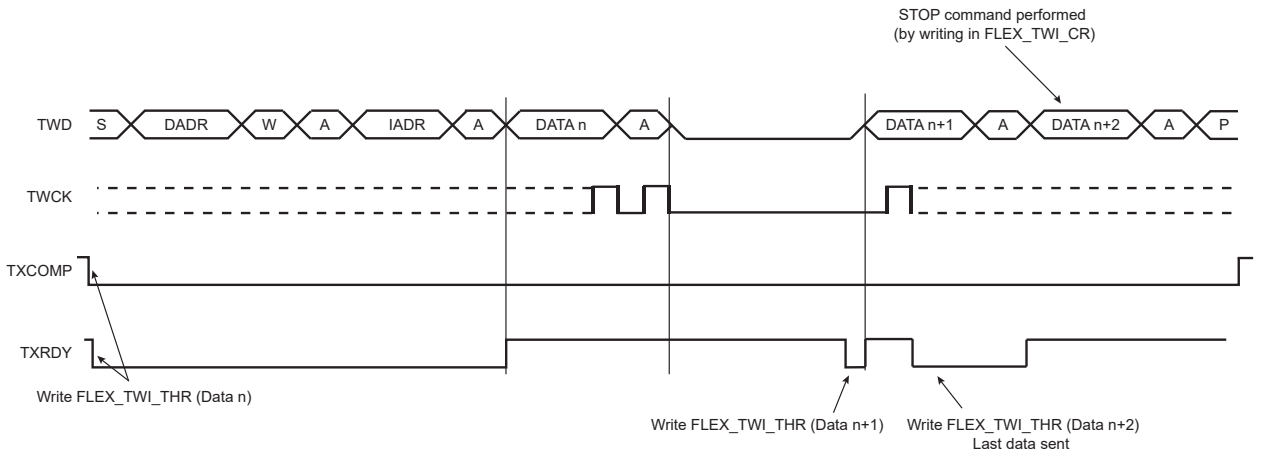


Figure 63.99. Host Write with One Byte Internal Address and Multiple Data Bytes



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 63.100. Host Read with One Data Byte

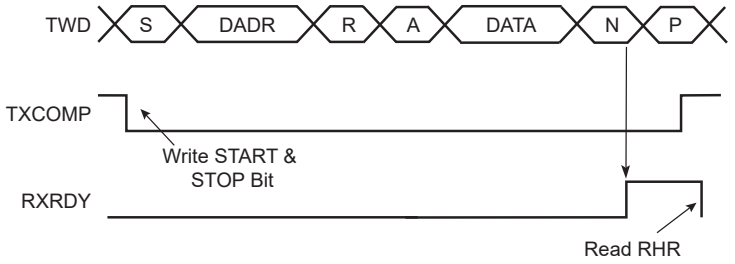


Figure 63.101. Host Read with Multiple Data Bytes

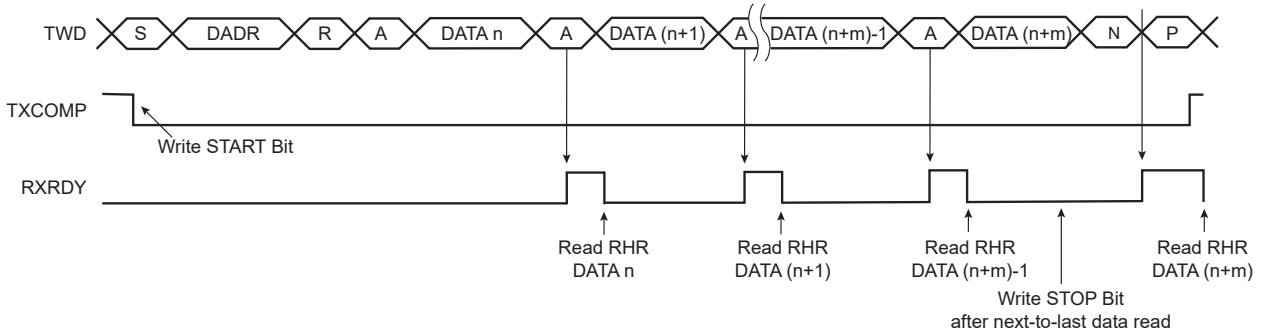
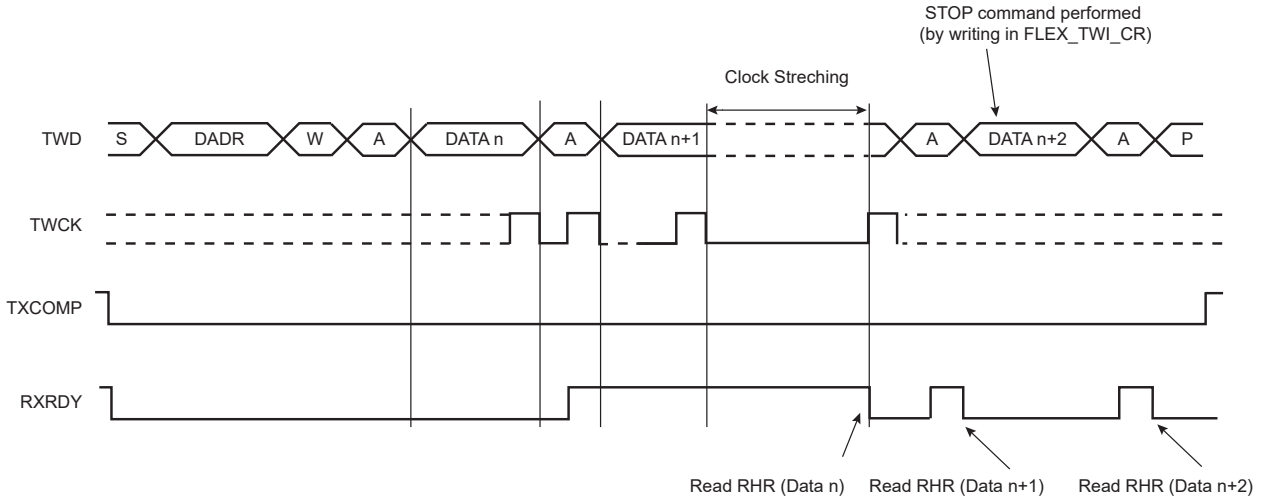


Figure 63.102. Host Read Clock Stretching with Multiple Data Bytes



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, for example, 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 63.103. Host Write with One, Two or Three Bytes Internal Address and One Data Byte

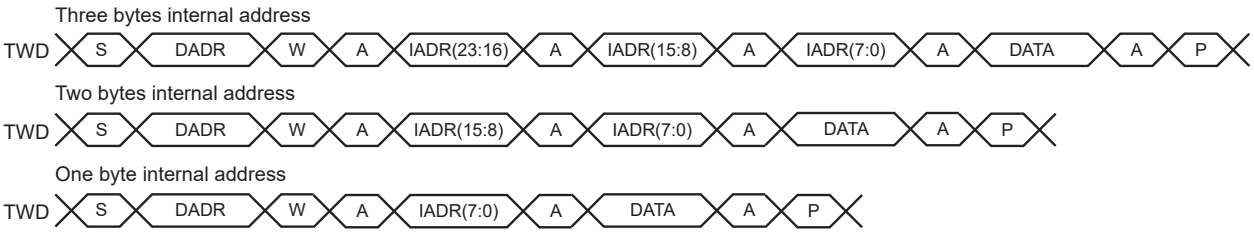
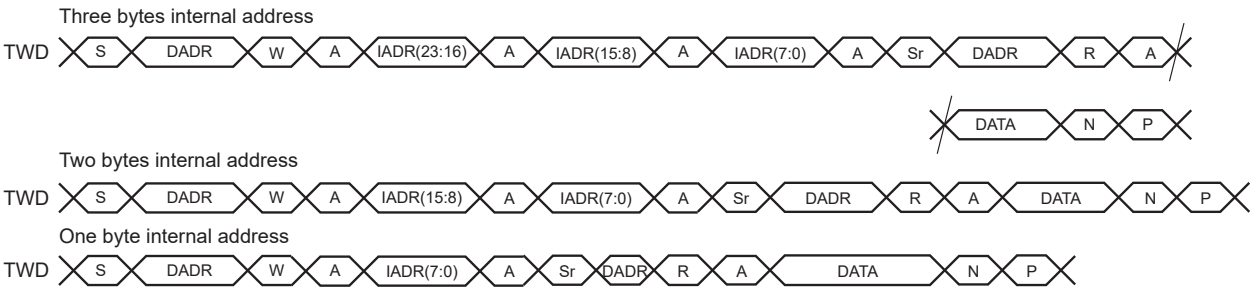


Figure 63.104. Host Read with One, Two or Three Bytes Internal Address and One Data Byte



63.9.3.6.2.10-bit Client Addressing

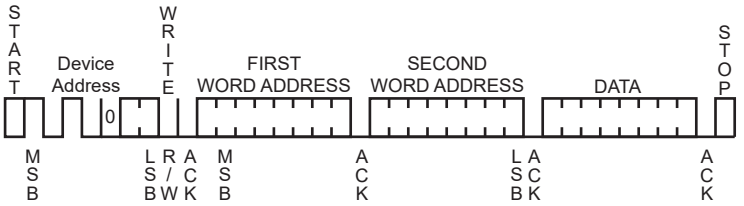
For a client address higher than seven bits, the user must configure the address size (IADRSZ) and set the other client address bits in the Internal Address register (FLEX_TWI_IADR). The two remaining internal address bytes, IADR[15:8] and IADR[23:16], can be used the same way as in 7-bit client addressing.

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 63.105. 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. Read FLEX_TWI_SR.SDA/SCL flags and check that the TWD (SDA) and TWCK (SCL) lines hold a high level.
3. Send the Bus Clear command by setting FLEX_TWI_CR.CLEAR.

Note: When TWD (SDA)=0, the Bus Clear command must be performed via the PIO. When TWCK (SCL)=0, no Bus Clear command can be issued.

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.

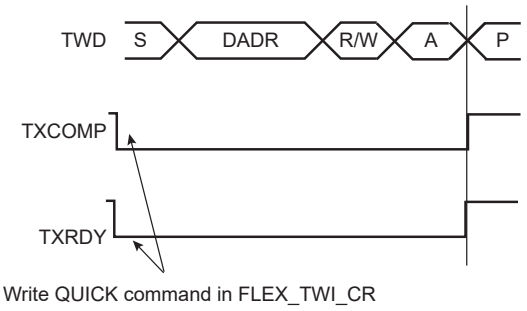
63.9.3.10.SMBus Quick Command (Host Mode Only)

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 63.106. 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)

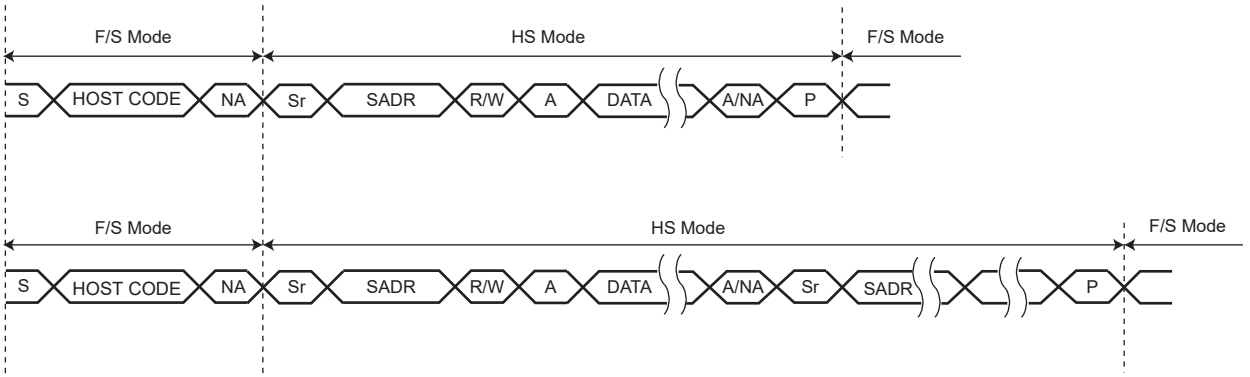
Note that to enable high-speed transfer, the client must send a NACK after receiving the host code.

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 63.107. 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.

The SCL Rise Boost feature can be enabled with FLEX_TWI_CR.SCLRBE and disabled with FLEX_TWI_CR.SCLRBD.

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 FLEX_TWI_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 63.108. TWI Write Operation with Single Data Byte without Internal Address

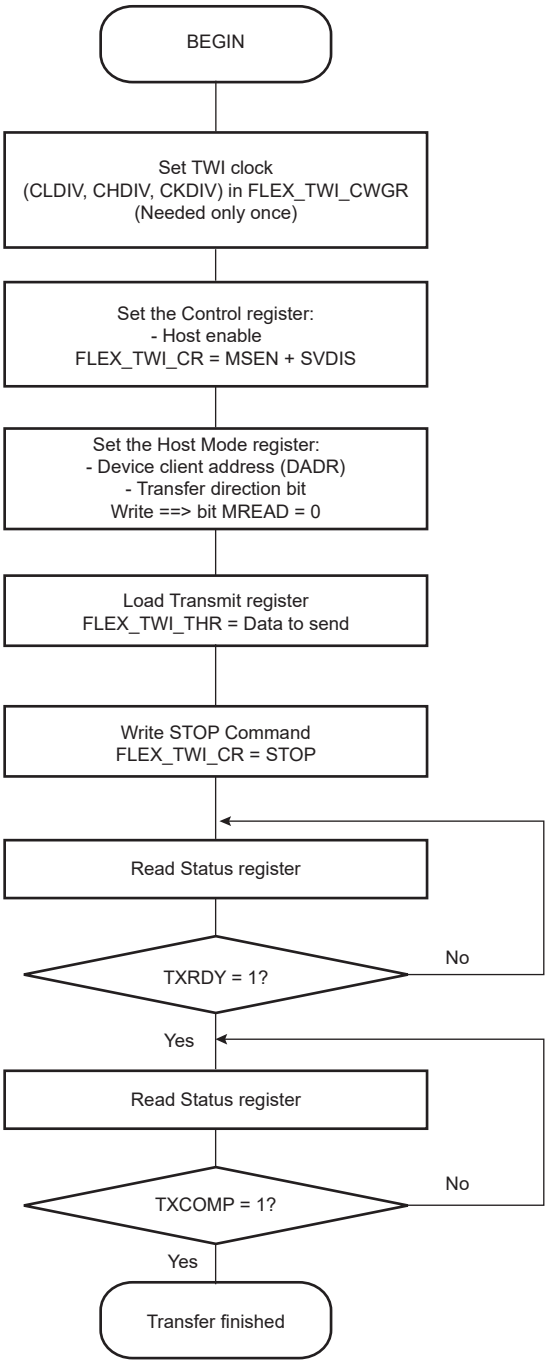


Figure 63.109. TWI Write Operation with Single Data Byte and Internal Address

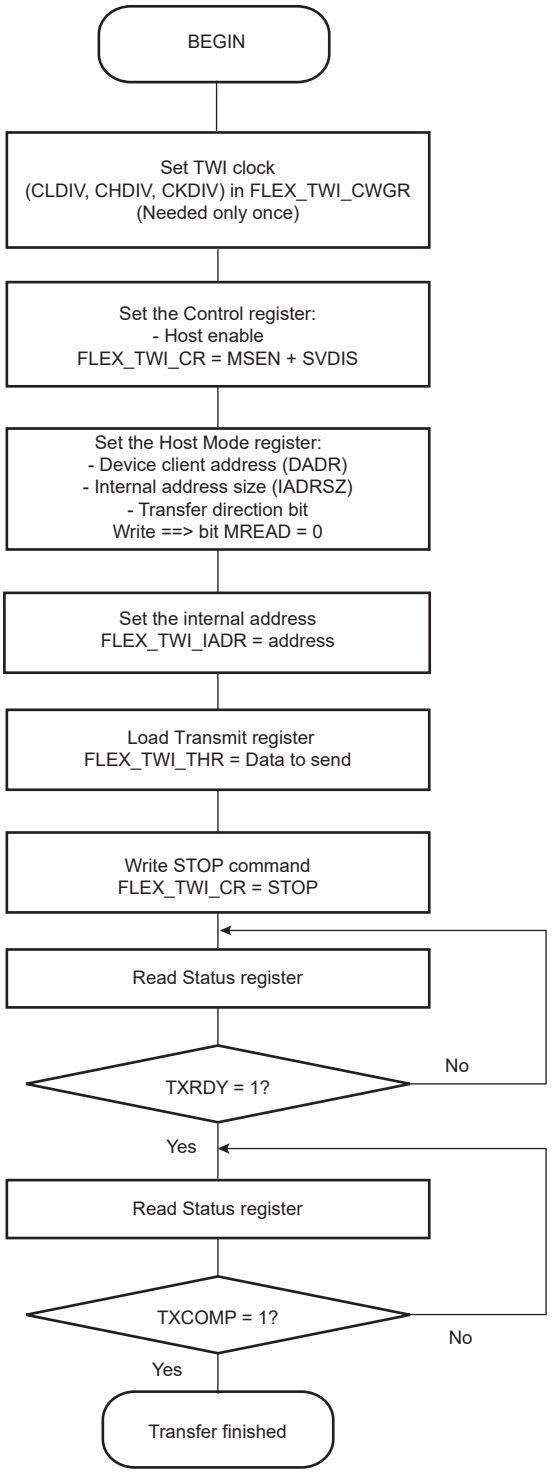


Figure 63.110. TWI Write Operation with Multiple Data Bytes with or without Internal Address

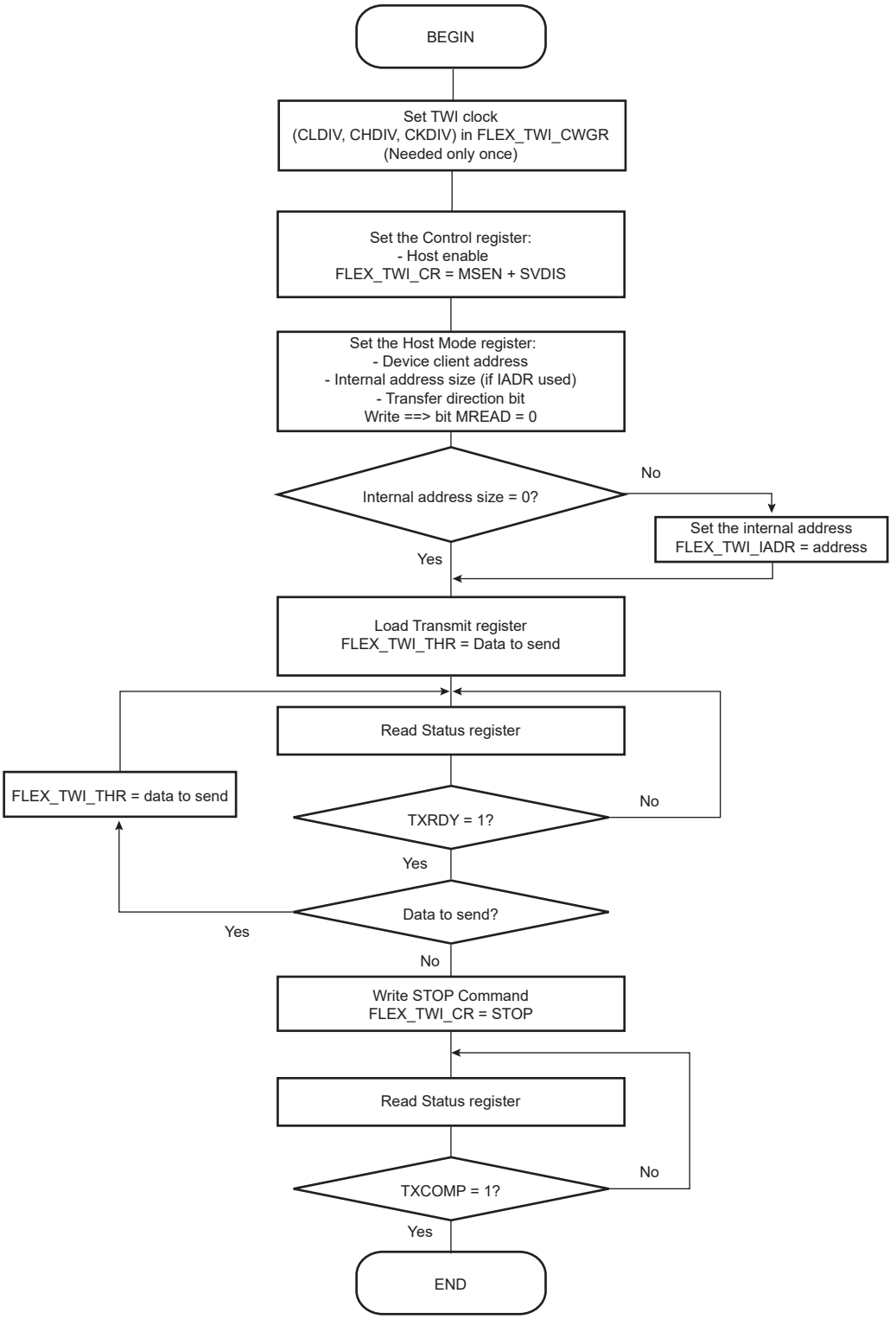


Figure 63.111. SMBus Write Operation with Multiple Data Bytes with or without Internal Address and PEC Sending

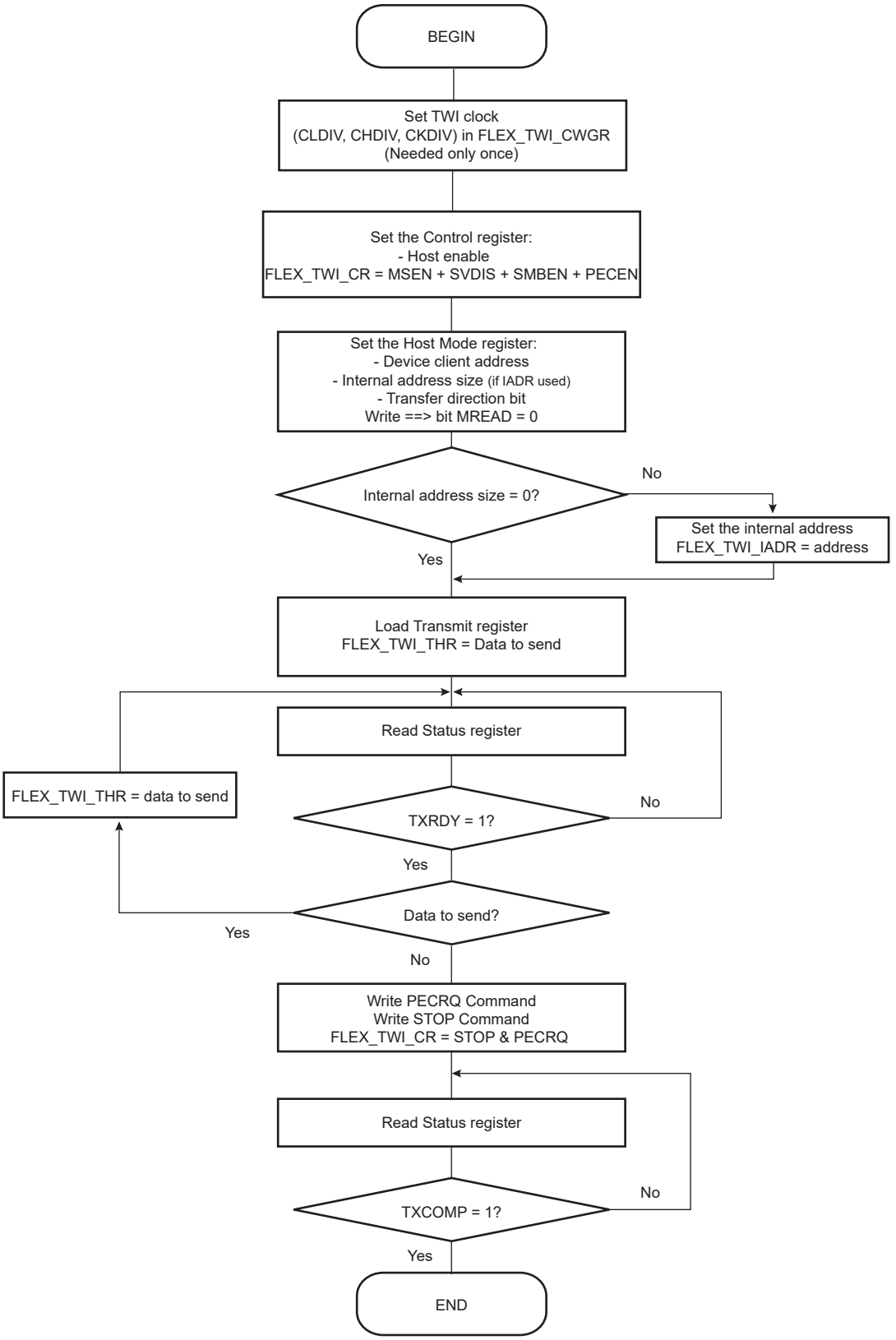


Figure 63.112. SMBus Write Operation with Multiple Data Bytes with PEC and Alternative Command Mode

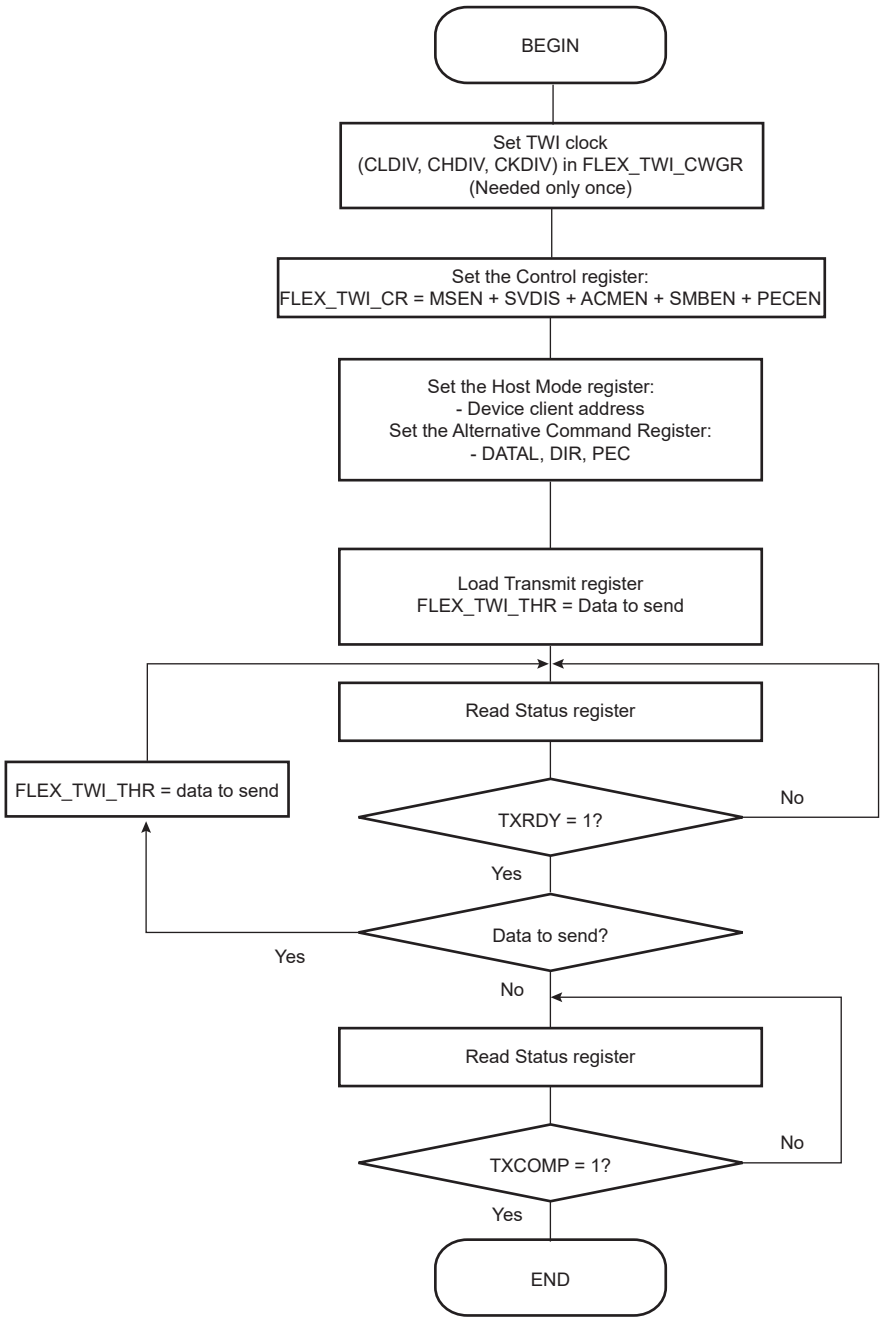


Figure 63.113. TWI Write Operation with Multiple Data Bytes and Read Operation with Multiple Data Bytes (Sr)

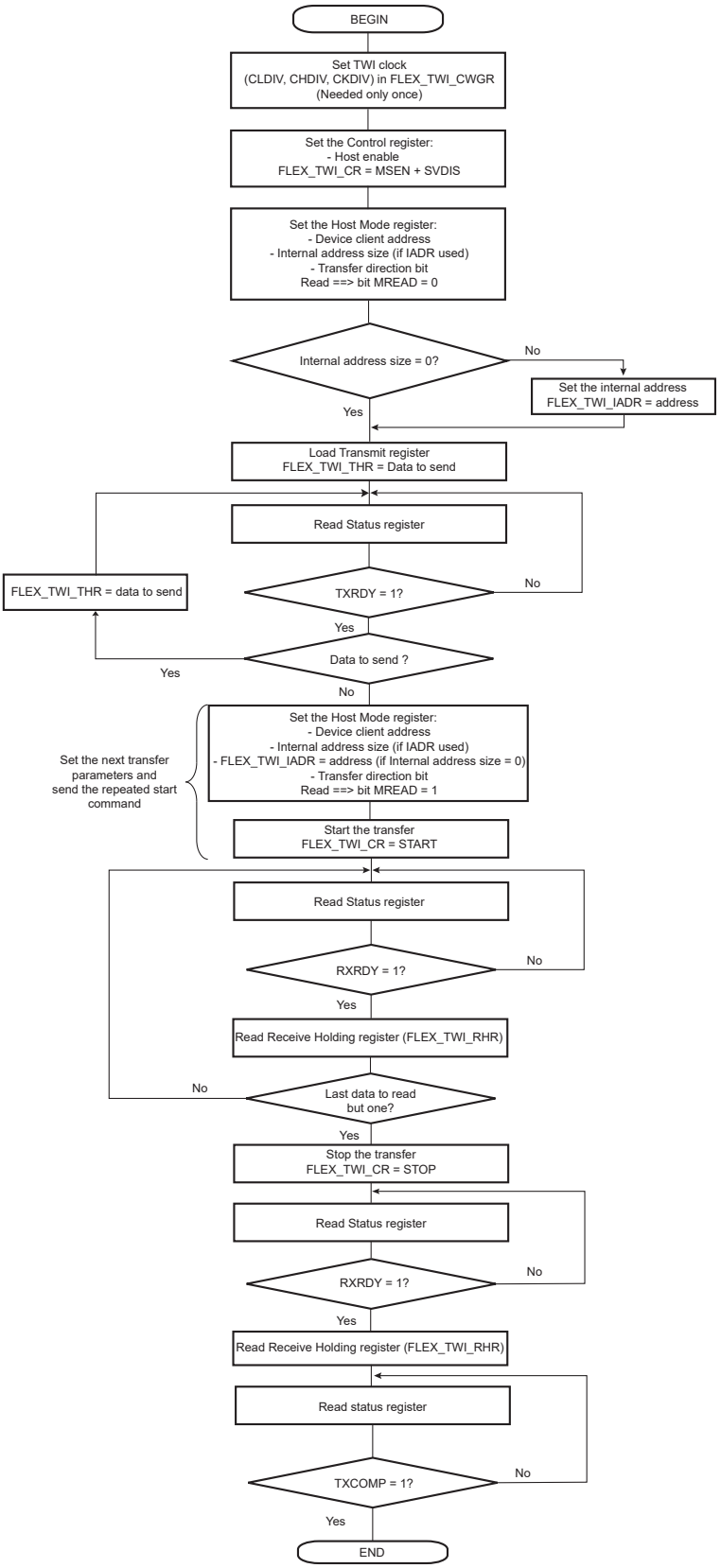


Figure 63.114. TWI Write Operation with Multiple Data Bytes + Read Operation and Alternative Command Mode + PEC

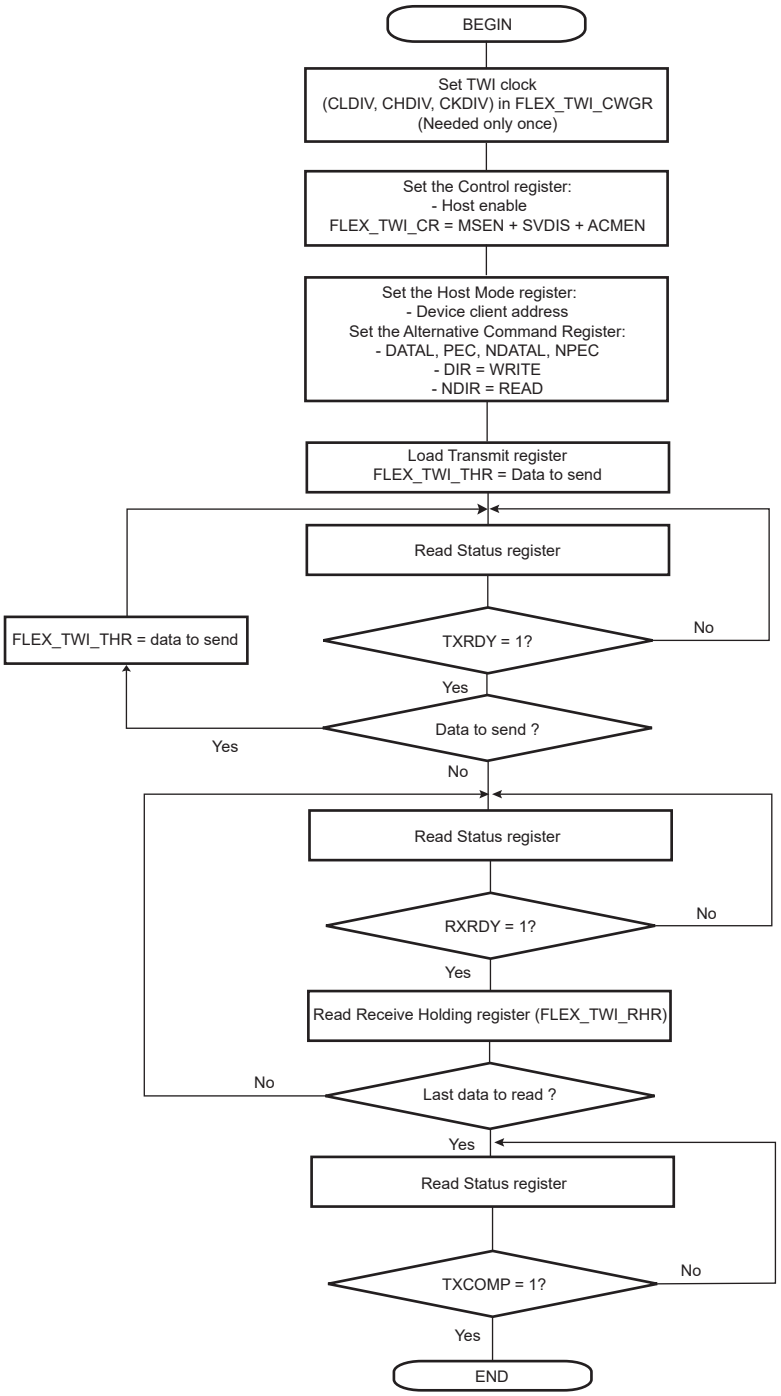


Figure 63.115. TWI Read Operation with Single Data Byte without Internal Address

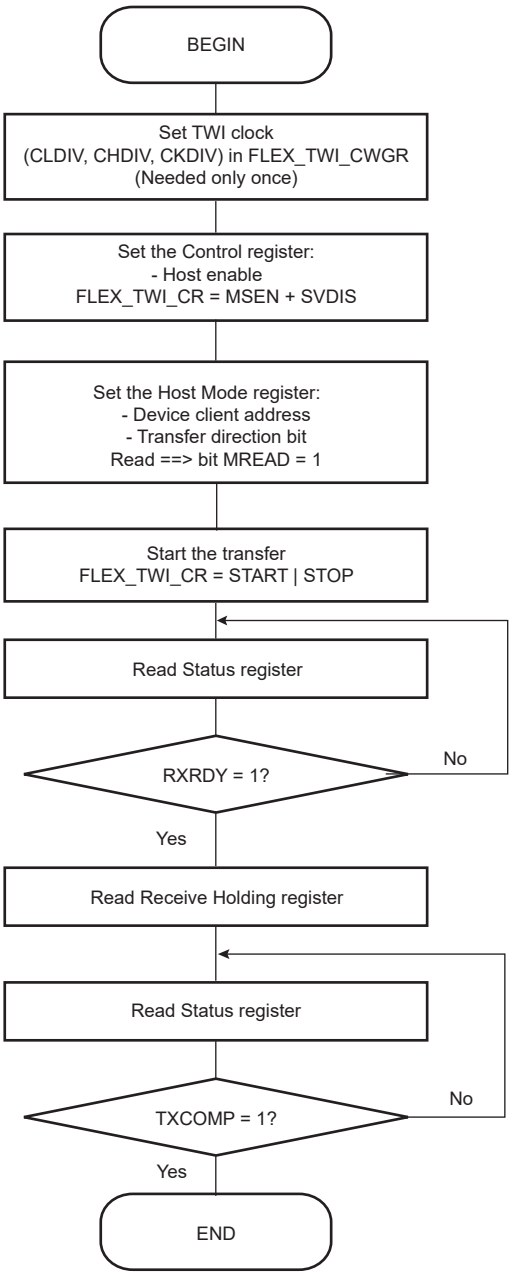


Figure 63.116. TWI Read Operation with Single Data Byte and Internal Address

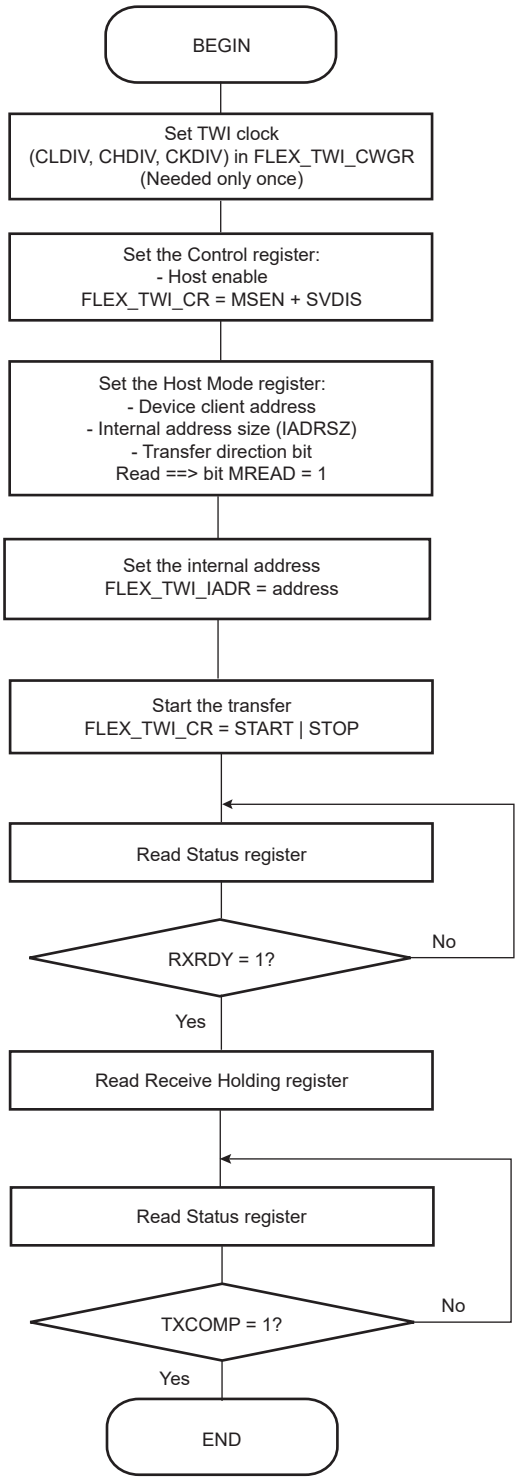


Figure 63.117. TWI Read Operation with Multiple Data Bytes with or without Internal Address

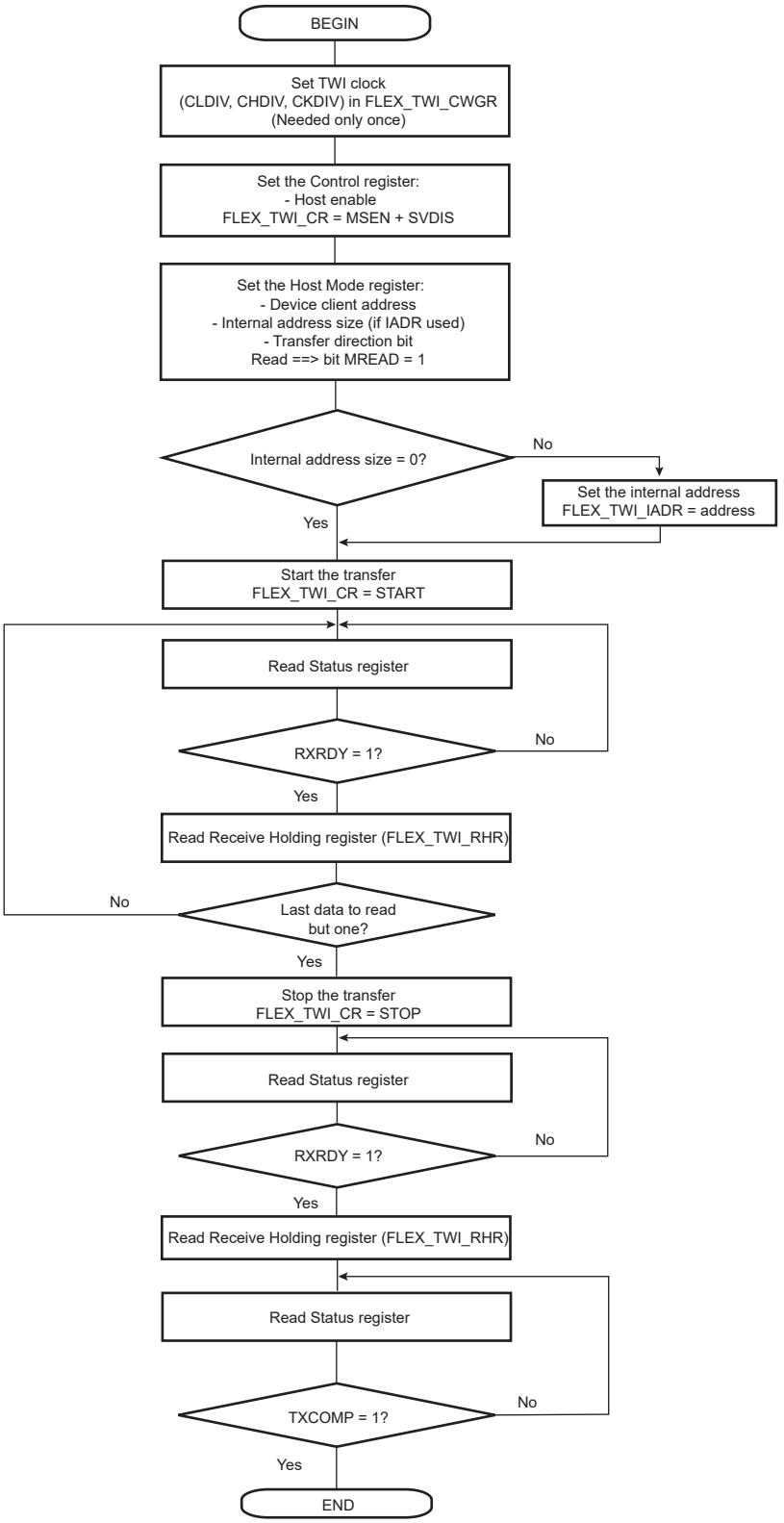


Figure 63.118. TWI Read Operation with Multiple Data Bytes with or without Internal Address with PEC

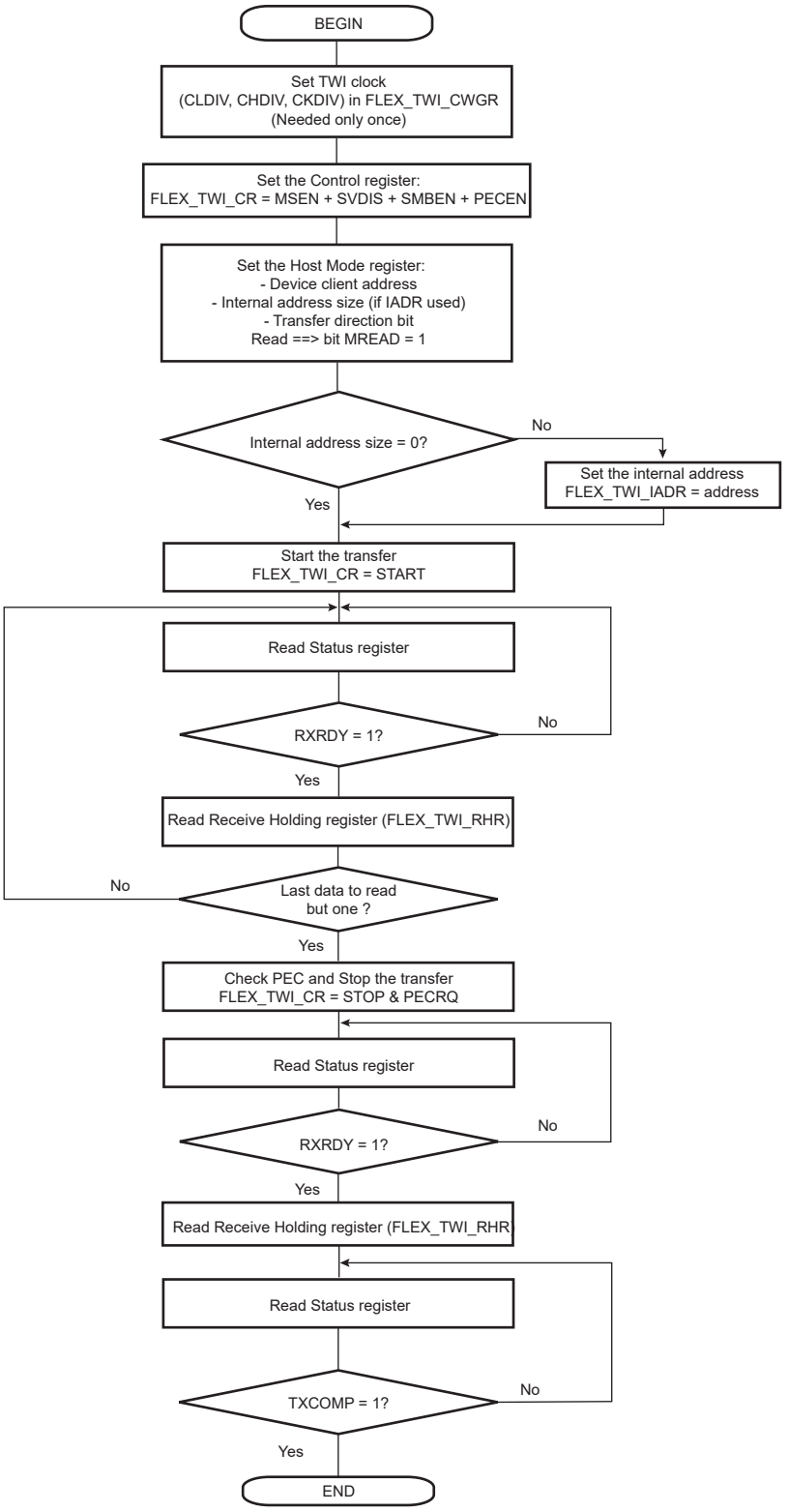


Figure 63.119. TWI Read Operation with Multiple Data Bytes with Alternative Command Mode with PEC

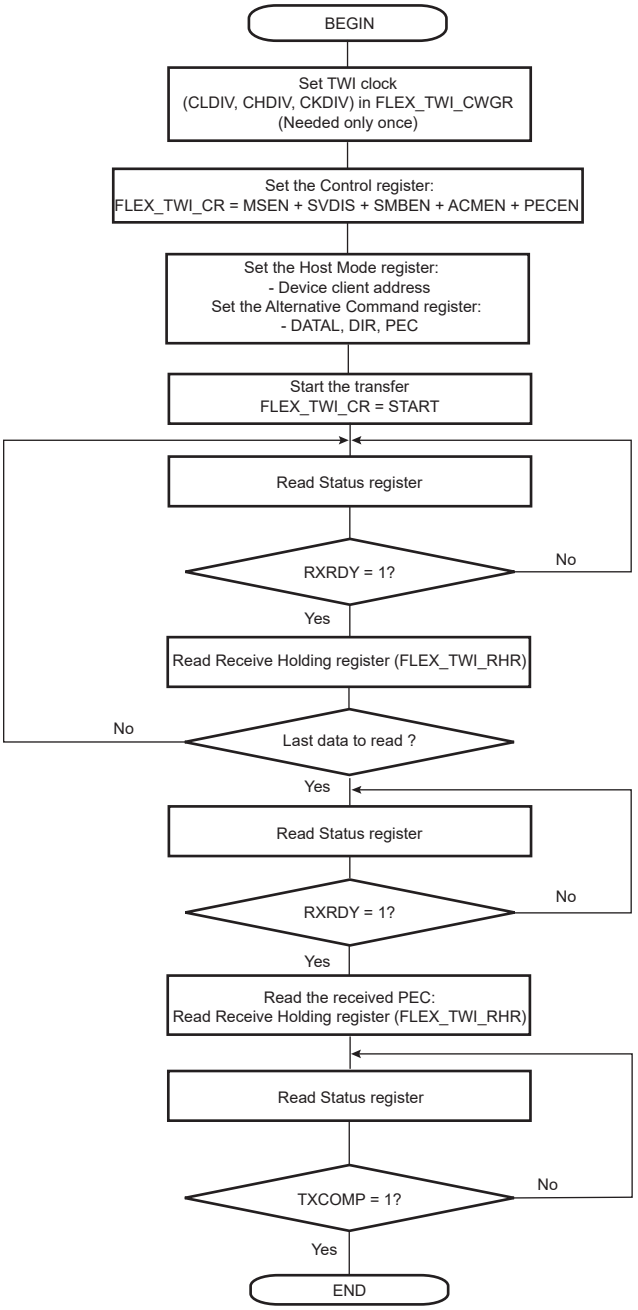


Figure 63.120. TWI Read Operation with Multiple Data Bytes + Write Operation with Multiple Data Bytes (Sr)

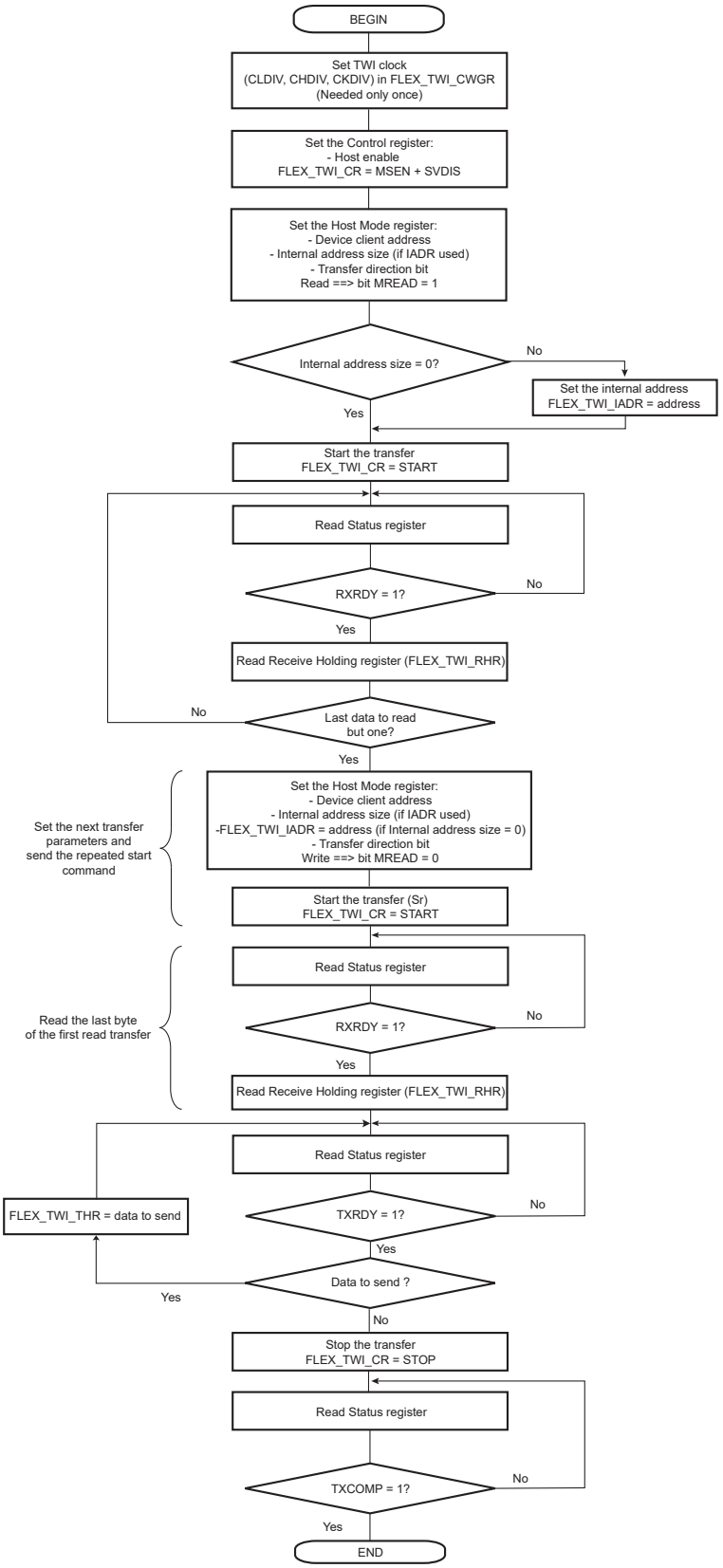
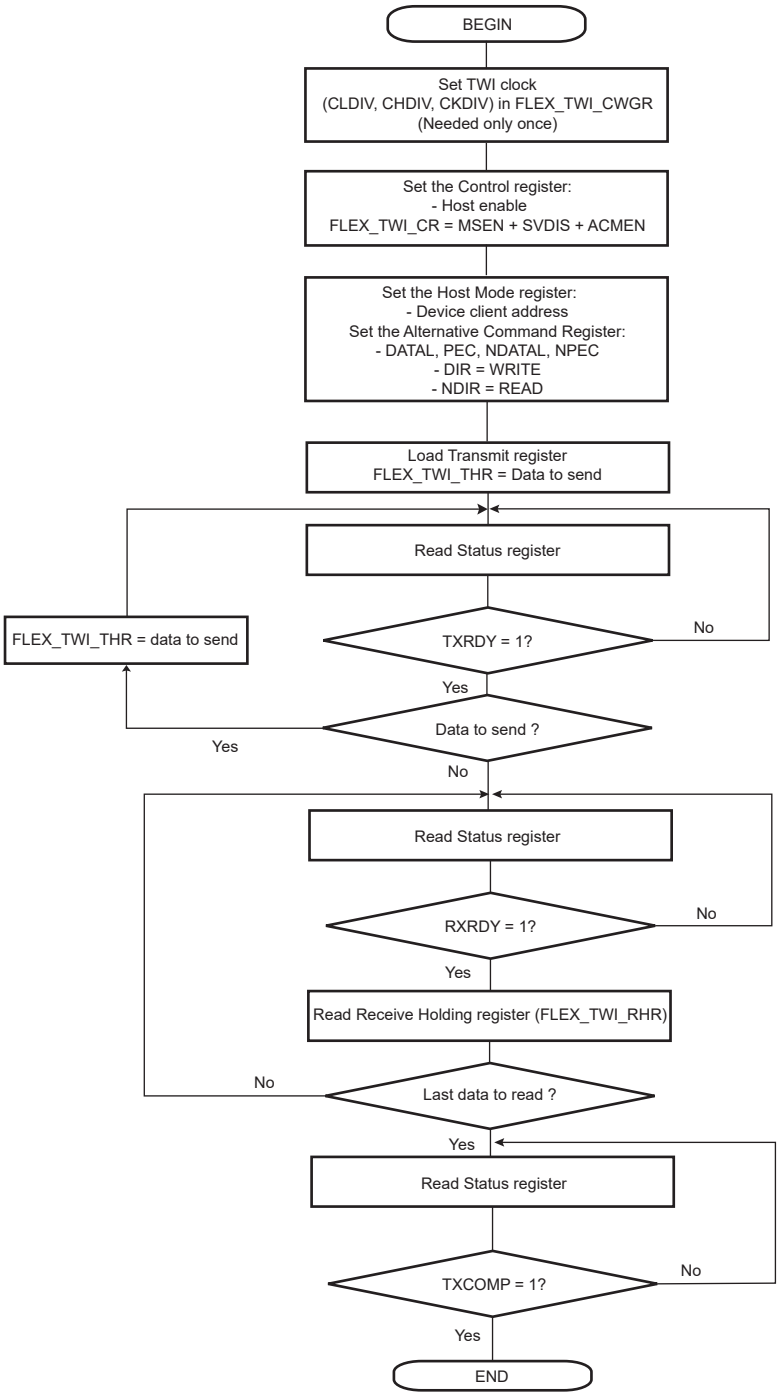


Figure 63.121. TWI Read Operation with Multiple Data Bytes + Write with Alternative Command Mode with PEC



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 63.122. User Sends Data While the Bus is Busy

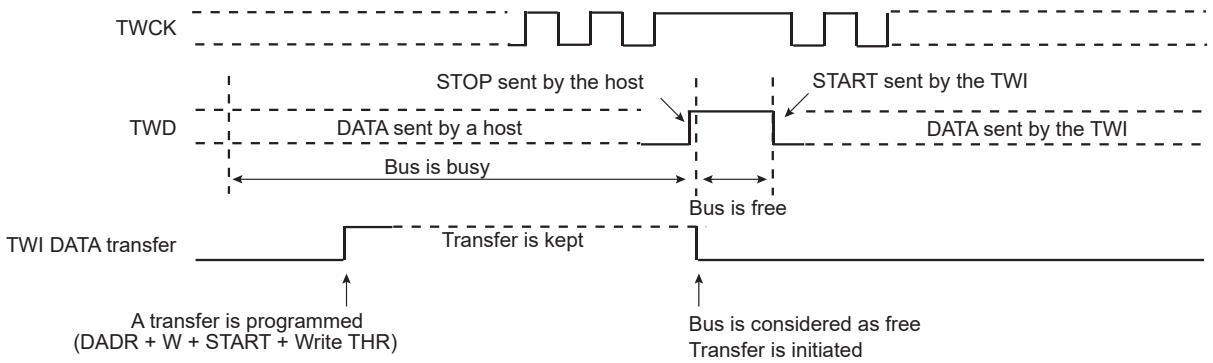
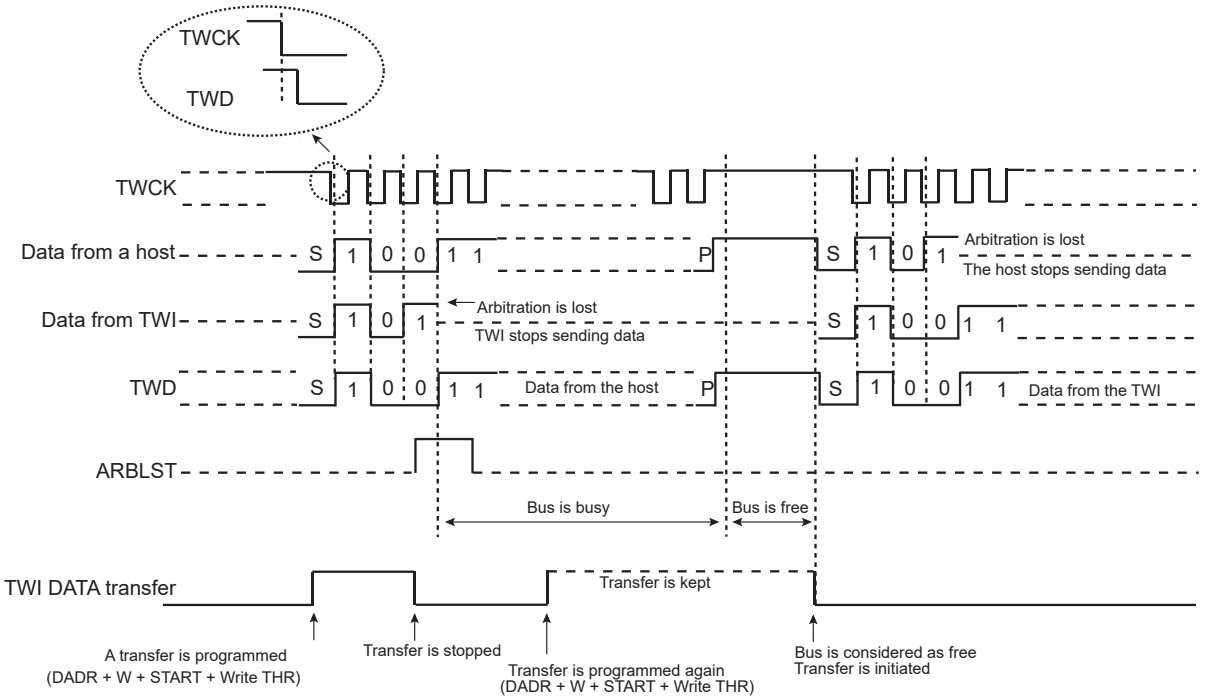
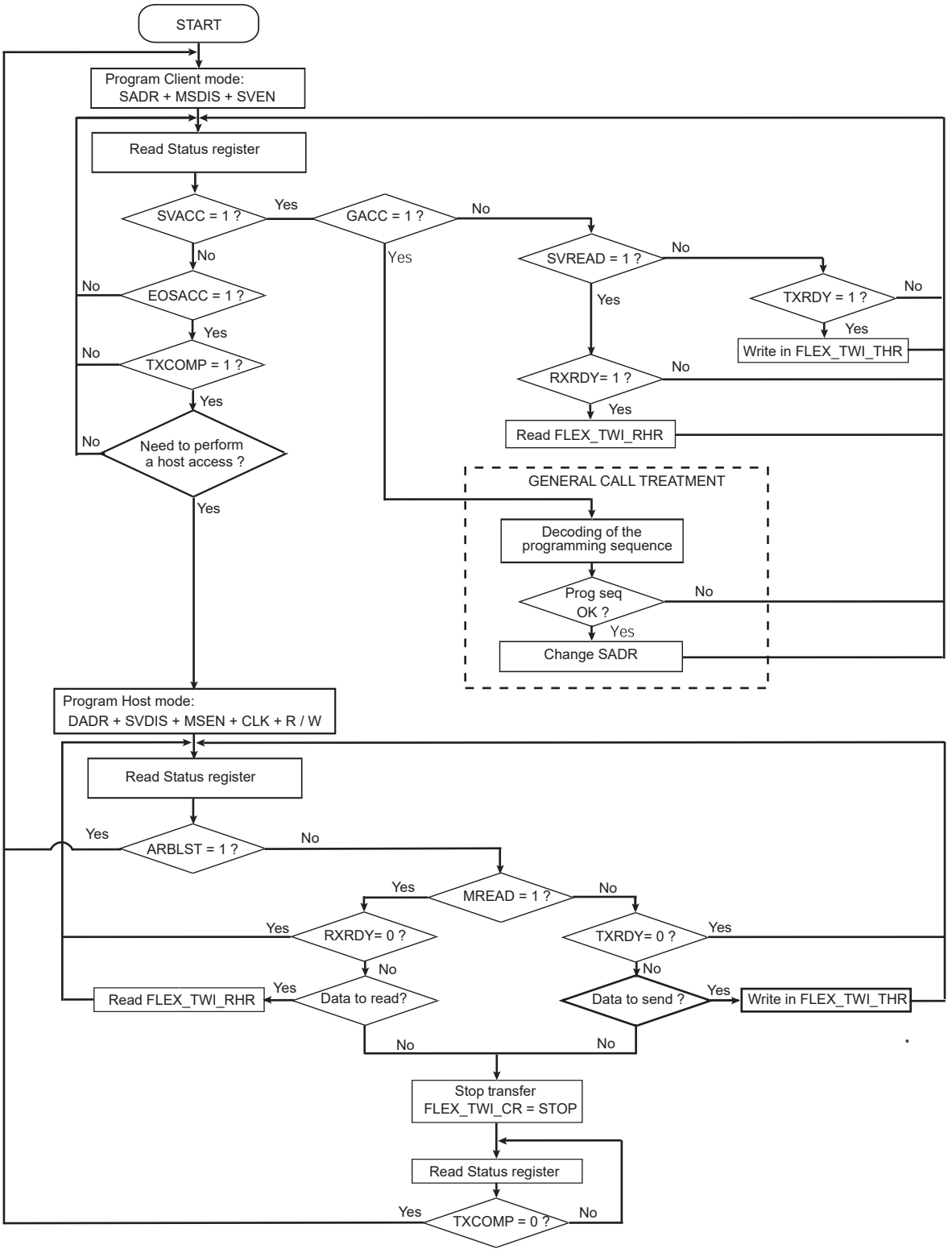


Figure 63.123. Arbitration Cases



The flowchart shown in the following figure gives an example of read and write operations in Multi-Host mode.

Figure 63.124. 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.

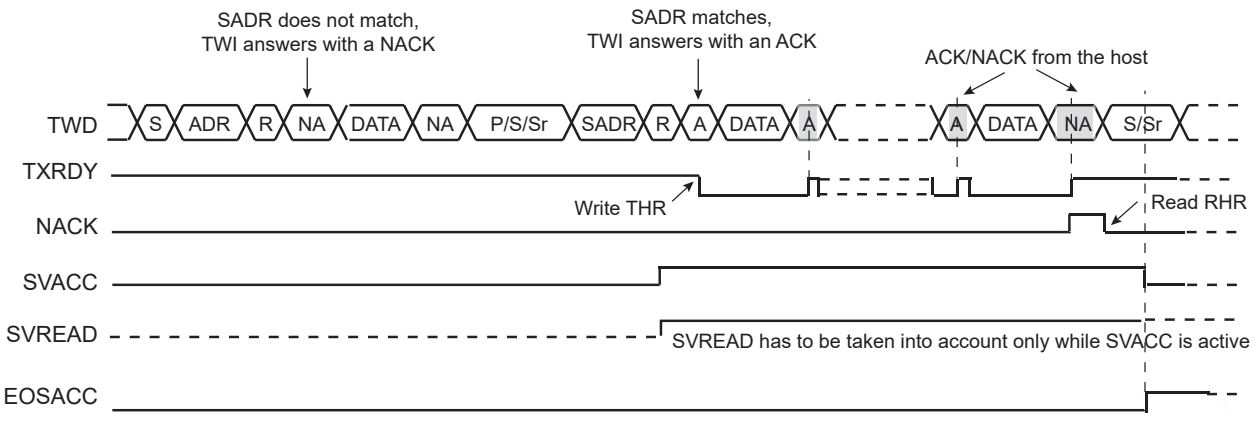
After a START or a REPEATED START condition is detected, the decoding of the address starts. If the client address (SADR) is decoded, SVACC is set and SVREAD indicates the direction of the transfer.

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 63.125. 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.

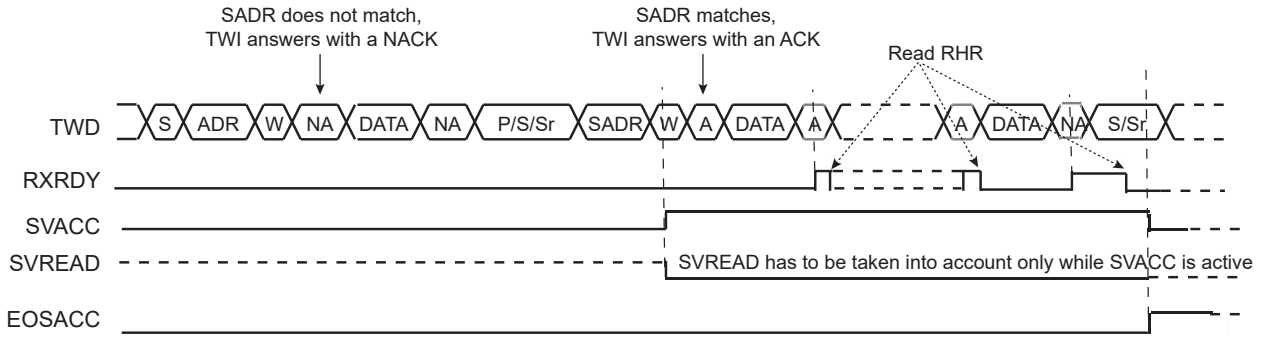
After a START or a REPEATED START, the decoding of the address starts. If the client address is decoded, SVACC is set and SVREAD indicates the direction of the transfer (SVREAD is low in this case).

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 63.126. 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.

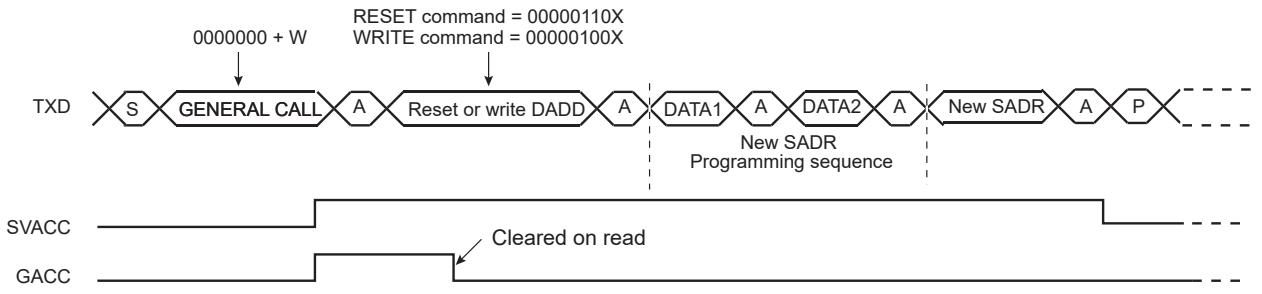
If a GENERAL CALL is detected, GACC is set.

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 63.127. 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.

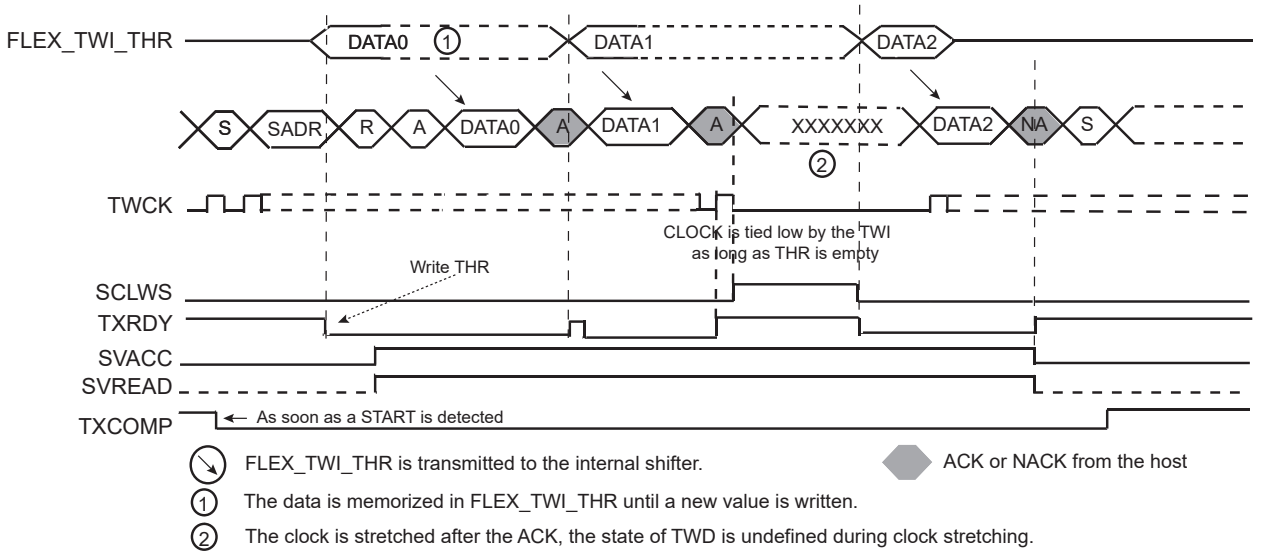
Note: Clock stretching can be disabled by setting the FLEX_TWI_SMR.SCLWSDIS bit. In that case, the UNRE and OVRE flags indicate an underrun (when FLEX_TWI_THR is not filled on time) or an overrun (when FLEX_TWI_RHR is not read on time).

— 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 63.128. Clock Stretching in Read Mode



Notes:

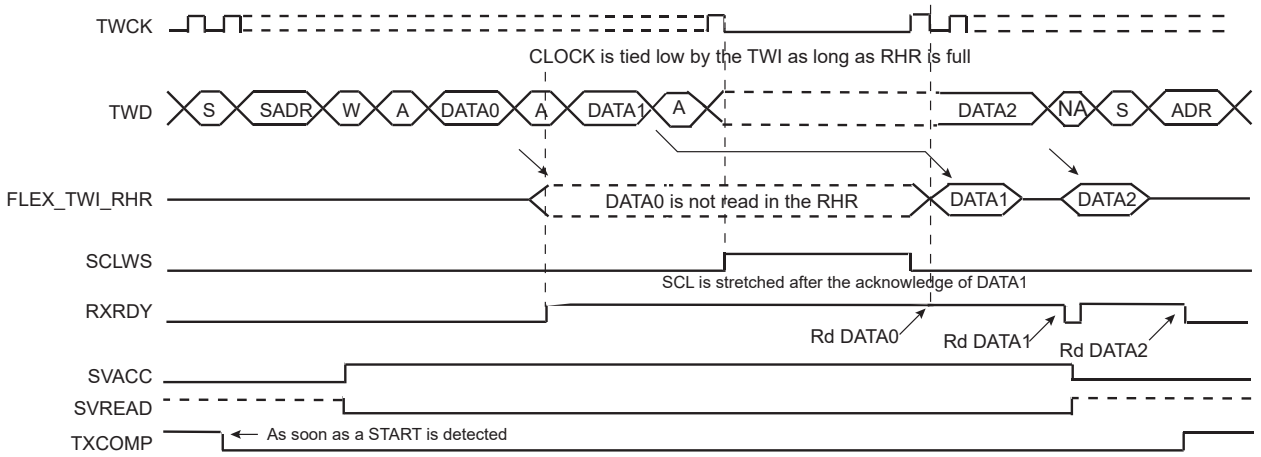
1. TXRDY is reset when data has been written in FLEX_TWI_THR to the internal shifter, and set when this data has been acknowledged or non acknowledged.
2. At the end of the read sequence, TXCOMP is set after a STOP or after a REPEATED_START + an address different from SADR.
3. SCLWS is automatically 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 63.129. 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.

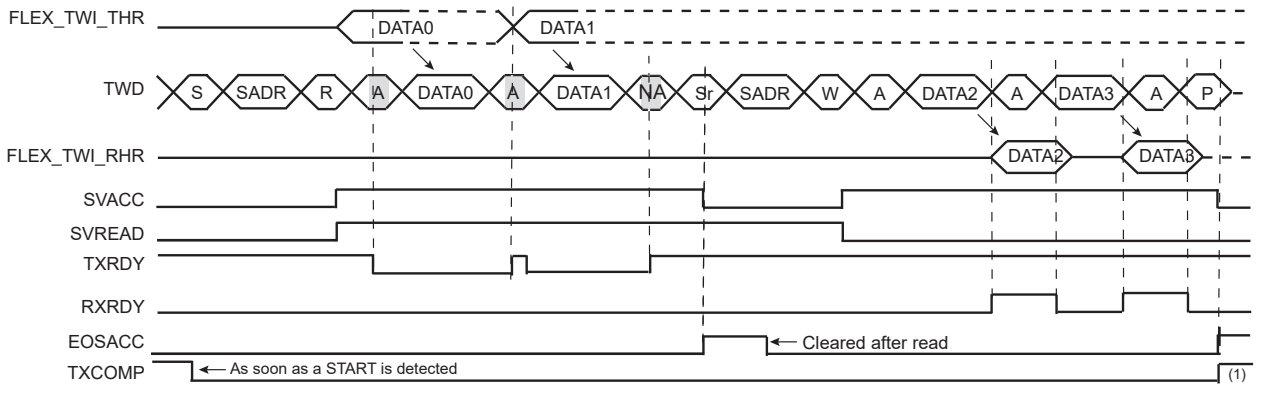
63.9.5.4.5.Reversal after a Repeated Start

— 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 63.130. Repeated Start and Reversal from Read Mode to Write Mode



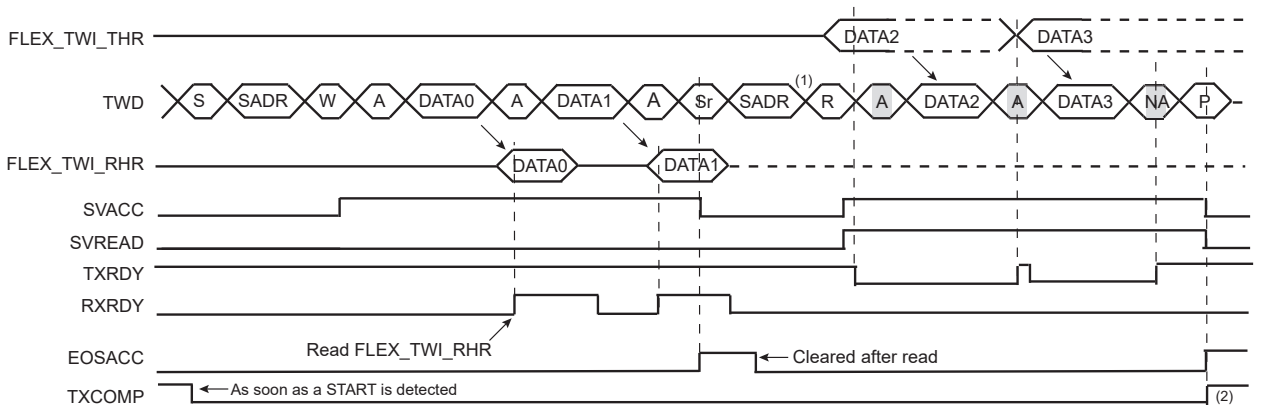
Note:
1. TXCOMP is only set at the end of the transmission because after the repeated start, SADR is detected again.

— 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 63.131. 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.

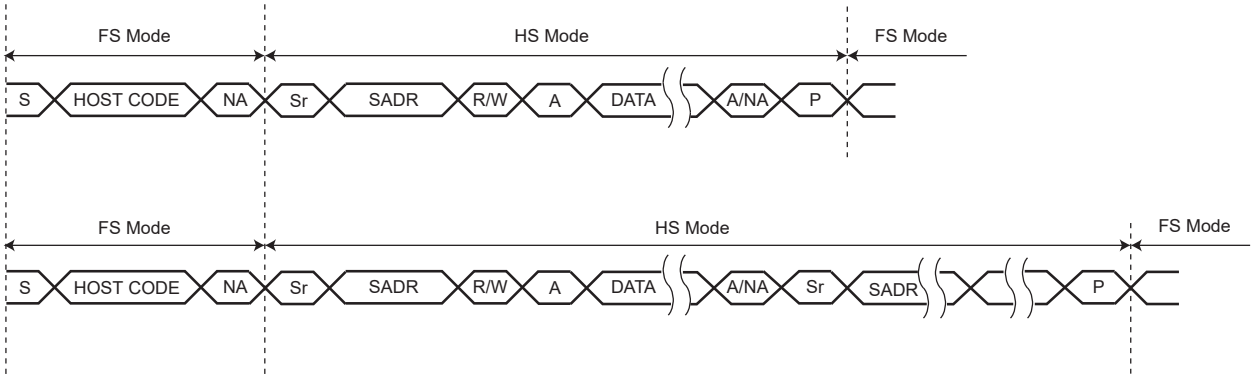
63.9.5.5.1.Read/Write Operation

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 63.132. 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 TWI Control register, and the transfer is configured in FLEX_TWI_ACR.

For a combined transfer with PEC, only the NPEC bit in FLEX_TWI_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.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 63.133. Read/Write in Client Mode

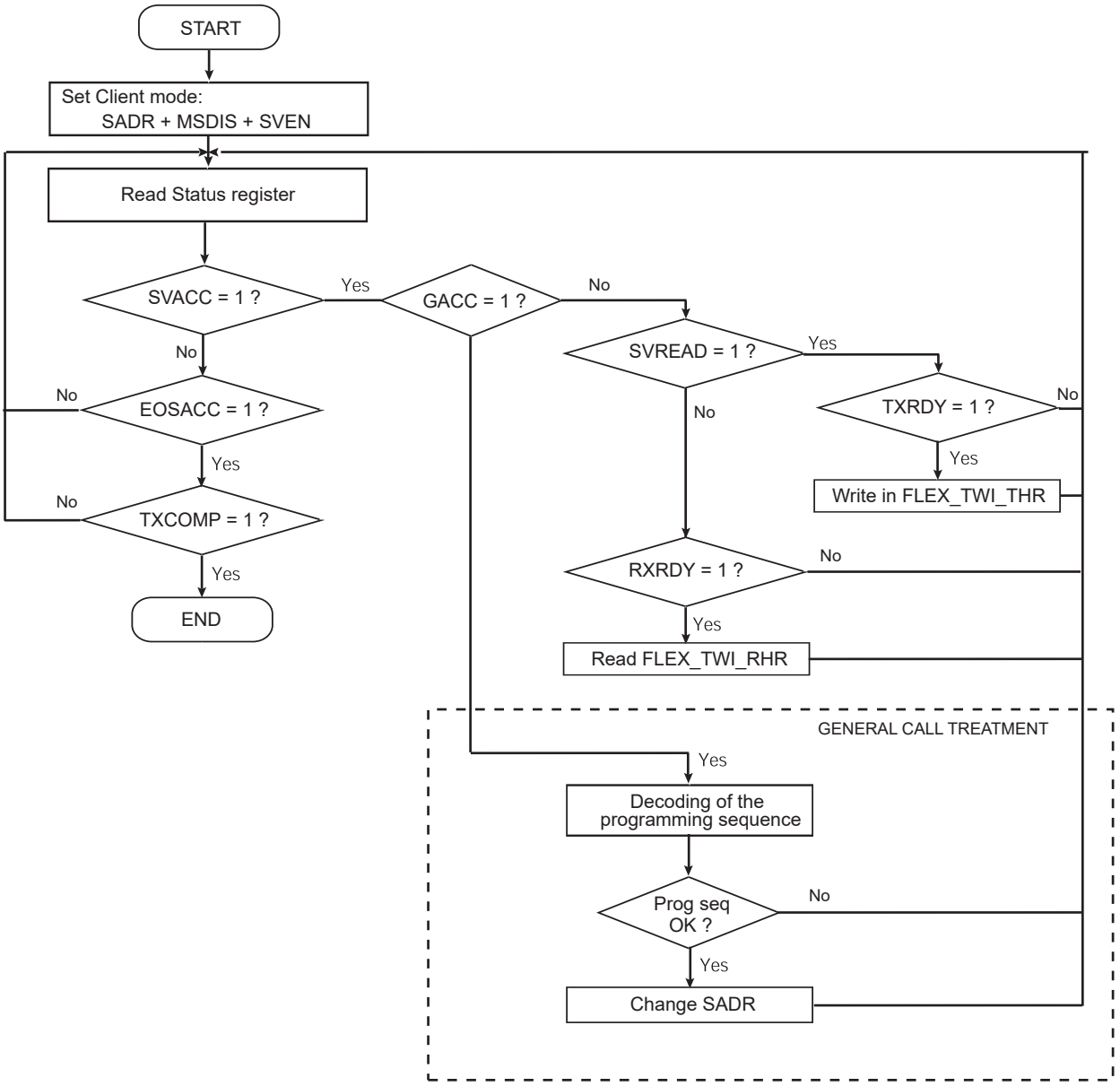


Figure 63.134. Read/Write in Client Mode with SMBus PEC

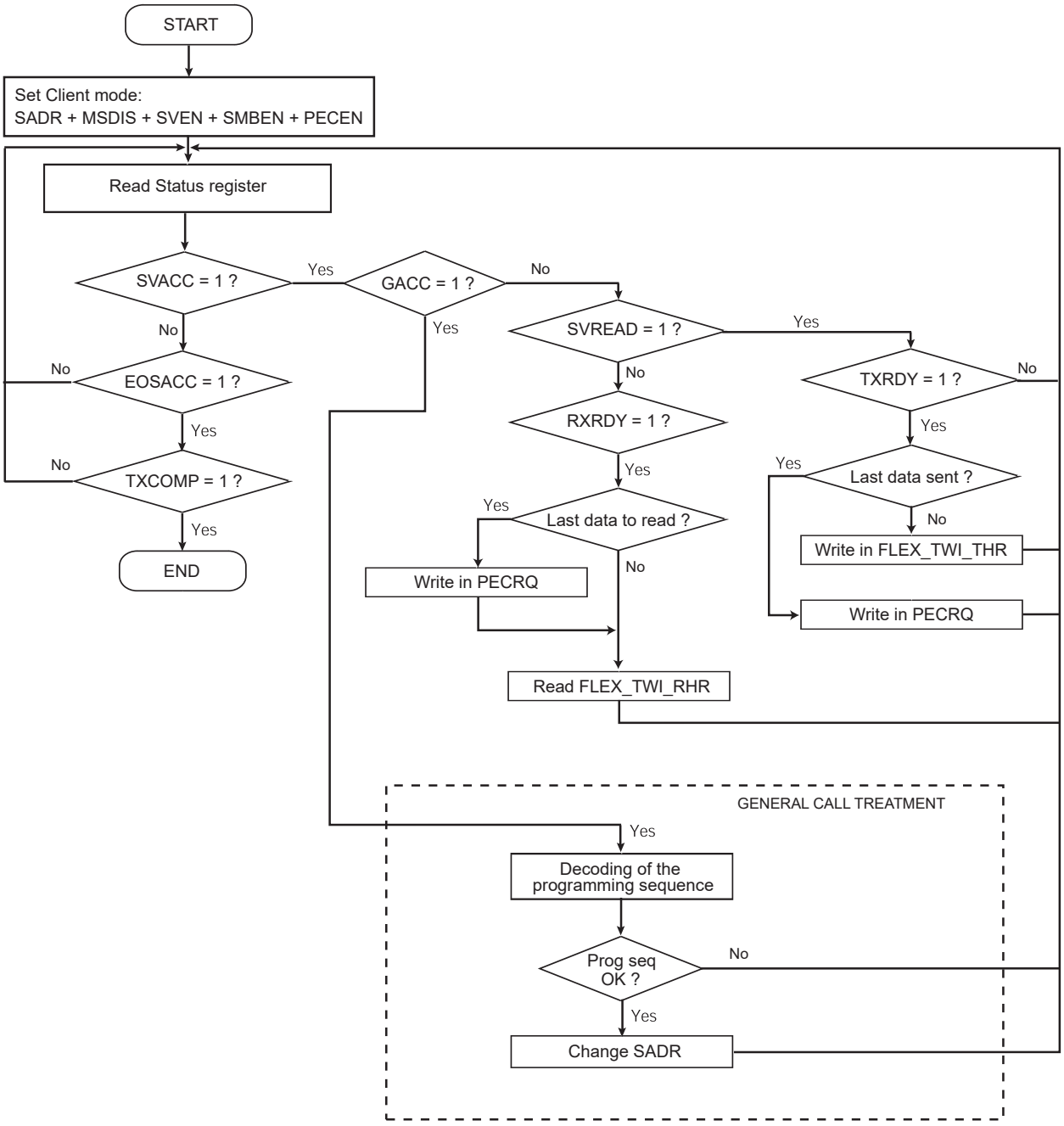
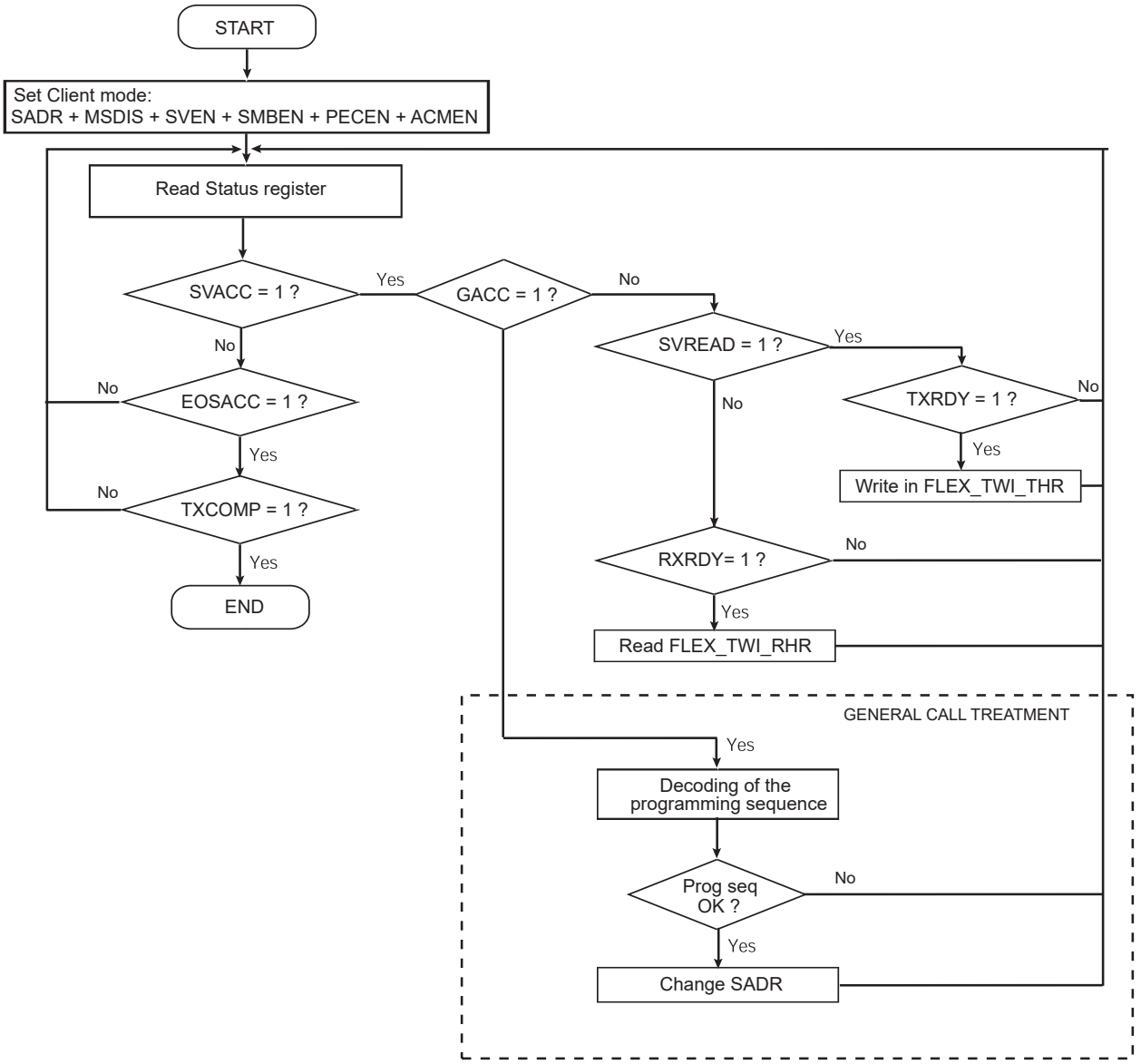


Figure 63.135. Read/Write in Client Mode with SMBus PEC and Alternative Command Mode



63.9.6. TWI FIFOs

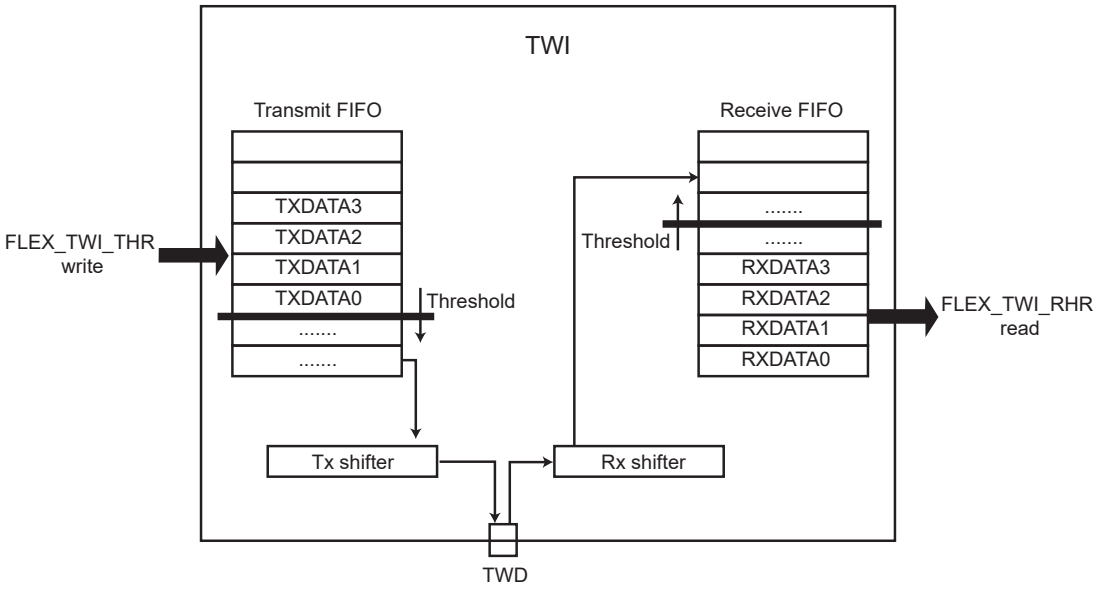
63.9.6.1. Overview

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 16-byte Transmit FIFO and a 16-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 63.136. 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.

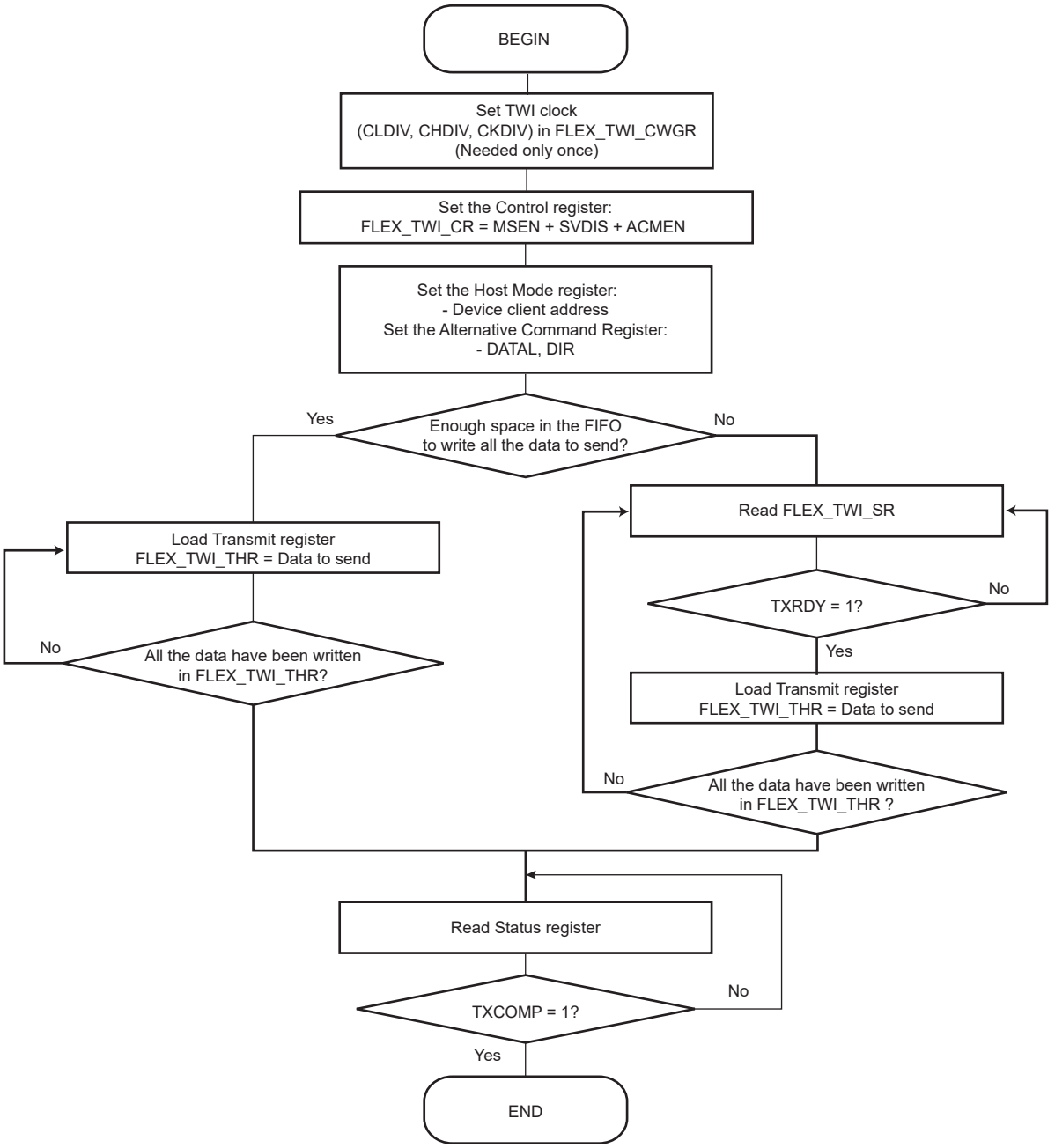
The Transmit FIFO level is provided in FLEX_TWI_FLR.TXFL. If the FIFO can accept the number of bytes to be transmitted, there is no need to monitor FLEX_TWI_SR.TXRDY and the bytes can be successively written in FLEX_TWI_THR.

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 63.137. Sending Data with FIFO Enabled in Host Mode



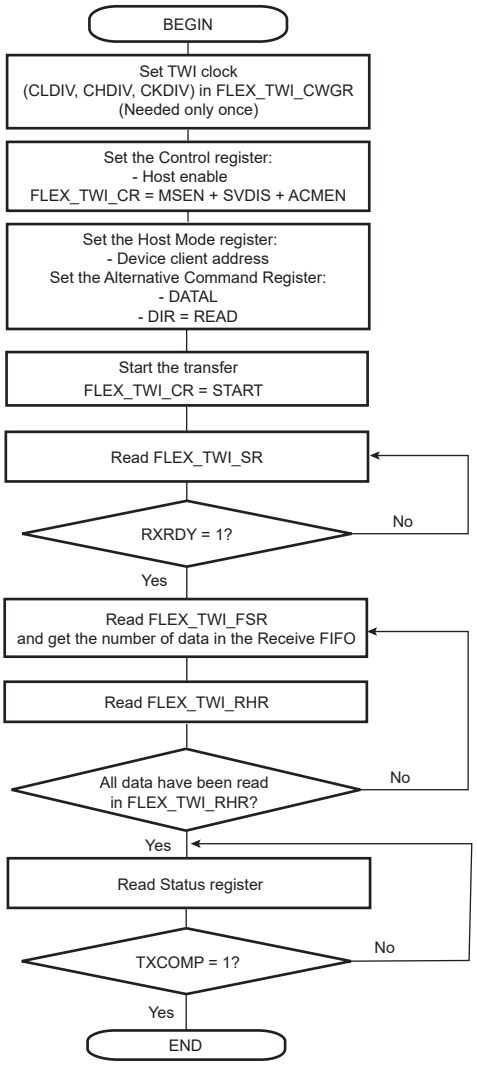
63.9.6.3.Receiving Data with FIFO Enabled

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.

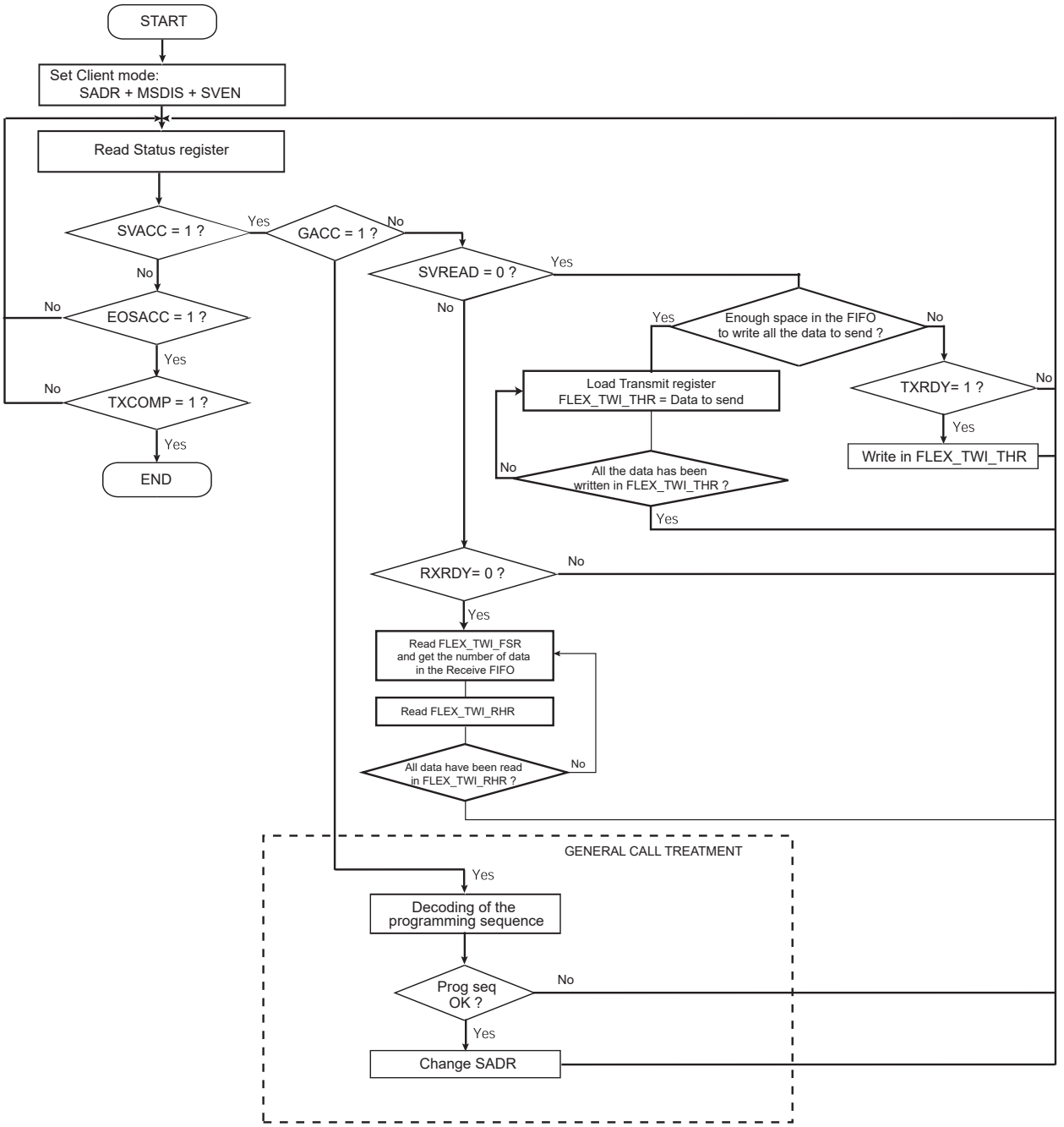
Figure 63.138. Receiving Data with FIFO Enabled in Host Mode



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 63.139. 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 63.140. TXRDY Behavior when TXRDYM = 0 in Host Mode

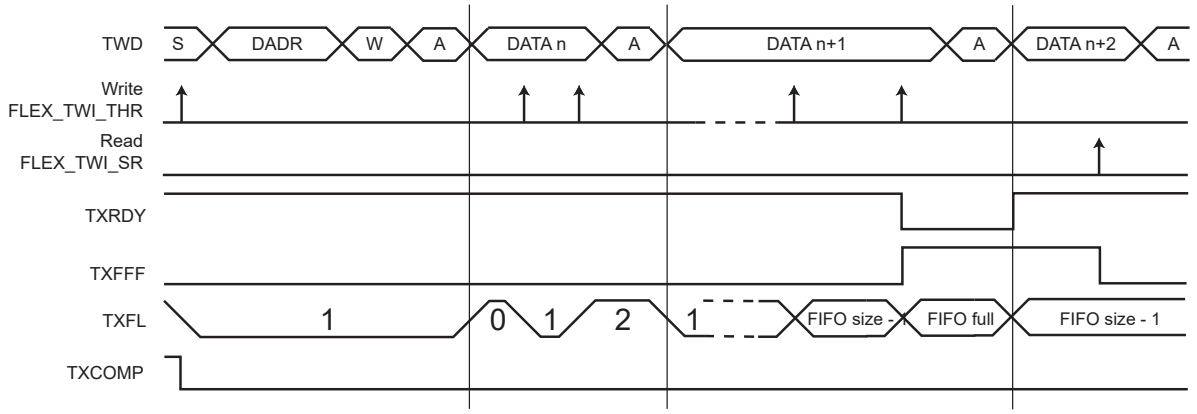


Figure 63.141. RXRDY Behavior when RXRDYM = 0 in Host and Client Modes

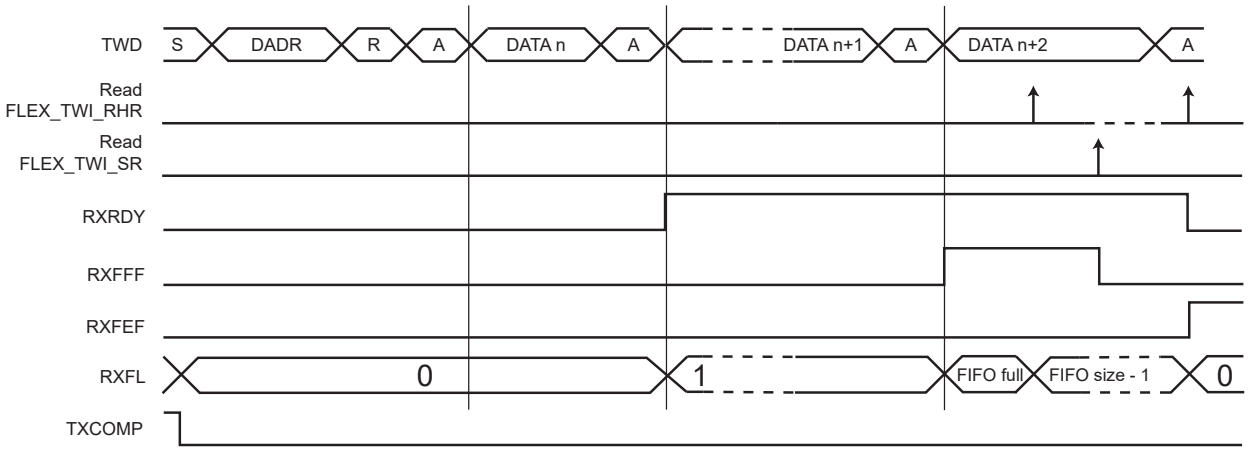
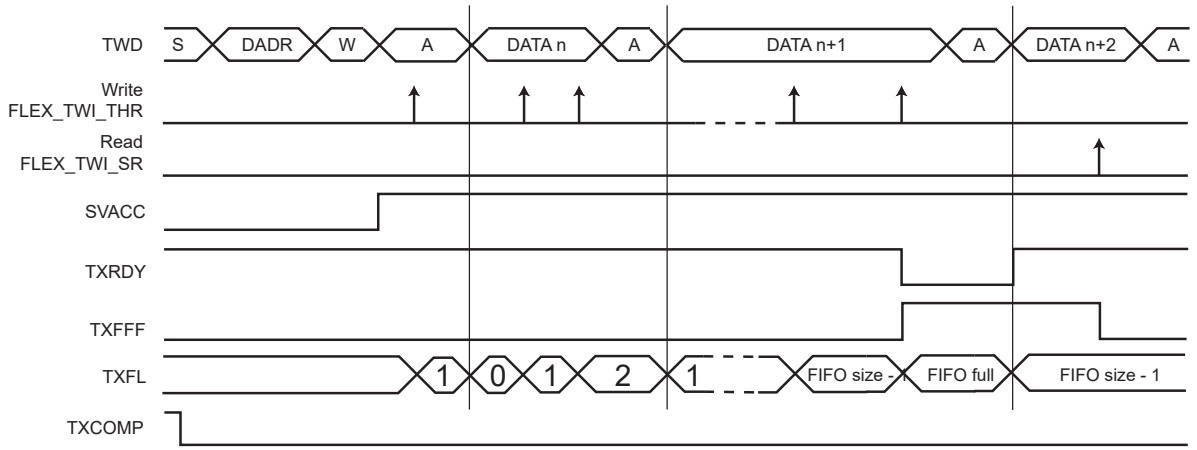


Figure 63.142. TXRDY Behavior when TXRDYM = 0 in Client Mode



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.DMA

The DMA 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 DMA 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 DMA 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 Overflow/Underflow Error

If the Transmit FIFO is full and a write access is performed on FLEX_TWI_THR, it generates a Transmit FIFO overflow 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 overflow error is generated and FLEX_TWI_FSR.TXFPTEF is set.

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 underflow error is generated and FLEX_TWI_FSR.RXFPTEF is set.

No 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 such error occurs, other FIFO flags may not behave as expected; their states should be ignored. A software reset must be performed using FLEX_TWI_CR.SWRST. Note that issuing a software reset during transmission may leave a client in an unknown state holding the TWD line. In this case, a Bus Clear command may instruct the client to release the TWD line (the first frame sent afterward may not be received properly by the client). See Bus Clear Command to initiate the Bus Clear command.

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 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 can be configured to analyze the peripheral TWIn-1 (TWI0 analyzes TWImax) in a full transparent mode via predefined internal connections between TWI instances (n is the index of the TWI instance).

The predefined internal connections provide 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 63.143. Sniffer Mode Application Overview

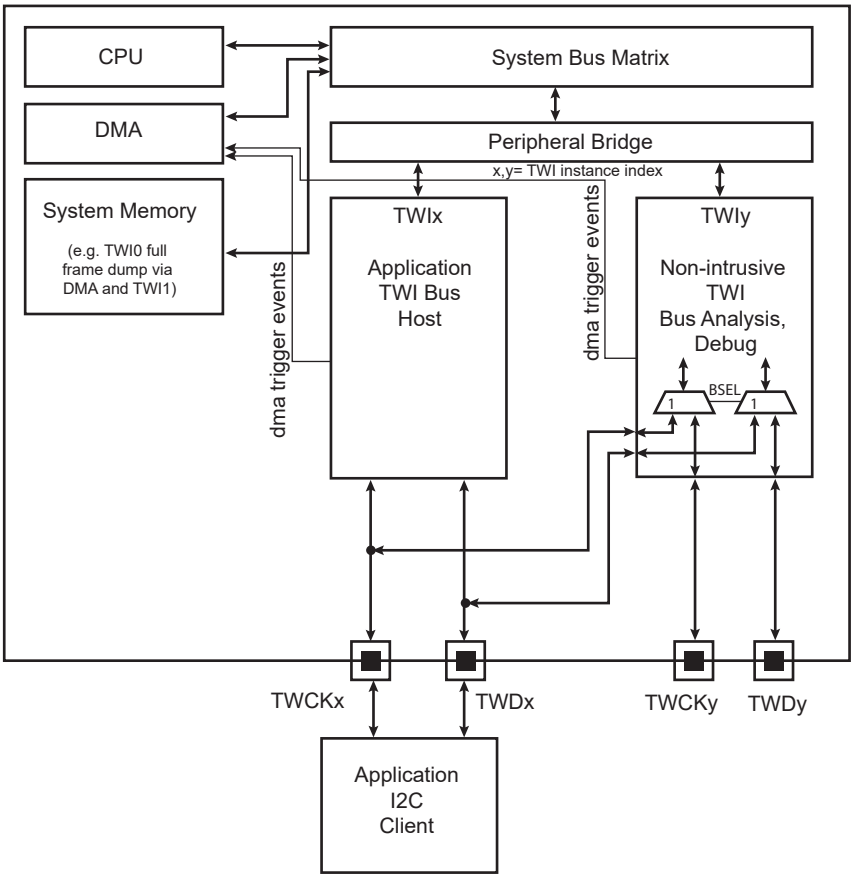
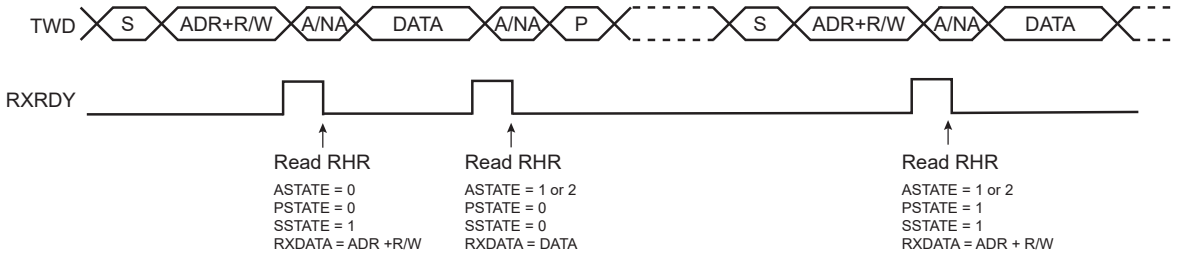


Figure 63.144. Client Sniffer Mode Log



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 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 (DEFAULT_MODE)	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
0x0208	FLEX_US_IER (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x020C	FLEX_US_IDR (DEFAULT_MODE)	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
0x020C	FLEX_US_IDR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x0210	FLEX_US_IMR (DEFAULT_MODE)	31:24								MANE
		23:16		CMP			CTSIC			
		15:8			NACK			ITER	TXEMPTY	TIMEOUT
		7:0	PARE	FRAME	OVRE			RXBRK	TXRDY	RXRDY

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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
0x0210	FLEX_US_IMR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x0214	FLEX_US_CSR (DEFAULT_MODE)	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
0x0214	FLEX_US_CSR (LON_MODE)	31:24				LBLOVFE	LRXD	LFET	LCOL	LTXD
		23:16								
		15:8						UNRE	TXEMPTY	
		7:0	LCRCE	LSFE	OVRE				TXRDY	RXRDY
0x0218	FLEX_US_RHR (DEFAULT_MODE)	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 (DEFAULT_MODE)	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]							
0x0228	FLEX_US_TTGR (LON_MODE)	31:24								
		23:16	PCYCLE[23:16]							
		15:8	PCYCLE[15:8]							
		7:0	PCYCLE[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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0240	FLEX_US_FIDI (LON_MODE)	31:24								
		23:16	BETA2[23:16]							
		15:8	BETA2[15:8]							
		7:0	BETA2[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]			
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	FLEX_US_LONMR	31:24								
		23:16	EOFS[7:0]							
		15:8								
		7:0			LCDS	DMAM	CDTAIL	TCOL	COLDET	COMMT
0x0264	FLEX_US_LONPR	31:24								
		23:16								
		15:8			LONPL[13:8]					
		7:0	LONPL[7:0]							
0x0268	FLEX_US_LONDL	31:24								
		23:16								
		15:8								
		7:0	LONDL[7:0]							
0x026C	FLEX_US_LONL2HDR	31:24								
		23:16								
		15:8								
		7:0	PB	ALTP	BLI[5:0]					
0x0270	FLEX_US_LONBL	31:24								
		23:16								
		15:8								
		7:0			LONBL[5:0]					
0x0274	FLEX_US_LONB1TX	31:24								
		23:16	BETA1TX[23:16]							
		15:8	BETA1TX[15:8]							
		7:0	BETA1TX[7:0]							
0x0278	FLEX_US_LONB1RX	31:24								
		23:16	BETA1RX[23:16]							
		15:8	BETA1RX[15:8]							
		7:0	BETA1RX[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x027C	FLEX_US_LONPRIO	31:24								
		23:16								
		15:8		NPS[6:0]						
		7:0		PSNB[6:0]						
0x0280	FLEX_US_IDTTX	31:24								
		23:16	IDTTX[23:16]							
		15:8	IDTTX[15:8]							
		7:0	IDTTX[7:0]							
0x0284	FLEX_US_IDTRX	31:24								
		23:16	IDTRX[23:16]							
		15:8	IDTRX[15:8]							
		7:0	IDTRX[7:0]							
0x0288	FLEX_US_ICDIFF	31:24								
		23:16								
		15:8								
		7:0					ICDIFF[3:0]			
0x028C ... 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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x02E8	FLEX_US_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS
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	MOSIIE	CSIE	TPMEN	CMPMODE				LSBHALF
		7:0	LLB	CRCEN	WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
0x0408	FLEX_SPI_RDR (DEFAULT_MODE)	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 (DEFAULT_MODE)	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			CRCERR	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			CRCERR	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			CRCERR	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			CRCERR	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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
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	FLEX_SPI_CRCR	31:24					DHRX	DCRX	FHE	CRM
		23:16	FRHL[3:0]							CRCS
		15:8								
		7:0	FRL[7:0]							
0x0450	FLEX_SPI_TPMR	31:24								
		23:16								
		15:8								
		7:0					OSR[1:0]		ALWAYS0	CSM
0x0454	FLEX_SPI_TPHR	31:24								
		23:16								
		15:8								
		7:0		OSR[1:0]		GAIN[1:0]		BOOST	CNT[1:0]	
0x0458 ... 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 (DEFAULT_MODE)	31:24			FIFODIS	FIFOEN		LOCKCLR		THRCLR
		23:16					SCLRBE	SCLRBD	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

Register Summary (continued)

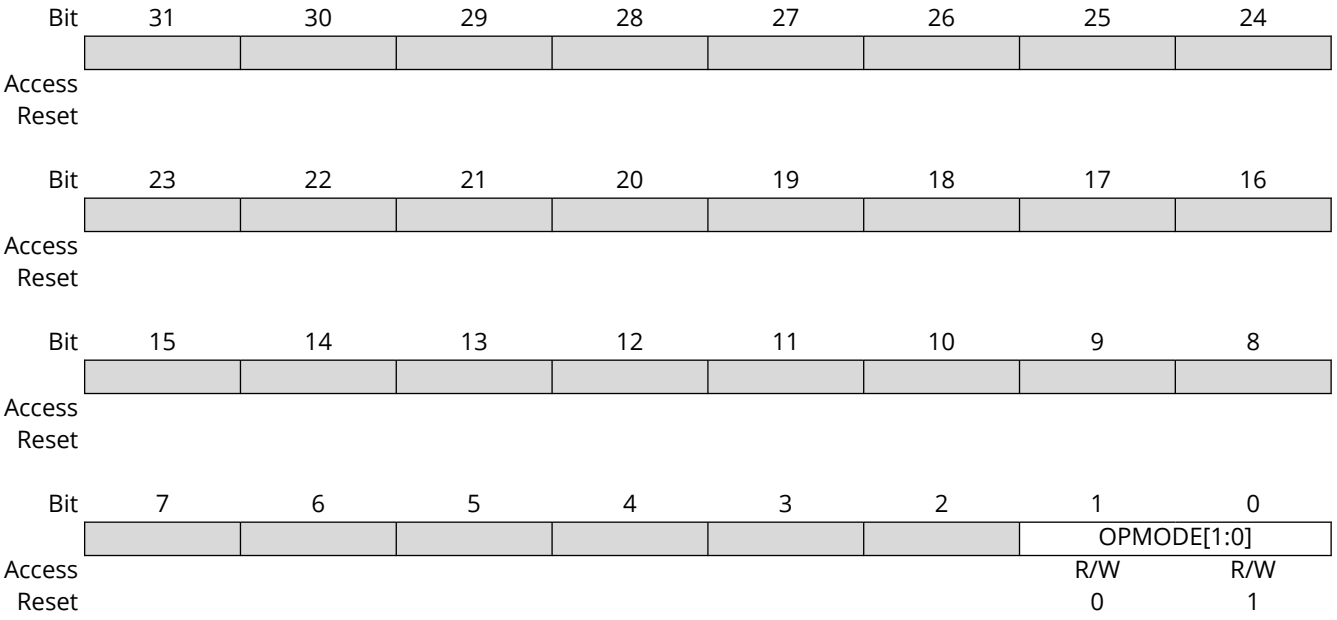
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0604	FLEX_TWI_MMR	31:24								NOAP
		23:16		DADR[6:0]						
		15:8		SCLRBL[1:0]		MREAD			IADRSZ[1:0]	
		7:0								
0x0608	FLEX_TWI_SMR	31:24								
		23:16		SADR[6:0]						
		15:8		MASK[6:0]						
		7:0	SNIFF	SCLWSDIS	BSEL	SADAT	SMHH	SMDA		NACKEN
0x060C	FLEX_TWI_IADR	31:24								
		23:16	IADR[23:16]							
		15:8	IADR[15:8]							
		7:0	IADR[7:0]							
0x0610	FLEX_TWI_CWGR	31:24		HOLD[6:0]						
		23:16				BRSRCCLK		CKDIV[2:0]		
		15:8	CHDIV[7:0]							
		7:0	CLDIV[7:0]							
0x0614 ... 0x061F	Reserved									
0x0620	FLEX_TWI_SR (DEFAULT_MODE)	31:24						SR	SDA	SCL
		23:16	LOCK		SMBHHM	SMBDAM	PECERR	TOUT		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		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 (DEFAULT_MODE)	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 (DEFAULT_MODE)	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]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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]							
0x064C ... 0x064F	Reserved									
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



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, LON) 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

Value	Description
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.
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.
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
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
1	DIV	Peripheral clock divided (DIV = 8) is selected
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
0x9	LON	LON

Value	Name	Description
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 (Default Mode)

Name: FLEX_US_IER (DEFAULT_MODE)
Offset: 0x208
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Enable Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Enable Register (LON_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 Enable Register (LON_MODE)

Name: FLEX_US_IER (LON_MODE)
Offset: 0x208
Reset: –
Property: Write-only

This register is relevant only if USART_MODE = 0x9 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
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				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
						UNRE	TXEMPTY	
Access						W	W	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Enable

Bit 27 – LRXD LON Reception Done Interrupt Enable

Bit 26 – LFET LON Frame Early Termination Interrupt Enable

Bit 25 – LCOL LON Collision Interrupt Enable

Bit 24 – LTXD LON Transmission Done Interrupt Enable

Bit 10 – UNRE Underrun Error Interrupt Enable

Bit 9 – TXEMPTY TXEMPTY Interrupt Enable

Bit 7 – LCRCE LON CRC Error Interrupt Enable

Bit 6 – LSFE LON Short Frame Error Interrupt Enable

Bit 5 – OVRE Overrun Error Interrupt Enable

Bit 1 – TXRDY TXRDY Interrupt Enable

Bit 0 – RXRDY RXRDY Interrupt Enable

63.10.9. USART Interrupt Disable Register (Default Mode)

Name: FLEX_US_IDR (DEFAULT_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

For LIN-specific configurations, see USART Interrupt Disable Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Disable Register (LON_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

63.10.10.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

63.10.11.USART Interrupt Disable Register (LON_MODE)

Name: FLEX_US_IDR (LON_MODE)
Offset: 0x20C
Reset: –
Property: Write-only

This configuration is relevant only if USART_MODE = 0x9 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
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				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
						UNRE	TXEMPTY	
Access						W	W	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	W	W	W				W	W
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Disable

Bit 27 – LRXD LON Reception Done Interrupt Disable

Bit 26 – LFET LON Frame Early Termination Interrupt Disable

Bit 25 – LCOL LON Collision Interrupt Disable

Bit 24 – LTXD LON Transmission Done Interrupt Disable

Bit 10 – UNRE Underrun Error Interrupt Disable

Bit 9 – TXEMPTY TXEMPTY Interrupt Disable

Bit 7 – LCRCE LON CRC Error Interrupt Disable

Bit 6 – LSFE LON Short Frame Error Interrupt Disable

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

63.10.12.USART Interrupt Mask Register (Default Mode)

Name: FLEX_US_IMR (DEFAULT_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Interrupt Mask Register (LIN_MODE).

For LON-specific configurations, see USART Interrupt Mask Register (LON_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

63.10.13.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

63.10.14.USART Interrupt Mask Register (LON_MODE)

Name: FLEX_US_IMR (LON_MODE)
Offset: 0x210
Reset: 0x00000000
Property: Read-only

This configuration is relevant only if USART_MODE = 0x9 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
				LBLOVFE	LRXD	LFET	LCOL	LTXD
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
						UNRE	TXEMPTY	
Access						R	R	
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 28 – LBLOVFE LON Backlog Overflow Error Interrupt Mask

Bit 27 – LRXD LON Reception Done Interrupt Mask

Bit 26 – LFET LON Frame Early Termination Interrupt Mask

Bit 25 – LCOL LON Collision Interrupt Mask

Bit 24 – LTXD LON Transmission Done Interrupt Mask

Bit 10 – UNRE Underrun Error Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 7 – LCRCE LON CRC Error Interrupt Mask

Bit 6 – LSFE LON Short Frame Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

63.10.15.USART Channel Status Register (Default Mode)

Name: FLEX_US_CSR (DEFAULT_MODE)
Offset: 0x214
Reset: 0x00000000
Property: Read-only

For LIN-specific configurations, see USART Channel Status Register (LIN_MODE).

For LON-specific configurations, see USART Channel Status Register (LON_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 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 TXEMPTY, TXRDY and RXRDY Behavior.

63.10.16.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.

63.10.17.USART Channel Status Register (LON_MODE)

Name: FLEX_US_CSR (LON_MODE)
Offset: 0x214
Reset: –
Property: Read-only

This configuration is relevant only if USART_MODE = 0x9 in the USART Mode Register.

Bit	31	30	29	28	27	26	25	24
				LBLOVFE	LRXD	LFET	LCOL	LTXD
Access				R	R	R	R	R
Reset				–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						UNRE	TXEMPTY	
Access						R	R	
Reset						–	–	

Bit	7	6	5	4	3	2	1	0
	LCRCE	LSFE	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	–	–	–				–	–

Bit 28 – LBLOVFE LON Backlog Overflow Error

Value	Description
0	No backlog overflow error occurred since the last RSTSTA command was issued.
1	At least one backlog error overflow occurred since the last RSTSTA command was issued.

Bit 27 – LRXD LON Reception End Flag

Value	Description
0	Reception on going or no reception occurred since the last RSTSTA command was issued.
1	At least one reception has been performed since the last RSTSTA command was issued.

Bit 26 – LFET LON Frame Early Termination

Value	Description
0	No frame has been terminated early due to collision detection since the last RSTSTA command was issued.
1	At least one transmission has been terminated due to collision detection since the last RSTSTA command was issued. (This stops the DMA until reset with RSTSTA bit).

Bit 25 – LCOL LON Collision Detected Flag

Value	Description
0	No collision occurred while transmitting since the last RSTSTA command was issued.
1	At least one collision occurred while transmitting since the last RSTSTA command was issued.

Bit 24 – LTXD LON Transmission End Flag

Value	Description
0	Transmission on going or no transmission occurred since the last RSTSTA command was issued.
1	At least one transmission has been performed since the last RSTSTA command was issued.

Bit 10 – UNRE Underrun Error

Value	Description
0	No LON underrun error has occurred since the last RSTSTA command was issued.
1	At least one LON underrun error has occurred 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 7 – LCRCE LON CRC Error

Value	Description
0	No CRC error has been detected since the last RSTSTA command was issued.
1	At least one CRC error has been detected since the last RSTSTA command was issued.

Bit 6 – LSFE LON Short Frame Error

Value	Description
0	No short frame received since the last RSTSTA command was issued.
1	At least one short frame received 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.

63.10.18.USART Receive Holding Register (Default Mode)

Name: FLEX_US_RHR (DEFAULT_MODE)
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 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.

63.10.19.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 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.

63.10.20.USART Transmit Holding Register (Default Mode)

Name: FLEX_US_THR (DEFAULT_MODE)
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 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.

63.10.21.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 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.

63.10.22.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)

63.10.23.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.

63.10.24.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.

For LON-specific configurations, see USART Transmitter Timeguard Register (LON_MODE).

Bit	31	30	29	28	27	26	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.

63.10.25.USART Transmitter Timeguard Register (LON_MODE)

Name: FLEX_US_TTGR (LON_MODE)
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
	PCYCLE[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
	PCYCLE[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
	PCYCLE[7:0]							
Access	R/W	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 – PCYCLE[23:0] LON PCYCLE Length

Value	Description
1–16777215	LON PCYCLE length in t_{bit}.

63.10.26.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.

For LON-specific configurations, see USART Transmitter Timeguard Register (LON_MODE).

Bit	31	30	29	28	27	26	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.

63.10.27.USART FI DI RATIO Register (LON_MODE)

Name: FLEX_US_FIDI (LON_MODE)
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
	BETA2[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
	BETA2[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
	BETA2[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 23:0 – BETA2[23:0] LON BETA2 Length

Value	Description
1–16777215	LON BETA2 length in t_{bit}. **Note:** The configured value must be lower than 2^{18}.

63.10.28.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.

63.10.29.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:

$t_{peripheral\ clock} \times (IRDA\_FILTER + 3) < 1.41\ \mu s$

63.10.30.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.

Value	Name	Description
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.

63.10.31.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 DMA Mode

Value	Description
0	The LIN mode register FLEX_US_LINMR is not written by the DMA.
1	The LIN mode register FLEX_US_LINMR (excepting that flag) is written by the DMA.

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.

Value	Description
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.
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.

63.10.32.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.

63.10.33.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

63.10.34.USART LON Mode Register

Name: FLEX_US_LONMR
Offset: 0x260
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	EOFS[7:0]							
Access	R/W	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
			LCDS	DMAM	CDTAIL	TCOL	COLDET	COMMT
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 23:16 – EOFS[7:0] End of Frame Condition Size

Value	Description
0–255	Defines the minimum transitionless time for the IP to detect a LON end of frame condition. t_{eof} = (EOFS+1) × t_{clock} × 8 × (2- OVER)

Bit 5 – LCDS LON Collision Detection Source

Value	Description
0	LON collision detection source is external.
1	LON collision detection source is internal.

Bit 4 – DMAM LON DMA Mode

Value	Description
0	The LON data length register FLEX_US_LONDL is not written by the DMA.
1	The LON data length register FLEX_US_LONDL is written by the DMA.

Bit 3 – CDTAIL LON Collision Detection on Frame Tail

Value	Description
0	Detect collisions after CRC has been sent but prior end of transmission in LON comm_type = 1 mode.
1	Ignore collisions after CRC has been sent but prior end of transmission in LON comm_type = 1 mode.

Bit 2 – TCOL Terminate Frame upon Collision Notification

Value	Description
0	Do not terminate the frame in LON comm_type = 1 mode upon collision detection.
1	Terminate the frame in LON comm_type = 1 mode upon collision detection if possible.

Bit 1 – COLDET LON Collision Detection Feature

Value	Description
0	LON collision detection feature disabled.
1	LON collision detection feature enabled.

Bit 0 – COMMT LON comm_type Parameter Value

Value	Description
0	LON comm_type = 1 mode.
1	LON comm_type = 2 mode.

63.10.35.USART LON Preamble Register

Name: FLEX_US_LONPR
Offset: 0x264
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
			LONPL[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
	LONPL[7:0]							
Access	R/W	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 – LONPL[13:0] LON Preamble Length

Value	Description
1–16383	LON preamble length in t_{bit} (without byte-sync).

63.10.36.USART LON Data Length Register

Name: FLEX_US_LONDL
Offset: 0x268
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	LONDL[7:0]							
Access	R/W	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 – LONDL[7:0] LON Data Length

Value	Description
0–255	LON data length is LONDL + 1 byte.

63.10.37.USART LON L2HDR Register

Name: FLEX_US_LONL2HDR
Offset: 0x26C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	PB	ALTP	BLI[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 – PB LON Priority Bit

Value	Description
0	LON priority bit reset.
1	LON priority bit set.

Bit 6 – ALTP LON Alternate Path Bit

Value	Description
0	LON alternate path bit reset.
1	LON alternate path bit set.

Bits 5:0 – BLI[5:0] LON Backlog Increment

Value	Description
0–63	LON backlog increment to be generated as a result of delivering the LON frame.

63.10.38.USART LON Backlog Register

Name: FLEX_US_LONBL
Offset: 0x270
Reset: 0x00000000
Property: Read-only

This register is relevant only if USART_MODE = 0x9 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
			LONBL[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 – LONBL[5:0] LON Node Backlog Value

Value	Description
1–63	LON node backlog value.

63.10.39.USART LON Beta1 Tx Register

Name: FLEX_US_LONB1TX
Offset: 0x274
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	BETA1TX[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
	BETA1TX[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
	BETA1TX[7:0]							
Access	R/W	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 – BETA1TX[23:0] LON Beta1 Length after Transmission

Value	Description
1–16777215	LON beta1 length after transmission in t_{bit}. **Note:** The configured value must be lower than 2^{18}.

63.10.40.USART LON Beta1 Rx Register

Name: FLEX_US_LONB1RX
Offset: 0x278
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	BETA1RX[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
	BETA1RX[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
	BETA1RX[7:0]							
Access	R/W	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 – BETA1RX[23:0] LON Beta1 Length after Reception

Value	Description
1–16777215	LON beta1 length after reception in t_{bit}. **Note:** The configured value must be lower than 2^{18}.

63.10.41.USART LON Priority Register

Name: FLEX_US_LONPRIO
Offset: 0x27C
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
		NPS[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
		PSNB[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 – NPS[6:0] LON Node Priority Slot

Value	Description
0–127	Node priority slot.

Bits 6:0 – PSNB[6:0] LON Priority Slot Number

Value	Description
0–127	Number of priority slots in the LON network configuration.

63.10.42.USART LON IDT Tx Register

Name: FLEX_US_IDTTX
Offset: 0x280
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	IDTTX[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
	IDTTX[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
	IDTTX[7:0]							
Access	R/W	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 – IDTTX[23:0] LON Indeterminate Time after Transmission (comm_type = 1 mode only)

Value	Description
0–16777215	LON indeterminate time after transmission in t_{bit}. **Note:** The configured value must be lower than 2^{18}.

63.10.43.USART LON IDT Rx Register

Name: FLEX_US_IDTRX
Offset: 0x284
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
	IDTRX[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
	IDTRX[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
	IDTRX[7:0]							
Access	R/W	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 – IDTRX[23:0] LON Indeterminate Time after Reception (comm_type = 1 mode only)

Value	Description
0–16777215	LON indeterminate time after reception in t_{bit}. **Note:** The configured value must be lower than 2^{18}.

63.10.44.USART IC DIFF Register

Name: FLEX_US_ICDIFF
Offset: 0x288
Reset: 0x00000000
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 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
					ICDIFF[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:0 – ICDIFF[3:0] IC Differentiator Number

63.10.45.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.

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.

63.10.46.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
0–16	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
0–16	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
0–16	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).

63.10.47.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.
1–16	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.
1–16	Indicates the number of data in the Transmit FIFO.

63.10.48.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

63.10.49.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

63.10.50.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

63.10.51.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 Underflow Error Flag
See FIFO Overflow/Underflow Error for details.

Value	Description
0	No Receive FIFO underflow occurred.
1	Receive FIFO underflow error occurred due to an incorrect software access (read data exceeding available data in FIFO). Receiver must be reset by writing FLEX_SPI_CR.SWRST=1.

Bit 6 – TXFPTEF Transmit FIFO Overflow Error Flag
See FIFO Overflow/Underflow Error for details.

Value	Description
0	No Transmit FIFO overflow occurred.
1	Transmit FIFO overflow error occurred due to an incorrect software access (written data exceeding available space). Transceiver must be reset by writing FLEX_SPI_CR.SWRST=1.

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.

63.10.52.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).

63.10.53.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.

63.10.54.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

Value	Description
0	No effect.
1	Restarts the comparison trigger to enable FLEX_SPI_RDR.

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.

63.10.55.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
	MOSIIE	CSIE	TPMEN	CMPMODE				LSBHALF
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
	LLB	CRCEN	WDRBT	MODFDIS	BRSRCCLK	PCSDEC	PS	MSTR
Access	R/W	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 – 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 15 – MOSIIE MOSI Inversion Enable

Value	Description
0	MOSI input is not inverted.
1	MOSI input is internally inverted before being processed by the SPI.

Bit 14 – CSIE Chip Select Inversion Enable

Value	Description
0	Chip select NPCS active polarity is low.
1	Chip select NPCS active polarity is high.

Bit 13 – TPMEN Two-Pin Mode Enable

Value	Description
0	Two-Pin mode is disabled.
1	Two-Pin mode is enabled.

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 8 – LSBHALF Last Bit Half Period Compatibility

Value	Description
0	Normal SPI mode of operation: all bits have a 1-bit time duration (standard SPI mode)
1	Compatibility mode when connected to an SPI client having only a half bit period duration time for the last bit of the frame (non-standard SPI mode)

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 6 – CRCEN CRC Enable

Value	Description
0	CRC calculation is disabled.
1	CRC calculation is enabled. BITS in FLEX_SPI_CSRx registers must be at value '0'.

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.

Value	Name	Description
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.

63.10.56.SPI Receive Data Register (Default Mode)

Name: FLEX_SPI_RDR (DEFAULT_MODE)
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.

63.10.57.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.

63.10.58.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.

63.10.59.SPI Transmit Data Register (Default Mode)

Name: FLEX_SPI_TDR (DEFAULT_MODE)
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 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.

63.10.60.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.

63.10.61.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
			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
Access			R	R	R	R	R	R
Reset			0	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 Underflow Error Flag
See FIFO Overflow/Underflow 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 underflow occurred.
1	Receive FIFO underflow error occurred due to an incorrect software access (read data exceeding available data in FIFO). Receiver must be reset by writing FLEX_SPI_CR.SWRST=1.

Bit 30 – TXFPTEF Transmit FIFO Overflow Error Flag
See FIFO Overflow/Underflow 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 overflow occurred.
1	Transmit FIFO overflow error occurred due to an incorrect software access (written data exceeding available space). Transceiver must be reset by writing FLEX_SPI_CR.SWRST=1.

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 13 – CRCERR CRC Error (cleared on read)

Value	Description
0	CRC calculation is disabled or no received frame contains a CRC error since the last read of FLEX_SPI_SR.
1	Since the last read of FLEX_SPI_SR, a received frame contains a CRC error.

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.
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 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 TXEMPTY, TDRE and RDRF Behavior.

63.10.62.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
			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
Access			W	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 13 – CRCERR CRC Error 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

63.10.63.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
			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
Access			W	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 13 – CRCERR CRC Error 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

63.10.64.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
			CRCERR	SFERR	CMP	UNDES	TXEMPTY	NSSR
Access			R	R	R	R	R	R
Reset			0	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 13 – CRCERR CRC Error 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

63.10.65.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 × $f_{peripheral\ clock}$ / 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 × $f_{peripheral\ clock}$
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 = $f_{peripheral\ clock}$ / 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: $\frac{\text{DLYBCS}}{\text{f}_{\text{peripheral clock}}}$(if DLYBCS ≠ 0)
If FLEX_SPI_MR.BRSRCCLK = 1: $\frac{\text{DLYBCS}}{\text{f}_{GCLK}}$
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.

63.10.66.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
0–16	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
0–16	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 transferred on SPI line), the chunk of 1 byte must be configured in the DMA. When FLEX_SPI_CSR0.BITS>0 (9 to 16 bits transferred 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 transferred 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 transferred 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 transferred 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 transferred 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. 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).

63.10.67.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.
1–16	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.
1–16	Indicates the number of data in the Transmit FIFO.

63.10.68.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.

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.

63.10.69.SPI CRC Register

Name: FLEX_SPI_CRCR
Offset: 0x44C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
					DHRX	DCRX	FHE	CRM
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	FRHL[3:0]							CRCS
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
	FRL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 27 – DHRX Disable Header Receiving

Value	Description
0	Header is received and supplied to FLEX_SPI_RDR as a classic data.
1	Header is received and checked but not supplied to FLEX_SPI_RDR (RDRF will not rise upon Header receiving).

Bit 26 – DCRX Disable CRC Receiving

Value	Description
0	CRC is received and supplied to FLEX_SPI_RDR as a classic data.
1	CRC is received and checked but not supplied to FLEX_SPI_RDR (RDRF will not rise upon CRC receiving).

Bit 25 – FHE Frame Header Excluded

Value	Description
0	The frame header is included in the CRC calculation.
1	The frame header is excluded from the CRC calculation.

Bit 24 – CRM Continuous Read Mode

Value	Description
0	A header is sent every frame in case of contiguous frames (without de-asserting the corresponding NPCS).
1	A header is sent only on the first frame in case of contiguous frames (without de-asserting the corresponding NPCS).

Bits 23:20 – FRHL[3:0] Frame Header Length
If FLEX_SPI_MR.TPMEN= 0, this value is the length of the frame header, in bytes.

If FLEX_SPI_MR.TPMEN= 1, this value is the length of the frame header minus 1, in bytes.

Bit 16 – CRCS CRC Size

Value	Name	Description
0	16B_CRC	CRC size is 16 bits.
1	32B_CRC	CRC size is 32 bits.

Bits 7:0 – FRL[7:0] Frame Length

If FLEX_SPI_MR.TPMEN= 0, this value is the length of the frame (header and CRC included), in bytes.
If FLEX_SPI_MR.TPMEN= 1, this value is the length in bytes of the frame starting from the SYNC field (included) with CRC included.

63.10.70.SPI Two-Pin Mode Register

Name: FLEX_SPI_TPMR
Offset: 0x450
Reset: 0x00000000
Property: Read/Write

This register can only be written if WPEN is cleared in the SPI 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
					OSR[1:0]		ALWAYS0	CSM
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bits 3:2 – OSR[1:0] Oversampling Rate
Defines the MCP3910 OSR setting used.

Bit 1 – ALWAYS0 Always Written to 0

Bit 0 – CSM Chip Select Mode

Value	Description
0	Chip select is not driven.
1	Chip select is driven and can be used to control enable pin of the external device. Depending on FLEX_SPI_MR.PCSDEC, an external decoder can be used to minimize the number of IOs used.

63.10.71.SPI Two-Pin Header Register

Name: FLEX_SPI_TPHR
Offset: 0x454
Reset: 0x00000000
Property: Read-only

This register can only be written if WPEN is cleared in the SPI 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
		OSR[1:0]		GAIN[1:0]		BOOST	CNT[1:0]	
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 6:5 – OSR[1:0] Oversampling Rate
Returns the MCP3910 Oversampling Rate setting.

Bits 4:3 – GAIN[1:0] Gain
Returns the MCP3910 Gain setting.

Bit 2 – BOOST Current Boost
Returns the MCP3910 Current Boost setting.

Bits 1:0 – CNT[1:0] Frame Counter
Returns the MCP3910 Frame Counter value.

63.10.72.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 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).

63.10.73.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.

63.10.74.TWI Control Register (Default Mode)

Name: FLEX_TWI_CR (DEFAULT_MODE)
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
					SCLRBE	SCLRBD	ACMDIS	ACMEN
Access					W	W	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 19 – SCLRBE SCL Rise Boost Enable

Value	Description
0	No effect.
1	SCL rise time is boosted in High-Speed mode. Duration of the boost is configured with FLEX_TWI_MMR.SCLRBL. See SCL Rising Time Control for details.

Bit 18 – SCLRBD SCL Rise Boost Disable

Value	Description
0	No effect.
1	SCL rise time boost is disabled. See SCL Rising Time Control for details.

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

When TWD (SDA)=0, the Bus Clear command must be performed via the PIO. When TWCK=0, no Bus Clear command can be issued.

Value	Description
0	No effect.
1	When Host mode is enabled and TWD (SDA)=1, sends 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.

Value	Description
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).

63.10.75.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 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).

63.10.76.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

63.10.77.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
Access								
Reset								

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

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	TWIn analyzes the TWCK and TWD pins of the peripheral TWIn-1 (TWI0 analyzes TWImax).

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.

63.10.78.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.

63.10.79.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[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
				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 30:24 – HOLD[6:0] 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}

63.10.80.TWI Status Register (Default Mode)

Name: FLEX_TWI_SR (DEFAULT_MODE)
Offset: 0x620
Reset: 0x03000009
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		MCACK
Access	R		R	R	R	R		R
Reset	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 Handling Errors in Alternative Command and TWI FIFOs).

Bit 21 – SMBHHM SMBus Host Header Address Match (cleared on read)

Value	Description
0	No SMBus Host Header Address received.

Value	Description
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 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.

63.10.81.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		MCACK
Access	R		R	R	R	R		R
Reset	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 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.

63.10.82.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

63.10.83.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

63.10.84.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

63.10.85.TWI Receive Holding Register (Default Mode)

Name: FLEX_TWI_RHR (DEFAULT_MODE)
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

63.10.86.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

63.10.87.TWI Transmit Holding Register (Default Mode)

Name: FLEX_TWI_THR (DEFAULT_MODE)
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

63.10.88.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

63.10.89.TWI SMBus Timing Register

Name: FLEX_TWI_SMBTR
Offset: 0x638
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in FLEX_TWI_WPMR.

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

63.10.90.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

63.10.91.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 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.

Value	Description
1	Read direction.

Bits 7:0 – DATAL[7:0] Data Length

Value	Description
0	No data to send (see Alternative Command).
1–255	Number of bytes to send during the transfer.

63.10.92.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)

63.10.93.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 = $(t_{low}/t_{ext\_ck})/2^{CKDIV}$

63.10.94.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
0–16	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
0–16	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).

63.10.95.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.
1–16	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.
1–16	Indicates the number of data in the Transmit FIFO.

63.10.96.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 Underflow Error Flag
See FIFO Overflow/Underflow Error for details.

Value	Description
0	No Receive FIFO underflow occurred.
1	Receive FIFO underflow error occurred due to an incorrect software access (read data exceeding available data in FIFO). Receiver must be reset by writing FLEX_SPI_CR.SWRST=1.

Bit 6 – TXFPTEF Transmit FIFO Overflow Error Flag
See FIFO Overflow/Underflow Error for details.

Value	Description
0	No Transmit FIFO overflow occurred.
1	Transmit FIFO overflow error occurred due to an incorrect software access (written data exceeding available space). Transceiver must be reset by writing FLEX_SPI_CR.SWRST=1.

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.

63.10.97.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

63.10.98.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

63.10.99.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

63.10.100.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).

63.10.101.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 allows the system to execute code directly from a serial Flash memory (XiP) without code shadowing to RAM. In the system, the mapped serial Flash memory is seen as any other memory.

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, instead of larger and more expensive parallel Flash memories.

The QSPI can be used in SPI legacy mode to interface to serial peripherals such as ADCs, DACs, LCD controllers, CAN controllers and sensors using register accesses, or in Serial Memory mode to interface to serial Flash memories or other devices in an automated way.

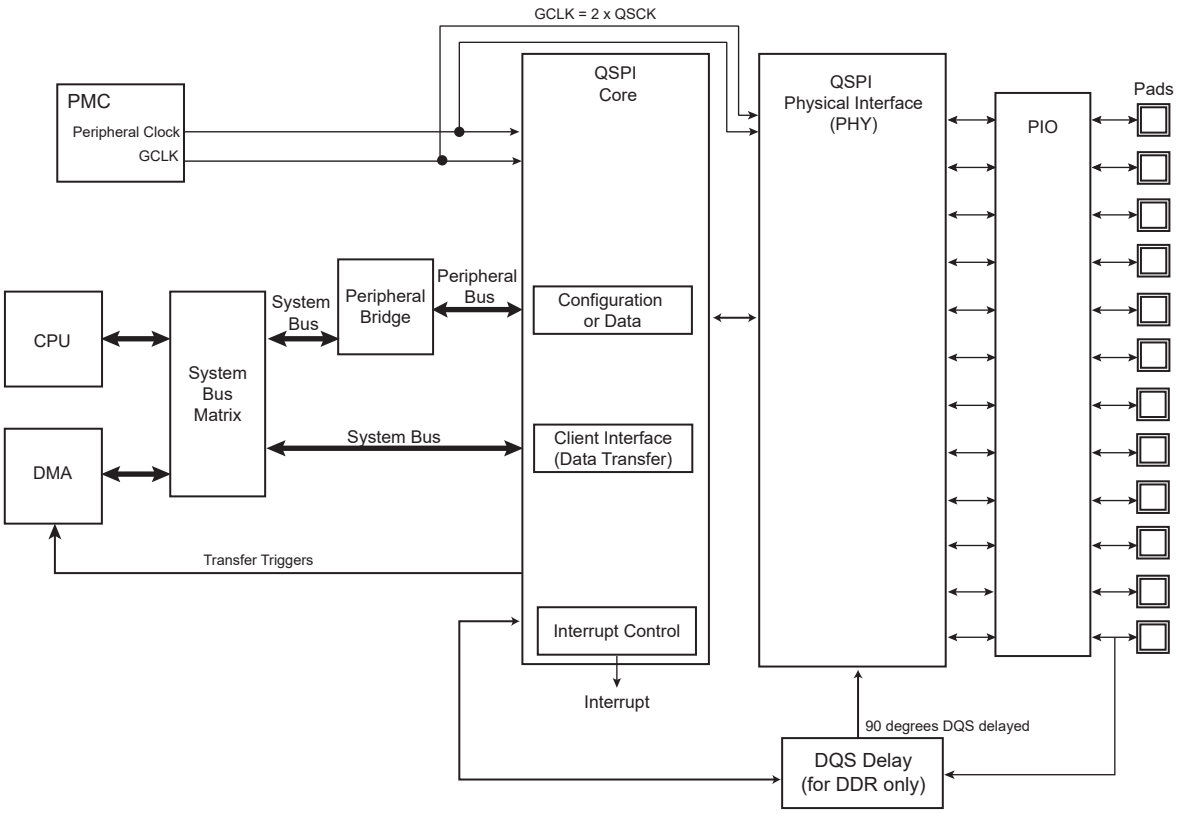
Note: Stacked devices with a rollover in the memory address space at each die boundary are not supported.

64.2. Embedded Characteristics

- Host SPI Interface
  - Octal/Quad/Dual/Single/Twin-Quad communication support
  - Single Data Rate (SDR) and Dual Data Rate (DDR) support
  - Flash/NANDmemory support (supports various vendors and modes)
  - Supports “Execute in Place” (XiP)—code execution by the system directly from a serial memory
- Legacy SPI Mode
  - Interface to serial peripherals such as ADCs and sensors
  - 8-bit/16-bit programmable data length
- Serial Memory Mode
  - Versatile instruction and timing registers for compatibility with all serial Flash memories and SPI devices
  - Up to 32-bit address mode to support serial Flash memories larger than 128 Mbits
  - “On-the-fly” zero latency scrambling/unscrambling
- Functional Safety Monitors and Reports
  - Abnormal functional behavior reports (access to undefined device address, access to locked registers, abnormal DMA requests, etc.)
  - Register write protection
- Connection to DMA Channel Capabilities for DMA Chip-Wide Integration
  - One channel for the receiver, one channel for the transmitter
- Supported standards are:
  - JESD251 (xSPI)
  - JESD251-1 (x4 Quad I/O with Data Strobe)
  - JESD216D (SFDP)

64.3. Block Diagram

Figure 64.1. QSPI Block Diagram



64.4. Signal Description

Table 64.1. Signal Description for External IOs

Pin Name	Pin Description	Type
QSCK	Serial clock	Output
MOSI (QIO0)[123]	Data output (data input/output 0)	Output (input/output)
MISO (QIO1)[123]	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[4]	Data input/output 5	Input/output
QIO6[4]	Data input/output 6	Input/output
QIO7[4]	Data input/output 7	Input/output
QCS	Peripheral chip select	Output
QINT	Optional. Interrupt output of an external memory device. Set to 0 if not used.	Input
QDQS[5678]	Data strobe (input for read accesses, output for write accesses)	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. QDQS is supplied by most Octal SPI memories.
6. Pre-cycle is not supported on the QDQS signal.
7. Preamble bits are not supported on the QDQS signal.
8. OCTAL SDR with DQS is not supported (QSPI_IFR.DDREN=0, QSPI_IFR.WIDTH=OCT_OUTPUT/OCT_IO/OCT_CMD, QSPI_IFR.DQSEN=1).

64.5. Product Dependencies

64.5.1. I/O Lines

The pins used to interface 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 must 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 (DMAC)

The QSPI can be used in conjunction with the system-wide Direct Memory Access Controller (DMAC) in order to reduce processor overhead. For a full description of the DMAC, refer to the section “DMA Controller (XDMAC)”.

64.6. Functional Description

64.6.1. Register Synchronization

As the Quad Serial Peripheral Interface and the QSPI Controller core use different clocks, the following events must be synchronized with the core after being configured:

- 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 (synchronization only when QSPI_MR.SMM is set to ‘1 ‘)
- QSPI_TDR.TD
- QSPI_WRACNT.NBWRA

Before accessing any of these bits/fields, check that the SYNCBSY bit in the Status register (QSPI_SR) is at 0 to ensure that no synchronization process is ongoing.

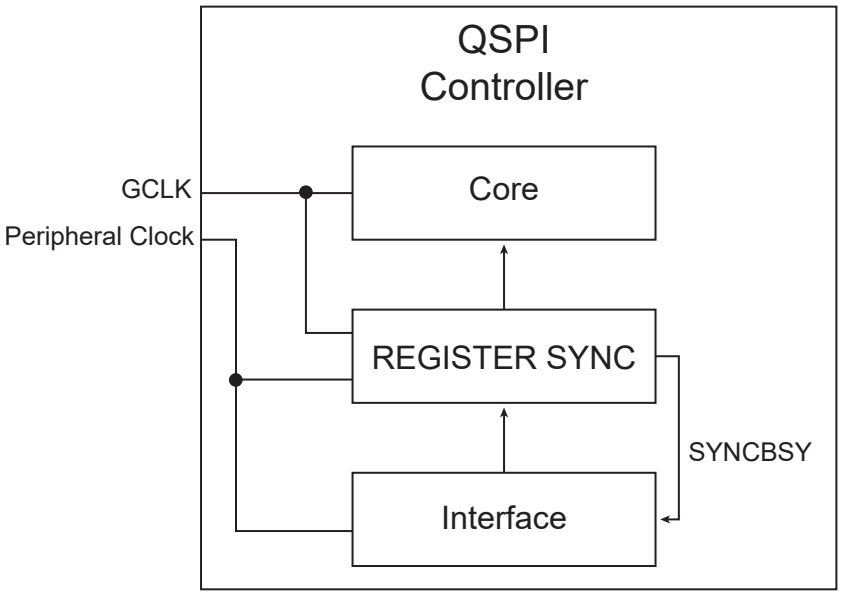
When synchronization is in progress, SYNCBSY is at 1.

As long as SYNCBSY is at 1, no access to the registers requiring synchronization is allowed.

When using UPDCFG in the Control register (QSPI_CR) to update the system configuration, SYNCBSY must be at 0 before and after writing UPDCFG. This ensures that the update is completed before going on to the next step.

Note: If the TDRE or RDRF flag is checked, checking QSPI_SR.SYNCBSY for QSPI_RDR.RD and QSPI_TDR.TD is not necessary. See Figure 64.13, Figure 64.17 and Figure 64.23 for detailed procedures.

Figure 64.2. Register Synchronization with QSPI Controller Core



64.6.2. Updating the QSPI Configuration

Once written in the registers, the configuration must be synchronized with the QSPI core.

At any time, the QSPI Controller core configuration can be updated by writing the QSPI_CR.UPDCFG bit to 1. This will update the QSPI core with the current register configuration. 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 Mode Register(1)
- QSPI Serial Clock Register
- QSPI Instruction Address Register
- QSPI Write Instruction Code Register
- QSPI Read Instruction Code Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register
- QSPI Refresh Register
- QSPI Write Access Counter Register

- QSPI Instruction Frame Register

Note:

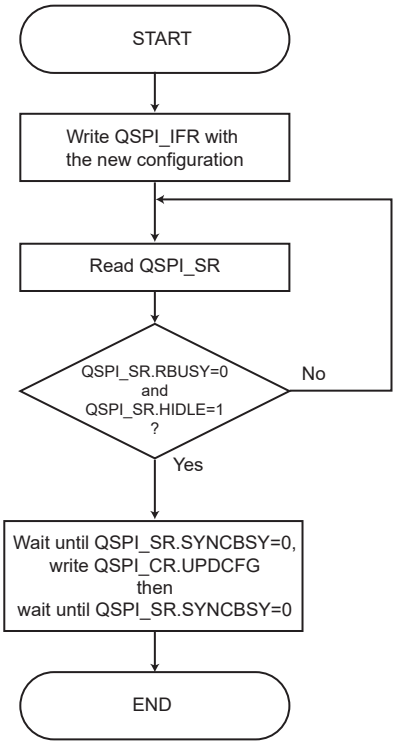
1. TAMPCLR in the Mode register (QSPI_MR) does not require synchronization with the QSPI core.

64.6.2.1. Changing Frame Configuration in Serial Memory Mode (XiP)

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 SMRM and TFRTYP in the Instruction Frame register (QSPI_IFR).

To modify QSPI_IFR after the initial configuration, proceed as illustrated in the figure below to ensure no frame loss. For details, see Instruction Frame and Table 64.3.

Figure 64.3. Changing QSPI_IFR



64.6.3. Serial Clock Phase and Polarity

Only Mode 0 is supported with QSPI_MR.SMM=1.

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). QSPI_SCR.CPHA programs the clock phase. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of these 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.

The table below shows the four modes and the corresponding parameter settings.

Table 64.2. QSPI Bus Clock Modes

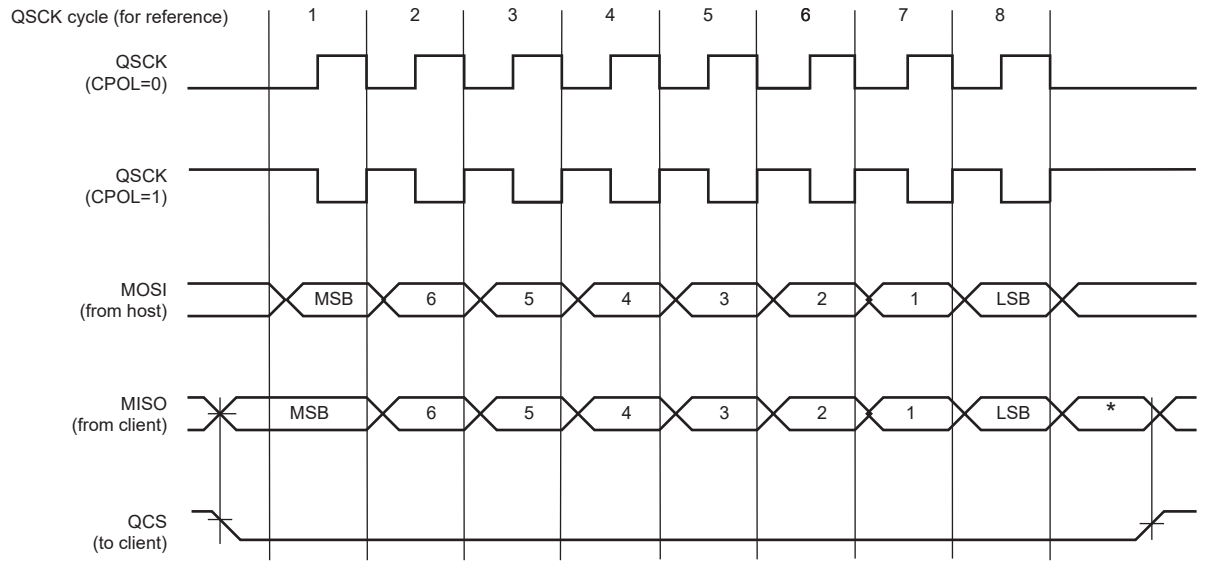
QSPI Clock Mode	QSPI_SCR.CPOL	QSPI_SCR.CPHA	Shift QSCK Edge	Capture QSCK Edge	QSCK Inactive Level
0	0	0	Falling	Falling	Low
1	0	1	Rising	Rising	Low
2	1	0	Rising	Rising	High

Table 64.2. QSPI Bus Clock Modes (continued)

QSPI Clock Mode	QSPI_SCR.CPOL	QSPI_SCR.CPHA	Shift QSCK Edge	Capture QSCK Edge	QSCK Inactive Level
3	1	1	Falling	Falling	High

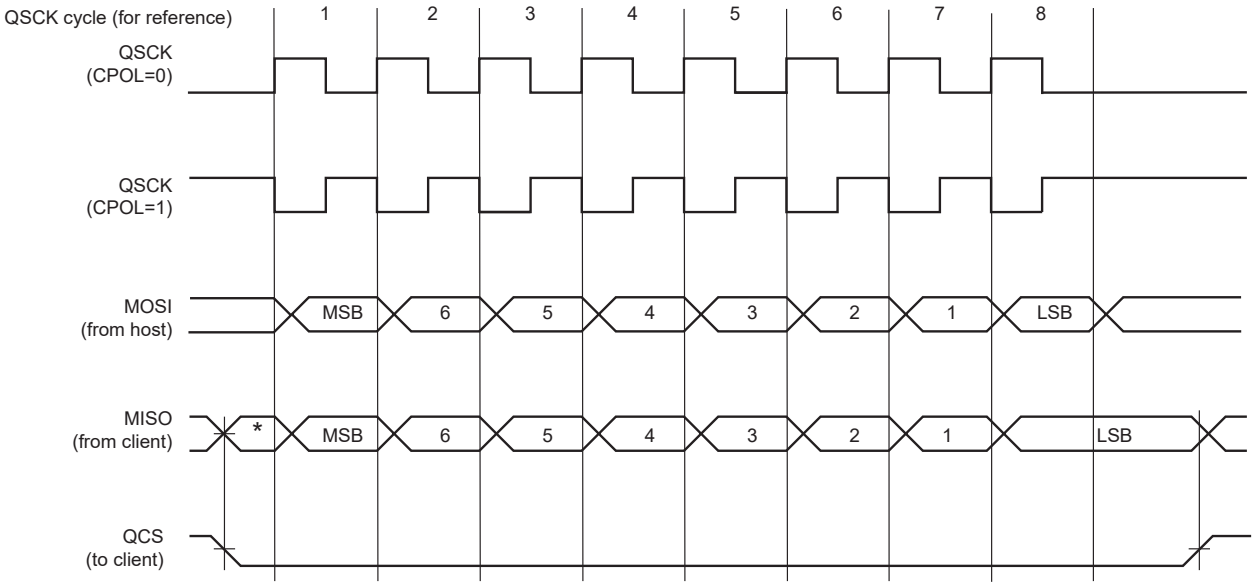
The figures below show examples of data transfers.

Figure 64.4. QSPI Transfer Format (QSPI_SCR.CPHA=0, 8 bits per transfer)



* Not defined, but normally the MSB of the previous character is received.

Figure 64.5. QSPI Transfer Format (QSPI_SCR.CPHA=1, 8 bits per transfer)



* Not defined, but normally the LSB of the previous character is transmitted.

64.6.4. Transfer Delays

Figure 64.6 and Figure 64.7 show 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—to adjust the minimum time of QCS at high level.
- The delay before QSCK, programmed by writing QSPI_SCR.DLYBS—to start delaying QSCK after the chip select has been asserted.
- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT.
  - QSPI_MR.SMM=0, to insert a delay between two consecutive transfers
  - QSPI_MR.SMM=1, to insert a delay between the last QSCK pulse and the QCS rise.
- The delay between consecutive transfers when SMM=1, programmed by writing QSPI_SCR.DLYBCT.

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)

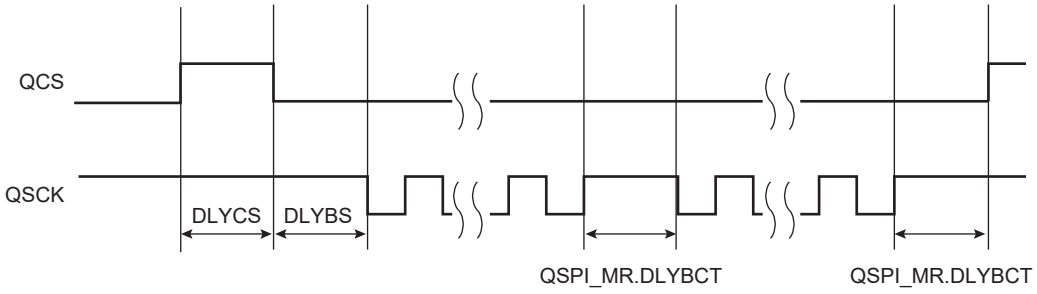
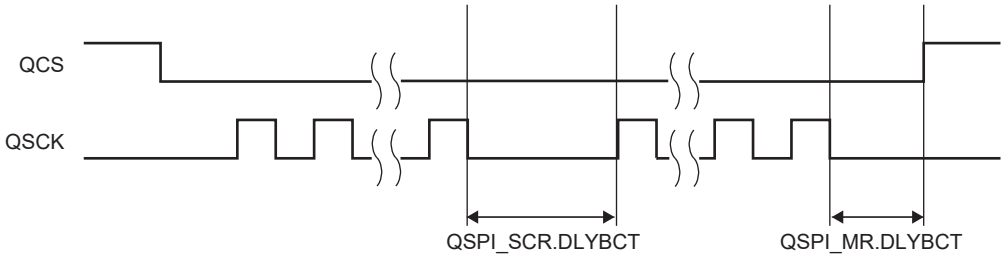


Figure 64.7. DLYBCT with SMM=1



64.6.5. 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.

64.6.6. Refresh Sequence

The QSPI uses some analog blocks:

- A DQS delay line

Once calibrated at first start of the QSPI, the product properties (power, voltage, temperature, etc.) continue to change over time, so the analog blocks used by the QSPI require an updated calibration.

The QSPI refresh sequence allows the QSPI to stop transfers on a regular basis to perform a calibration update. A 1 ms refresh interval is a safe value for Refresh register (QSPI_REFRESH).

The following analog blocks are refreshed during the refresh sequence:

- DQS delay

64.6.6.1. 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 (QSPI_CR.QSPIEN is written to 1).
- QSPI_REFRESH.REFRESH is not set to 0.
- QSPI_MR.DQSDLYEN is set.

An automatic refresh sequence of analog blocks is performed:

- after the first QSPI enable,
- when the internal refresh counter is set to 0 if REFRESH is not set to 0.

64.6.6.2. On-Demand Refresh

A refresh sequence can be launched at 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.

64.6.7. 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.

64.6.7.1. SPI Mode Operations

The QSPI in standard SPI mode operates on the GCLK 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).

Note: GCLK must be set to 2 x 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 Interrupt 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 GCLK clock can be switched off at this time (after the current frame has ended). See Deactivation Procedure for a detailed suspend procedure.

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.

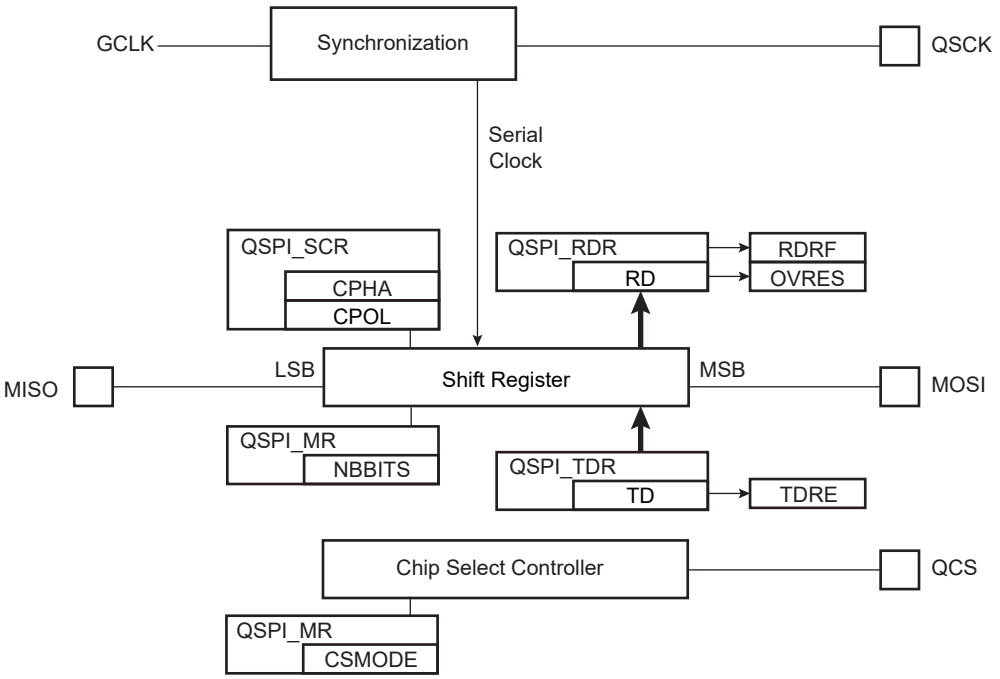
If QSPI_RDR has not been read before new data is received, the Overrun Error Status (OVRES) bit in QSPI_ISR is set. As long as this flag is set, new data will overwrite the old QSPI_RDR.RD value. The user must read QSPI_ISR to clear the OVRES bit.

The figure below shows a block diagram of the SPI when operating in Host mode. Figure 64.9 shows a flow chart describing how transfers are handled.

64.6.7.2. Legacy SPI Mode Block Diagram

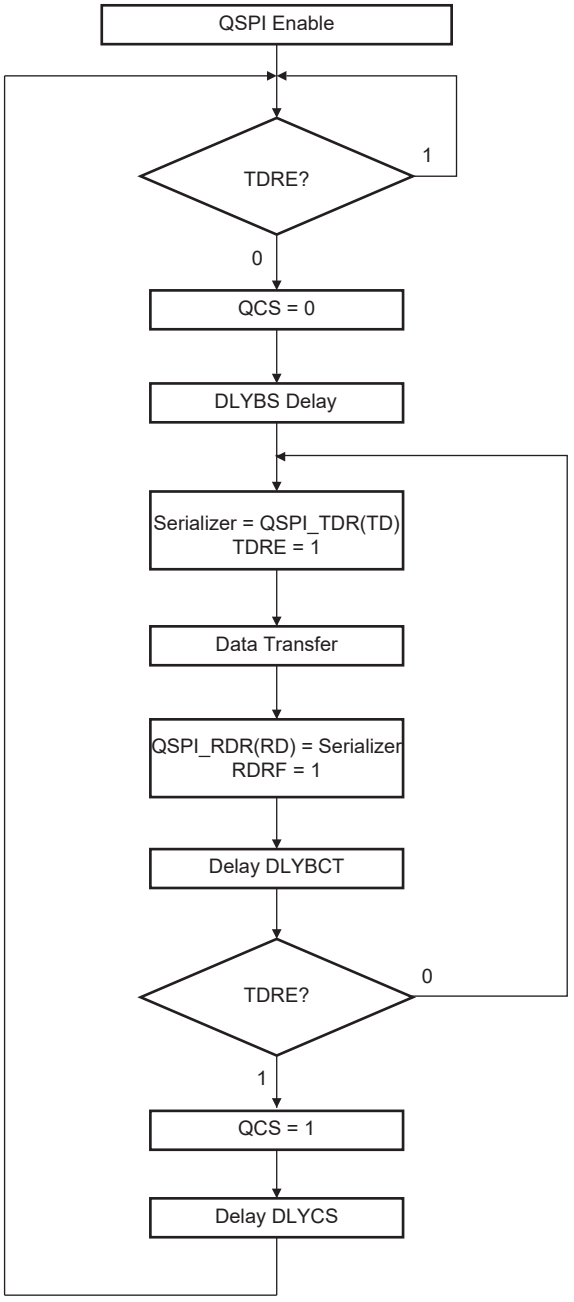
See QSPI SPI Mode.

Figure 64.8. Legacy SPI Mode Block Diagram



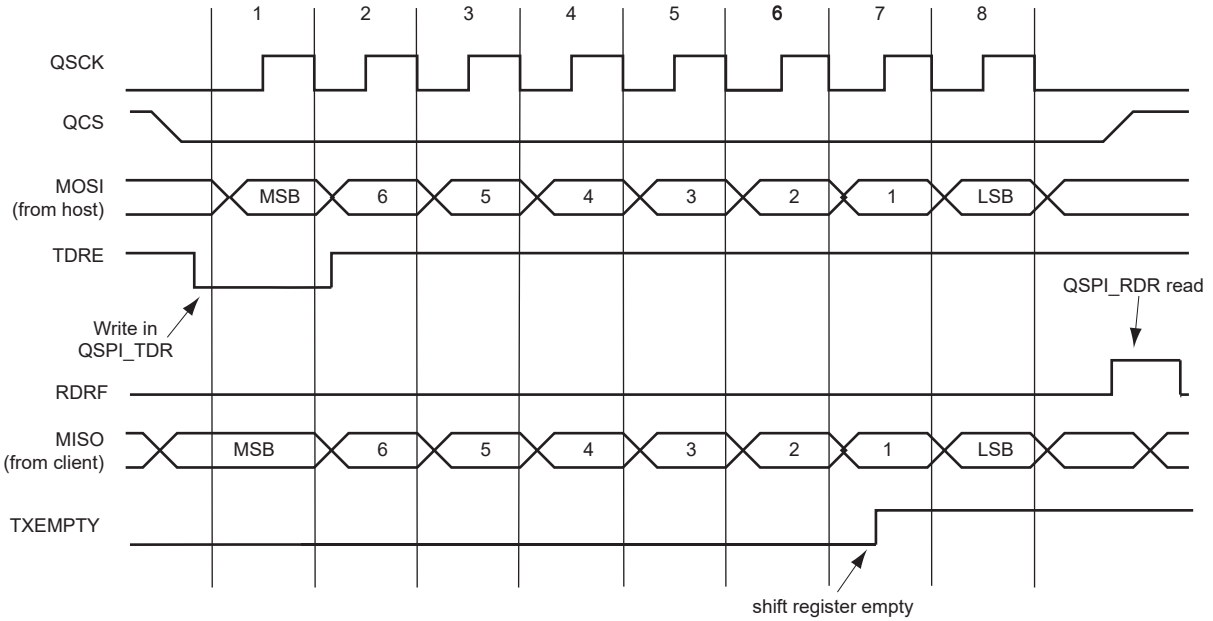
64.6.7.3.SPI Mode Flow Diagram

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_ISR during an 8-bit data transfer, 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

64.6.7.4.Chip Select Management 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_ISR 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.

64.6.7.5.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 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 QSPI_TDR by the DMA as well. This means that 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, on some SPI client devices, the communication might get lost. It may be necessary to configure CSMODE to 1.

64.6.8. 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, or 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 reading in the QSPI memory space depending on QSPI_IFR.TFRTYP and QSPI_IFR.SMRM configuration.

64.6.8.1. Initialization Procedure

The QSPI initialization procedure must follow the specific steps below to ensure proper behavior.

1. Enable the GCLK clock.
2. Configure QSPI_REFRESH, QSPI_MR, QSPI_SCR, QSPI_SMR, etc.
3. Wait for QSPI_SR.SYNCBSY=0, then write QSPI_CR.UPDCFG. Wait for QSPI_SR.SYNCBSY=0.
4. Write QSPI_CR.QSPIEN.
5. Wait for QSPI_SR.QSPIENS=1.
6. If automatic refresh is enabled, wait for QSPI_ISR.RFRSHD=1 by polling or interrupt (see Automatic Refresh).

64.6.8.2. Deactivation Procedure

The QSPI deactivation procedure is as follows.

1. Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
2. Send a QSPI_CR.LASTXFER command.
3. If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA=1.
4. Set QSPI_CR.QSPIDIS and wait for QSPI_SR.QSPIENS=0.
5. Disable the GCLK clock.
6. Disable the peripheral clock.

64.6.8.3. GCLK Frequency Change Procedure

The GCLK frequency change procedure is as follows.

1. Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
2. Send a QSPI_CR.LASTXFER command.
3. If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA=1.
4. Set QSPI_CR.QSPIDIS and wait for QSPI_SR.QSPIENS=0.
5. Disable the GCLK clock.
6. If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA=1.
7. Change GCLK frequency.
8. Write QSPI_CR.QSPIEN.
9. Wait for QSPI_SR.QSPIENS=1.
10. If automatic refresh is enabled, wait for QSPI_ISR.RFRSHD=1 by polling or interrupt (see Automatic Refresh).

64.6.8.4. End of Frame Procedure

If the QSPI needs reconfiguring, first apply the following procedure to close the current frame safely:

1. Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
2. Send a QSPI_CR.LASTXFER command.
3. If QSPI_SR.CSS=0 or QSPI_SR.CSS1=0, wait for QSPI_ISR.CSRA=1.
4. Set QSPI_CR.QSPIDIS and wait for QSPI_SR.QSPIENS=0.

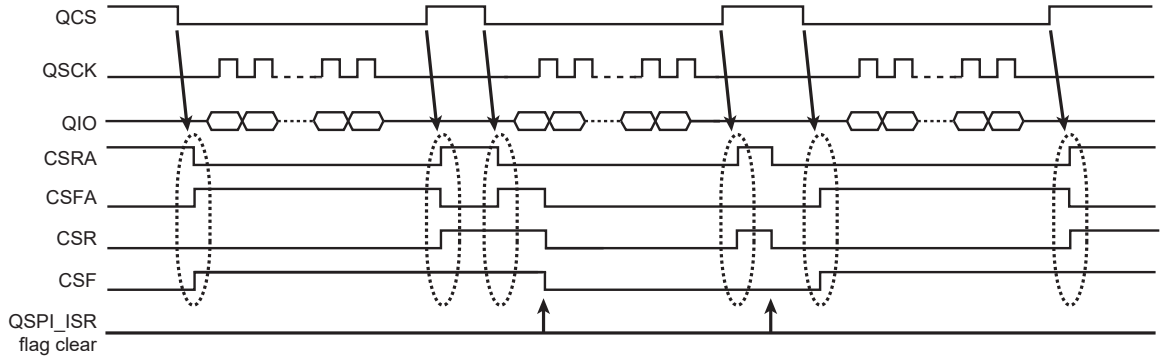
Note: Consider using a memory barrier when writing the application code before executing the above procedure.

64.6.8.5.Device Selection Flags

To control the QSPI frames, the following flags showing the state of the device selection are available in QSPI_ISR:

- CSR and CSRA indicate when a rising edge of QCS has been detected. CSRA is automatically cleared when a QCS falling edge is detected.
- CSF and CSFA indicate when a falling edge of QCS has been detected. CSFA is automatically cleared when a QCS rising edge is detected.

Figure 64.11. Device Selection Flags



64.6.8.6.Instruction Frame

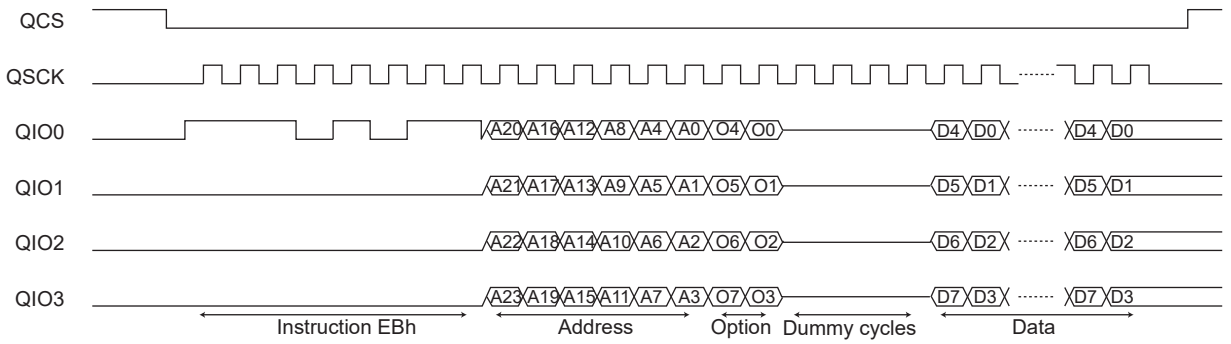
In order to control serial Flash memories, the QSPI can send instructions via the SPI bus (READ, PROGRAM, ERASE, LOCK, etc.). The QSPI includes a complete Instruction Frame register (QSPI_IFR) to ensure compatibility with all serial Flash memories.

An instruction frame includes:

- (Optional) An instruction code (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.
- An option code (size: 1/2/4/8 bits). The option code can be used to activate some memory features.
- Dummy cycles. Dummy are required by some instructions.
- Data bytes. 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 the Single-bit SPI, Dual SPI, Quad SPI or Octal SPI protocol.

Figure 64.12. Instruction Frame



64.6.8.7.Instruction Frame Transmission

If the instruction frame includes the instruction code and/or the option code, the user must configure 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. For a frame without data (QSPI_IFR.DATAEN=0), QSPI_WICR is used for instruction and option codes.

QSPI_IFR must be configured with the instruction frame to send.

The instruction frame is configured by the following QSPI_IFR bits and fields:

- WIDTH field—configures which data lanes are used to send the instruction code, the address, the option code and to transfer the data.
- INSTEN bit—enables an instruction code.
- ADDREN bit—enables an address after the instruction code.
- OPTEN bit—enables an option code after the address.
- DATAEN bit—enables the transfer of data (READ or PROGRAM instruction).
- OPTL field—configures 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—configures the QSPI address field size.
- TFRTYP field—defines the type of memory access. See the table below.
- CRM bit—enables Continuous Read mode, see Continuous Read Mode.
- DDREN bit—configures the Double Data Rate mode; the instruction code is still transmitted in Single Data Rate mode. The instruction code can be transmitted in DDR mode by writing a 1 to QSPI_IFR.DDRCMDEN.
- NBDUM field—configures 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 required by the serial Flash memory.
- END bit—defines the endianness of the targeted memory.
- SMRM bit—when TFRTYP=0, defines if the instruction frame transmission is triggered by register accesses or QSPI memory space accesses.
- APBTFRTYP bit—defines the peripheral bus register transfer to memory type (read or write) when QSPI_IFR.TFRTYP is written to 0.
- DQSEN bit—defines if the targeted memory supplies a DQS signal.

- DDRCMDEN bit—defines if the instruction code must be sent in DDR mode when QSPI_IFR.DDREN bit is written to 1.
- HFWBEN field—enables the HyperFlash Write Buffer command support. In this mode, a new command is generated for each write access. See HyperFlash Mode.
- PROTTYP bit—defines the QSPI protocol type.

See Instruction Frame register.

Depending on TFRTYP and SMRM memory accesses, the applicable methods are as follows.

Table 64.3. Memory Access Methods

QSPI_IFR Configuration	Memory Accesses via the System Bus	Memory Accesses via the Peripheral Bus
TFRTYP=0 and SMRM=1	No	Yes
TFRTYP=0 and SMRM=0	Yes	No
TFRTYP=1 or SMRM=0	Yes	No

64.6.8.7.1.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 (such as the WRITE ENABLE command) or if QSPI_IFR.SMRM is set to 1, the user must first configure the address to send by writing ADDR in the Instruction Address register (QSPI_IAR) (if 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 defines whether the access is a read or a write access. Write frames are triggered by writing QSPI_TDR, and read frames (or frames with no data such as WRITE_ENABLE) are triggered by setting QSPI_CR.STTFR. Each time a new transfer is issued, an SPI transfer is performed with a byte size or halfword size if the QSPI_IFR.WIDTH field is configured to either OCT_OUTPUT, OCT_IO or OCT_CMD and QSPI_IFR.DDREN=1. Reading QSPI_RDR triggers a new read access to the next sequential data in the memory[(1)]. QSPI_TDR can be written when the flag TDRE is set. The QSPI transfer ends by writing QSPI_CR.LASTXFER. See Figure 64.13 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.

  Note:

  a. If QSPI_RDR is read while the QPSI controller is busy receiving the data, the QSPI_ISR.RBSYERR error flag is set, indicating a possible data corruption.

64.6.8.7.2.Write Access Counter

The field NBWRA, in the Write Access Counter register (QSPI_WRACNT) 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. NBWRA is reset and begins counting after each start of the instruction frame. If

QSPI_IFR.PROTTYP=3 and QSPI_IFR.HFWBEN=1, NBWRA is reset after setting QSPI_IFR.HFWBEN bit to 1 and counts data on every frame.

64.6.8.7.3.Memory Array Access

To access the memory array, QSPI_IFR.TFRTYP must be set to 1. In the case of write access to the memory array, 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 by writing QSPI_TDR. Each time a new transfer trigger is issued, an SPI transfer is performed with a byte size or halfword size if the QSPI_IFR.WIDTH field is set 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 Figure 64.17 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 non-sequential (addresses are not consecutive), a new instruction frame may be sent (depending on optimization) with the last system bus access address. This way, the system can read/write data at a random location in the serial memory. 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 and QSPI_SR.HIDLE at 1 before terminating the frame. The last SPI transfer ends by writing QSPI_CR.LASTXFER. See Figure 64.19 and Figure 64.21 for details.

  The following figures illustrate instruction transmission management.

Figure 64.13. Instruction Transmission Flow Diagram SMRM=1 and TFRTYP=0 (Memory Register Access)

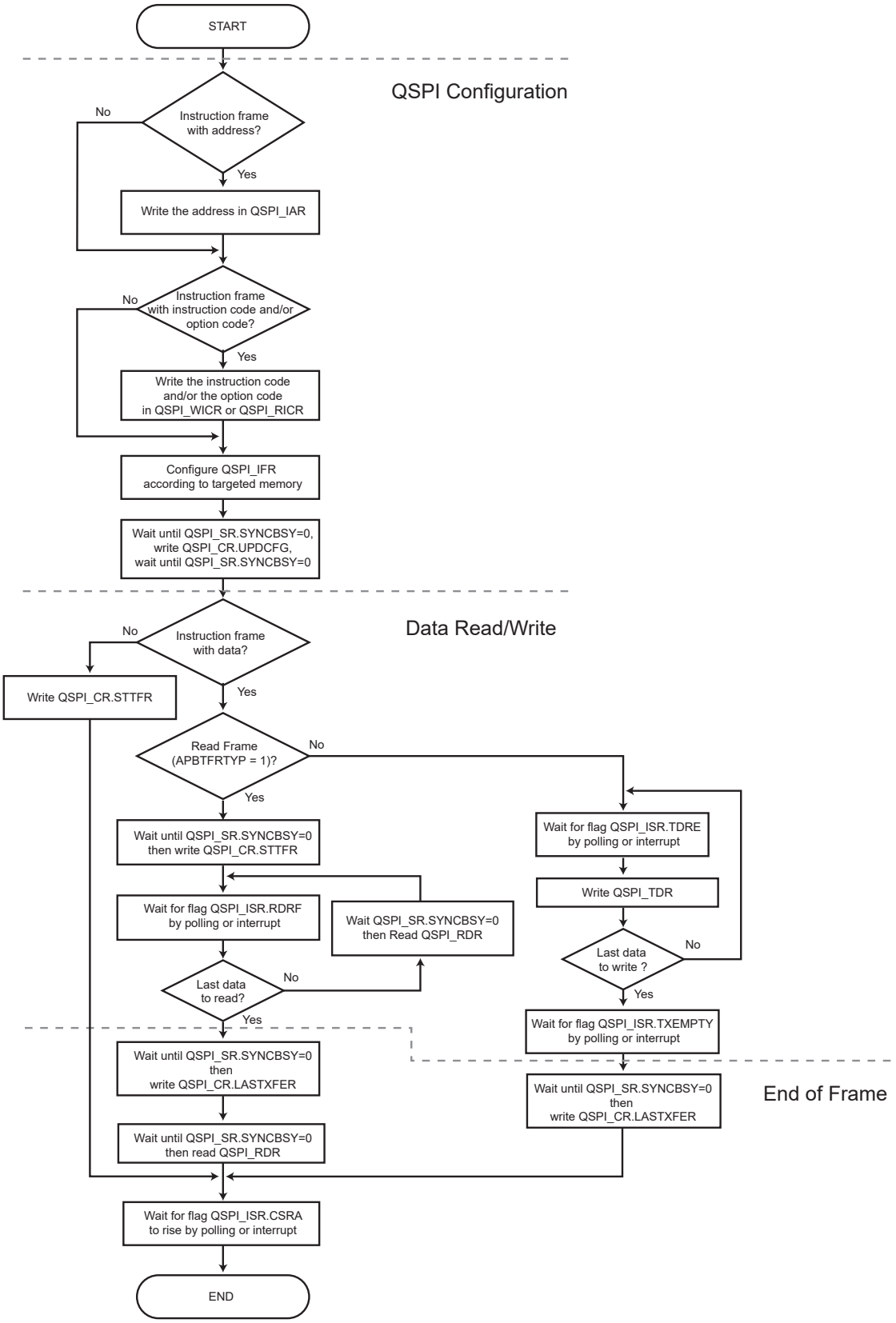


Figure 64.14. HyperBus Instruction Transmission Flow Diagram SMRM=1 and TFRTYP=0 (Memory Register Access)

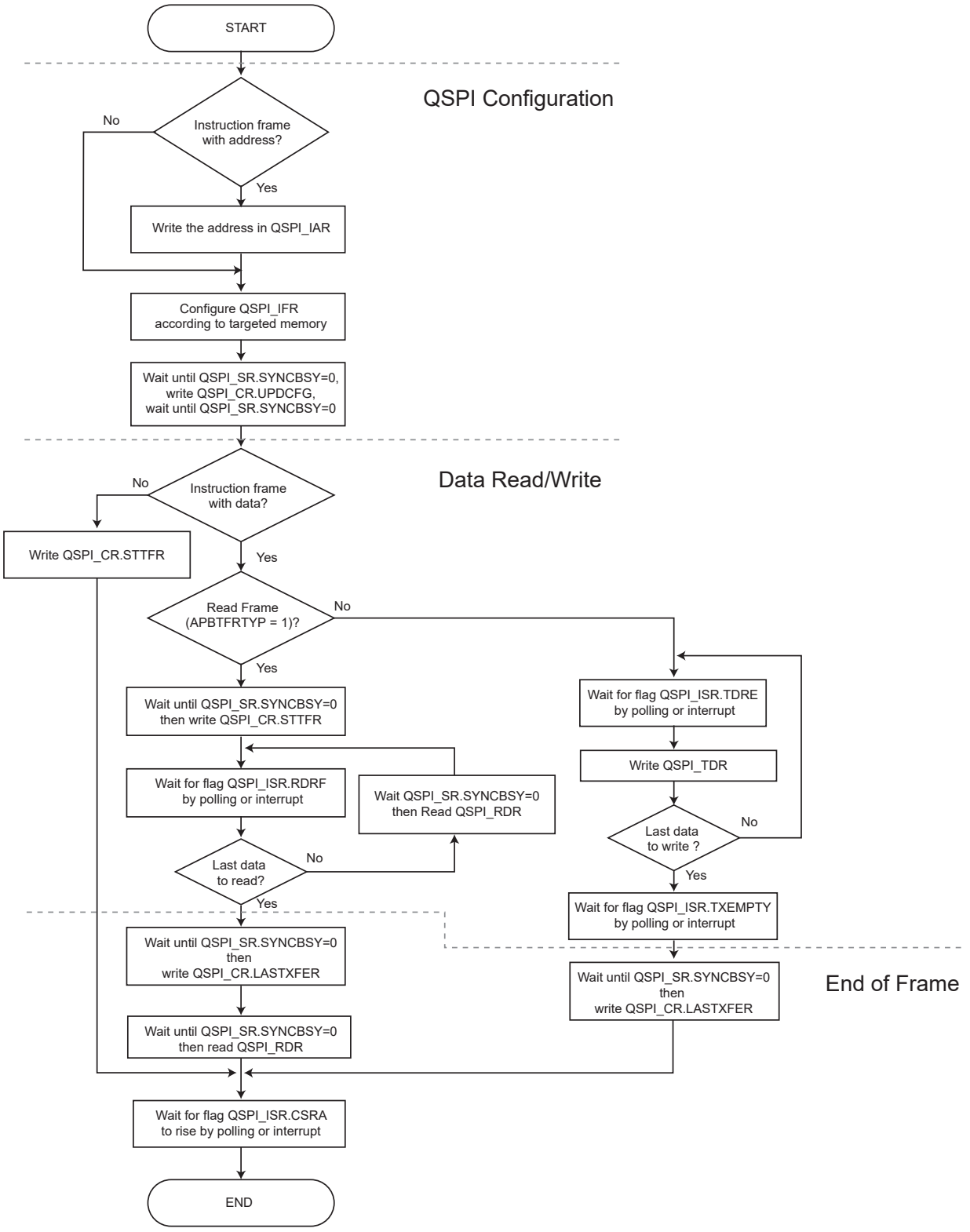


Figure 64.15. Instruction Transmission Flow Diagram SMRM=0 and TFRTYP=0 (Memory Register Access)

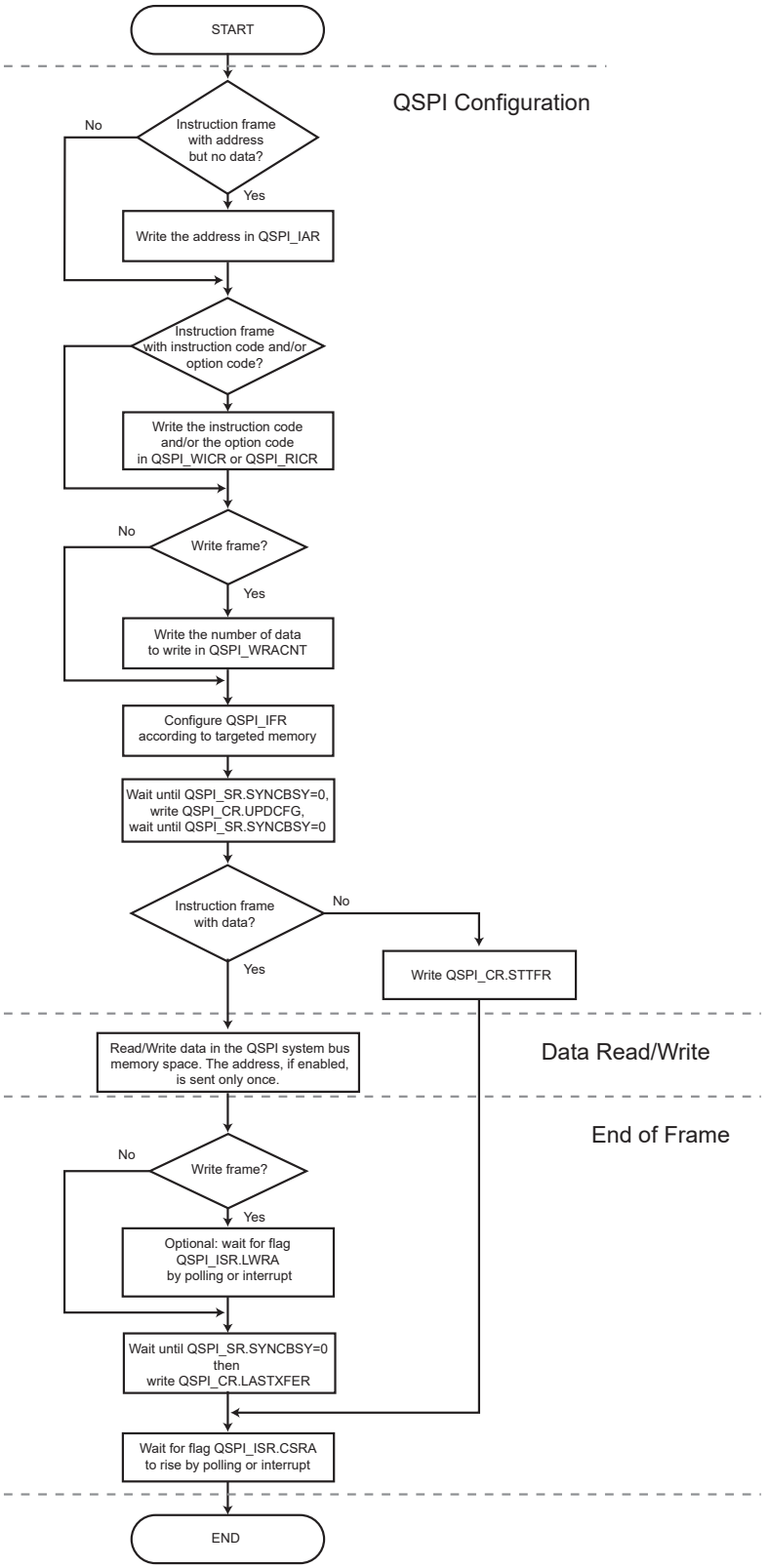


Figure 64.16. HyperBus Instruction Transmission Flow Diagram SMRM=0 and TFRTYP=0 (Memory Register Access)

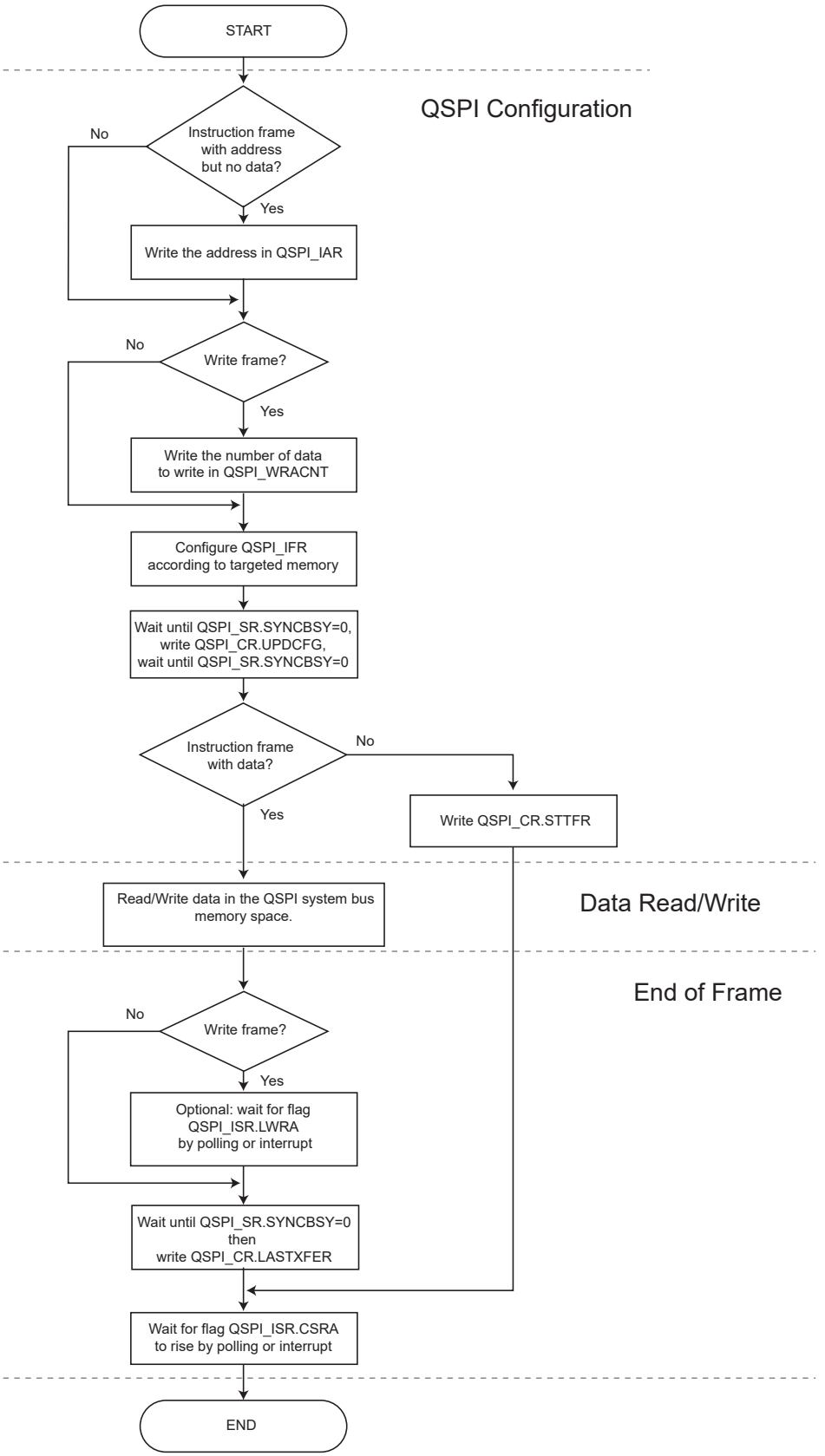


Figure 64.17. Instruction Transmission Flow Diagram SMRM=1 and TFRTYP=0 (Memory Write Access)

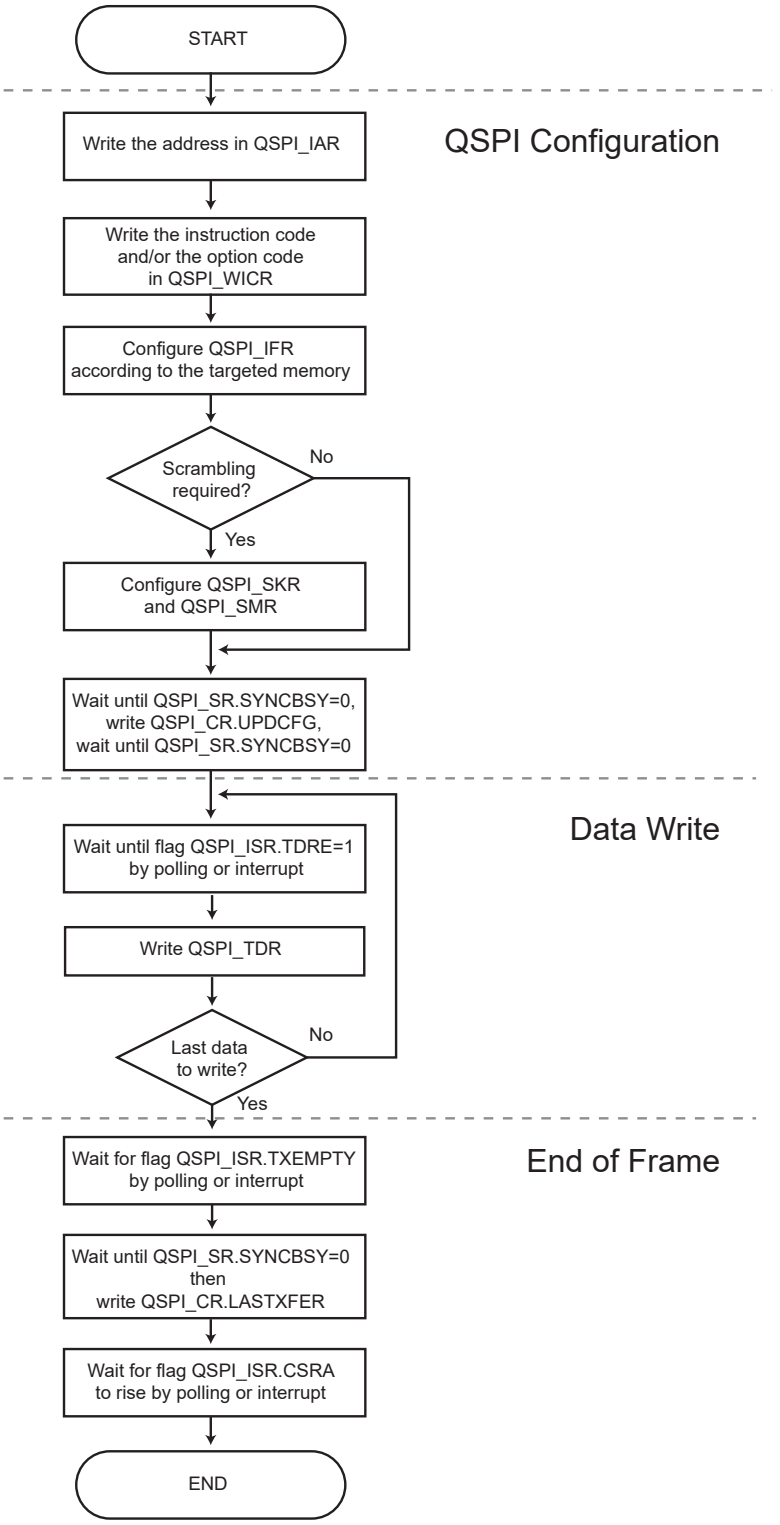


Figure 64.18. HyperBus Instruction Transmission Flow Diagram SMRM=1 and TFRTYP=0 (Memory Write Access)

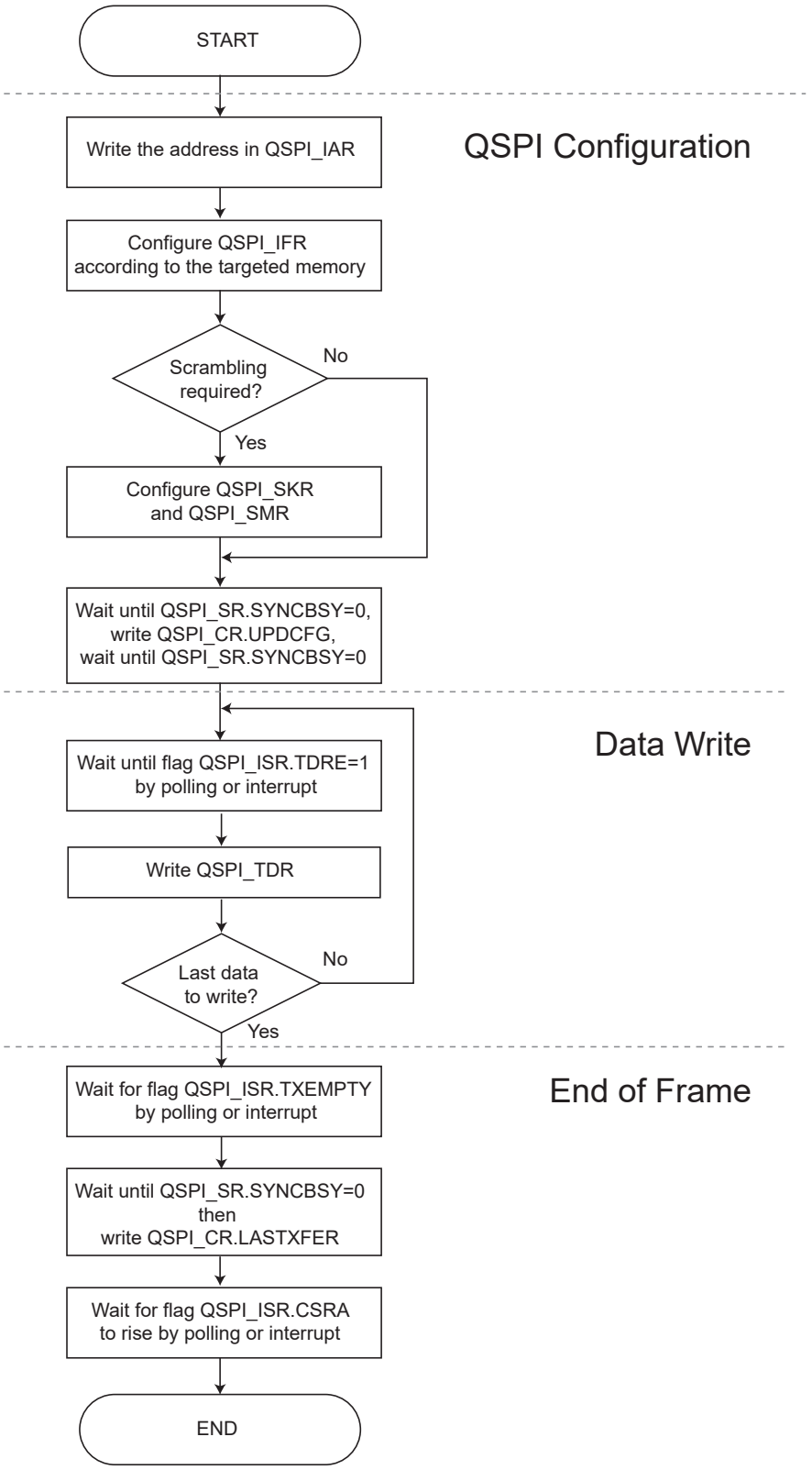


Figure 64.19. Instruction Transmission Flow Diagram TFRTYP=1 (Memory Write Access)

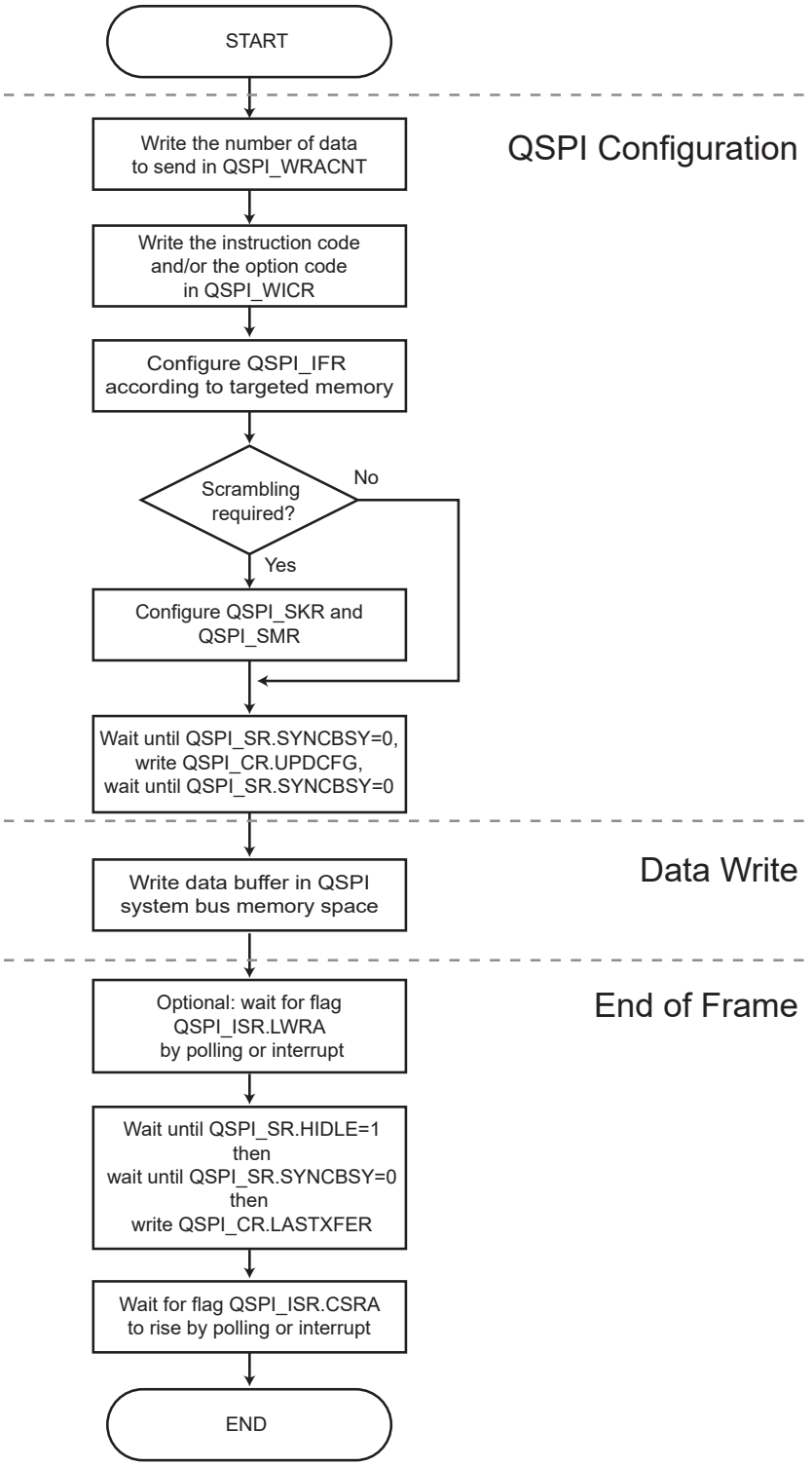


Figure 64.20. HyperBus Instruction Transmission Flow Diagram TFRTYP=1 (Memory Write Access)

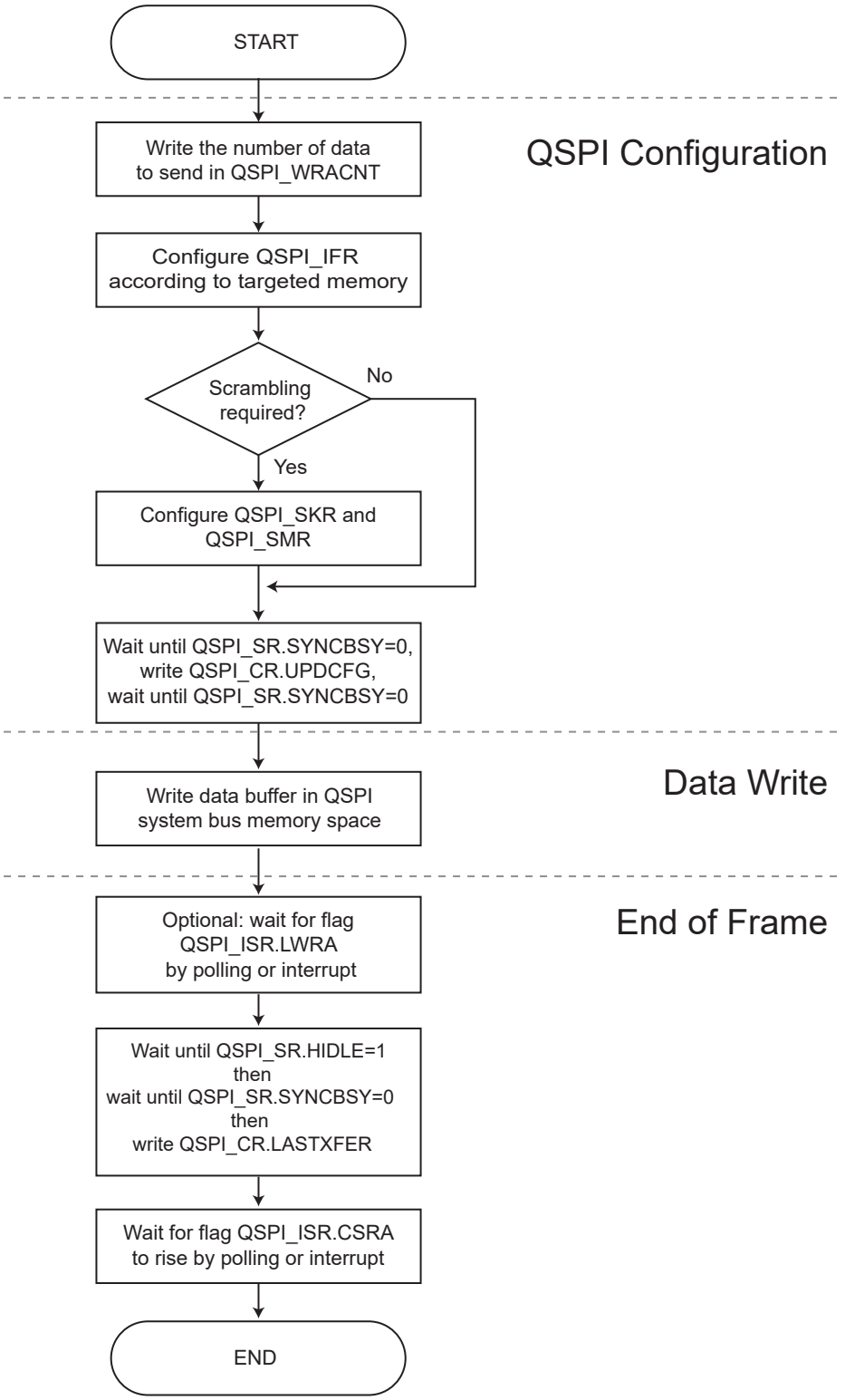


Figure 64.21. Instruction Transmission Flow Diagram TFRTYP=1 (Memory Read Access)

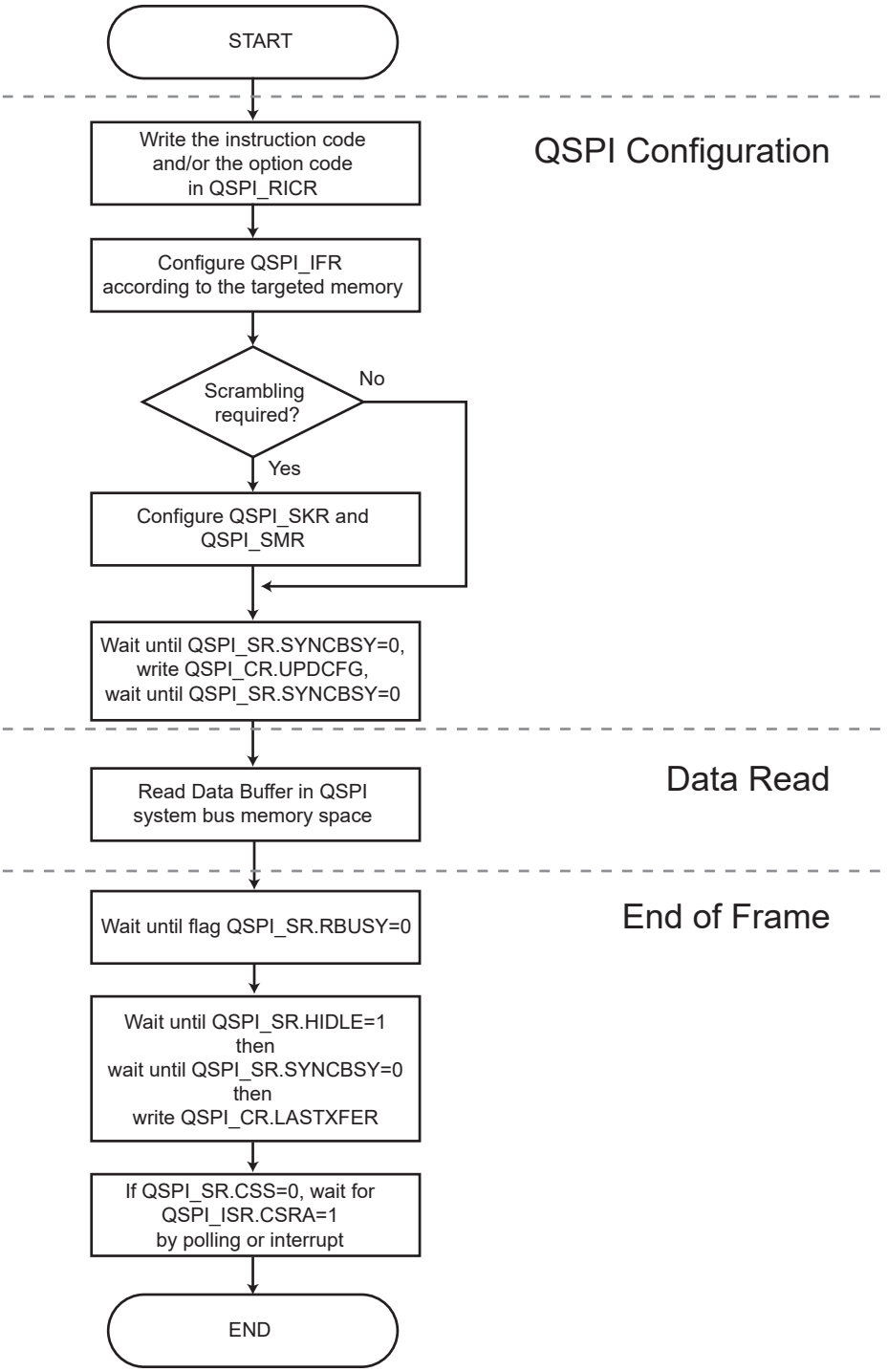


Figure 64.22. HyperBus Instruction Transmission Flow Diagram TFRTYP=1 (Memory Read Access)

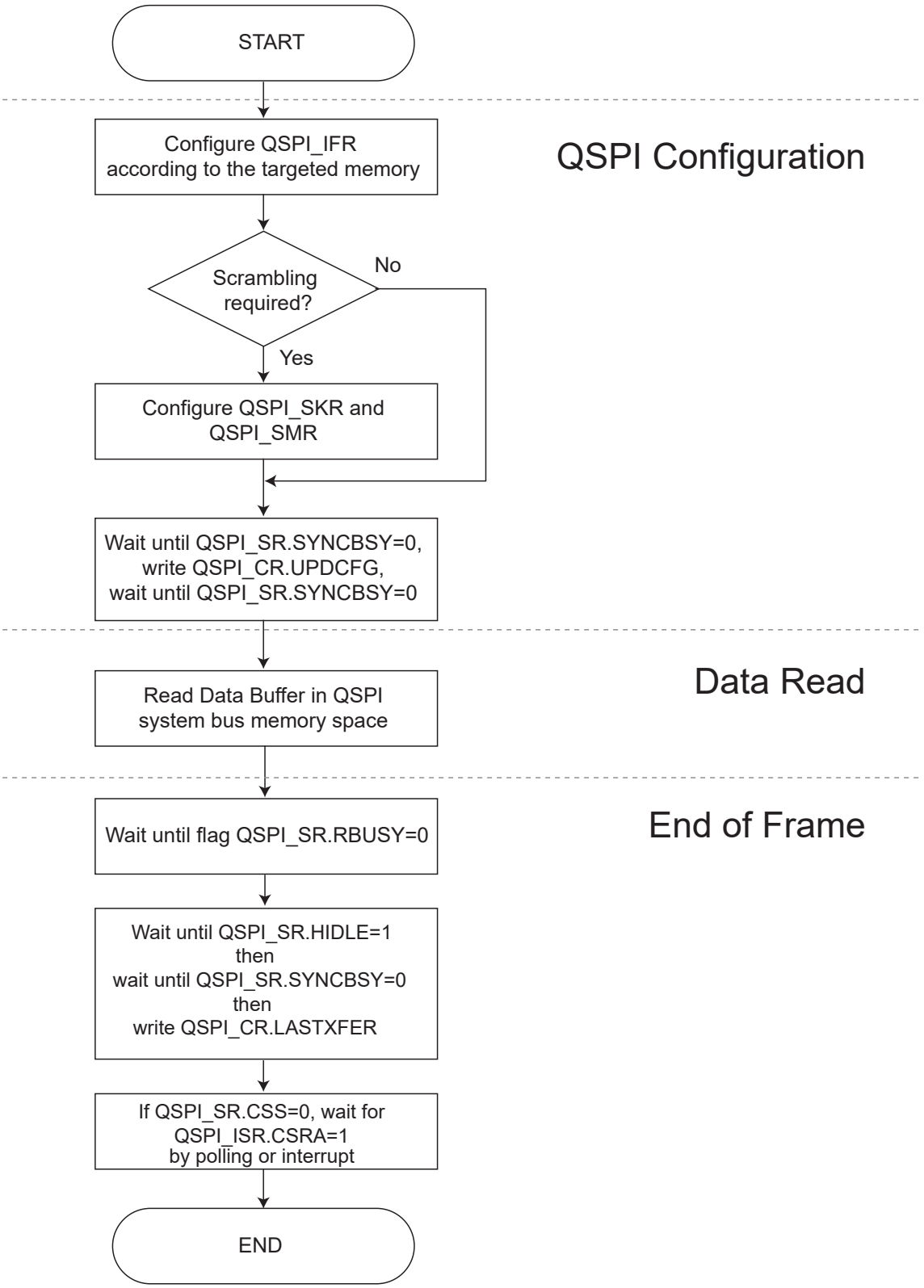


Figure 64.23. HyperFlash “Write Buffer” Flow Diagram SMRM=1 and TFRTYP=0 (Memory Write Access)

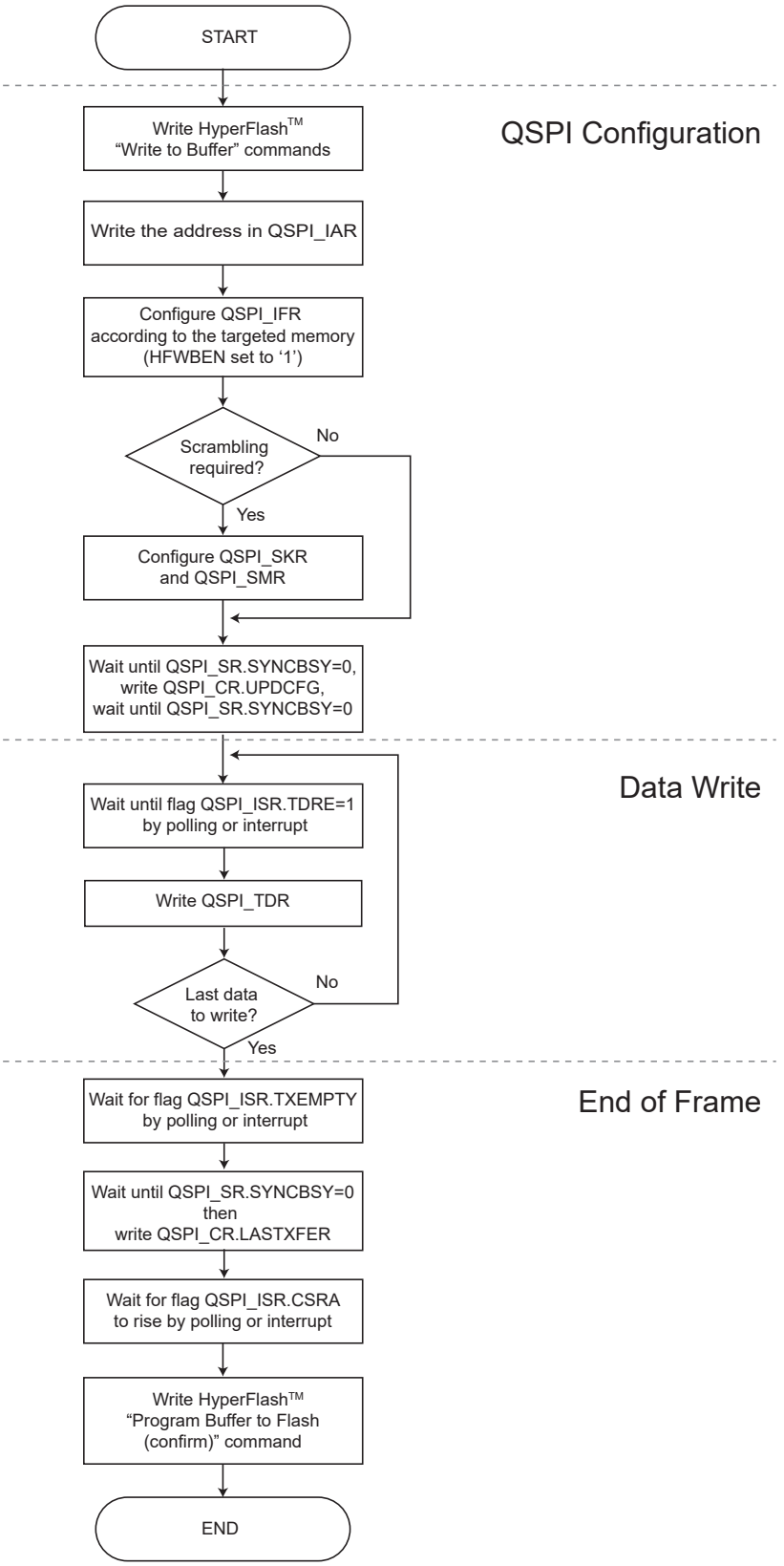
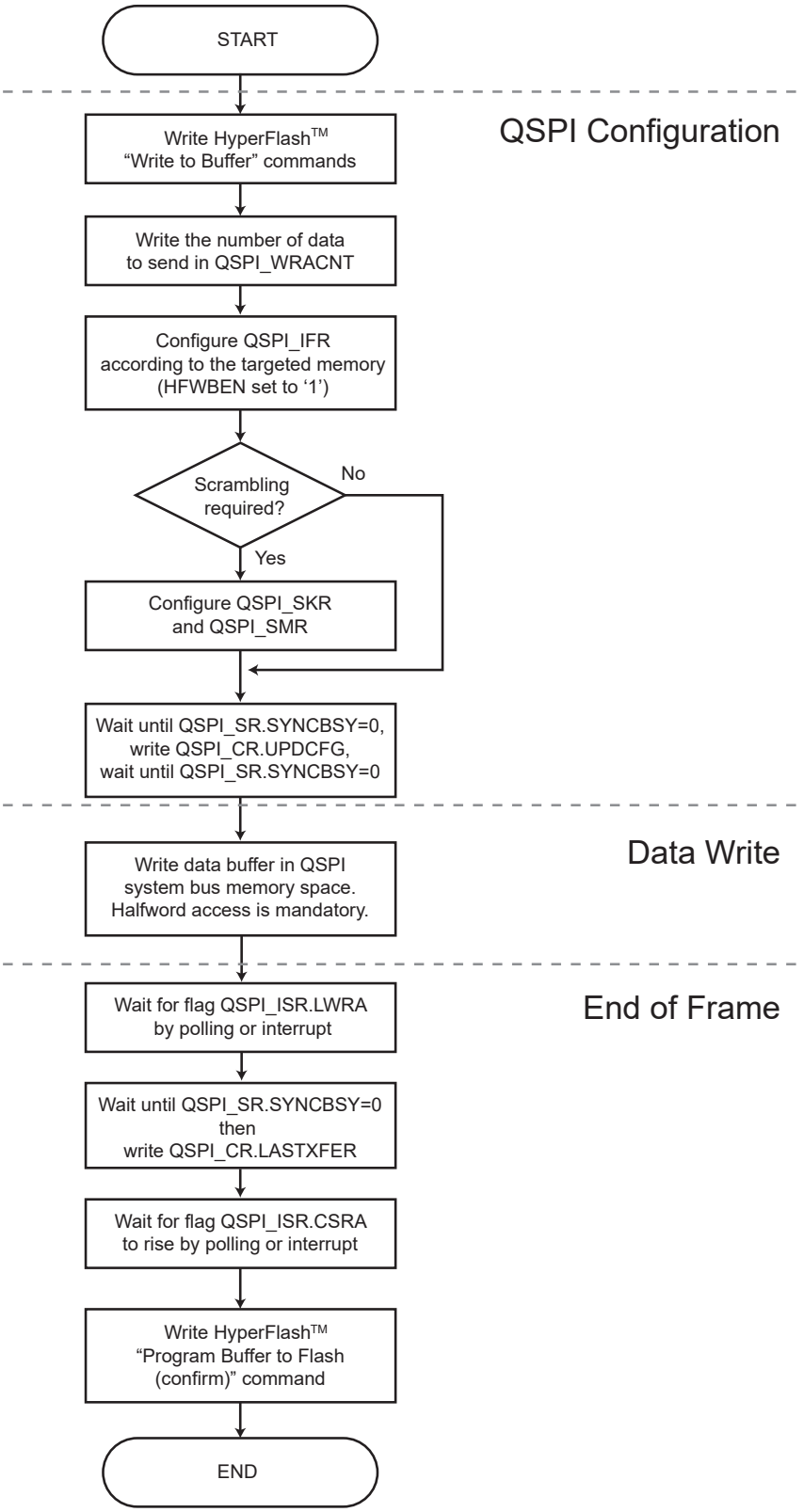


Figure 64.24. HyperFlash “Write Buffer” Flow Diagram TFRTYP=1 (Memory Write Access)



64.6.8.8.Write Memory Transfer

Many QSPI Flash memories impose a sequential write of their data buffer. When using this type of memory, the access to the memory must be performed in a strongly ordered way.

64.6.8.9.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 a random address into the serial Flash memory, allowing the CPU to execute code directly from it (XiP, or Execute in Place).

In order to read 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 matches the address of the data inside the serial Flash memory.

When the Read mode is used, several instruction frames can be sent before writing QSPI_CR.LASTXFR. Each time the system bus read accesses become non-sequential (addresses are not consecutive), a new instruction frame may be sent (depending on optimization) with the corresponding address.

64.6.8.10.Continuous Read Mode

The QSPI is compatible with the Continuous Read mode which is implemented in some serial Flash memories.

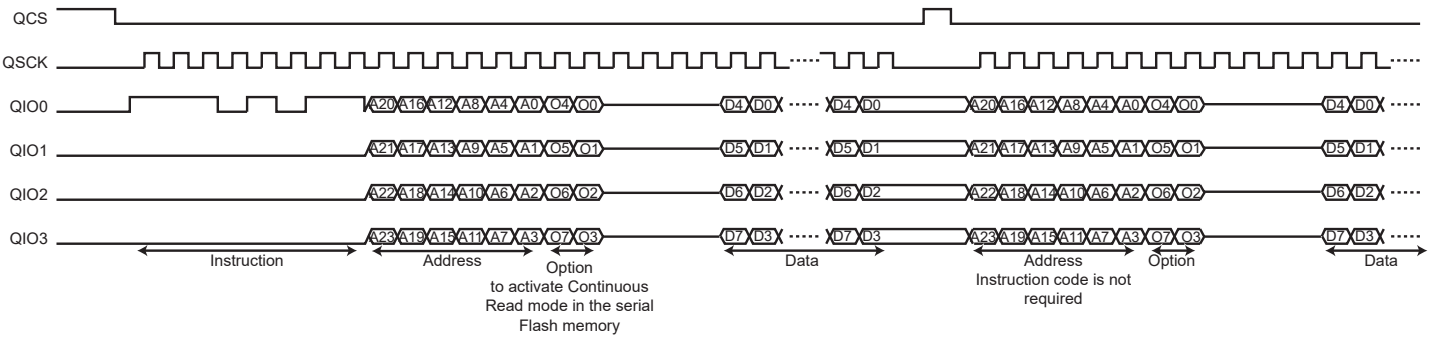
In Continuous Read mode, the instruction overhead is reduced by excluding the instruction code from the instruction frame. When the Continuous Read mode is activated in a serial Flash memory by a specific option code, the instruction code is stored in the memory. For the next instruction frames, the 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 may lead 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 set to 1). The Continuous Read mode is enabled in the serial Flash memory by sending a specific option code.

CAUTION Check the connected Memory Continuous read mode compatibility before enabling.

Figure 64.25. Continuous Read Mode



64.6.8.11.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).

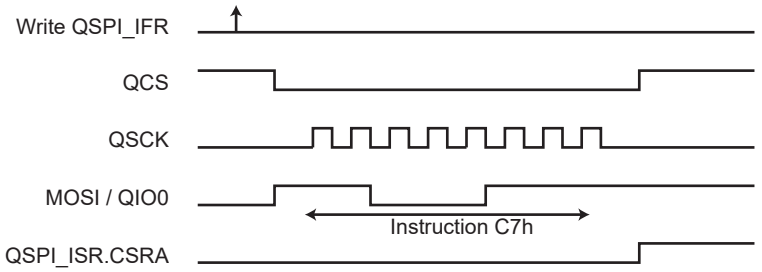
Example 1:

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 64.26. Instruction Transmission Waveform 1



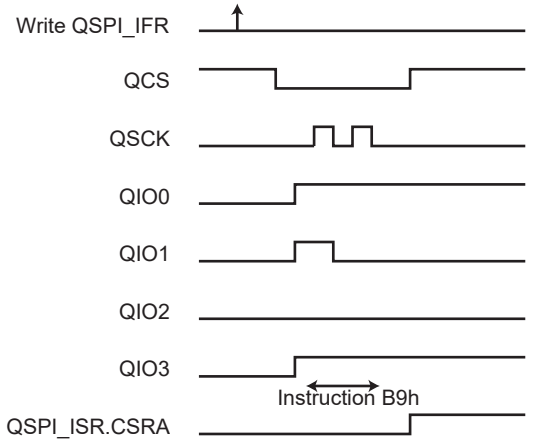
Example 2:

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 64.27. Instruction Transmission Waveform 2



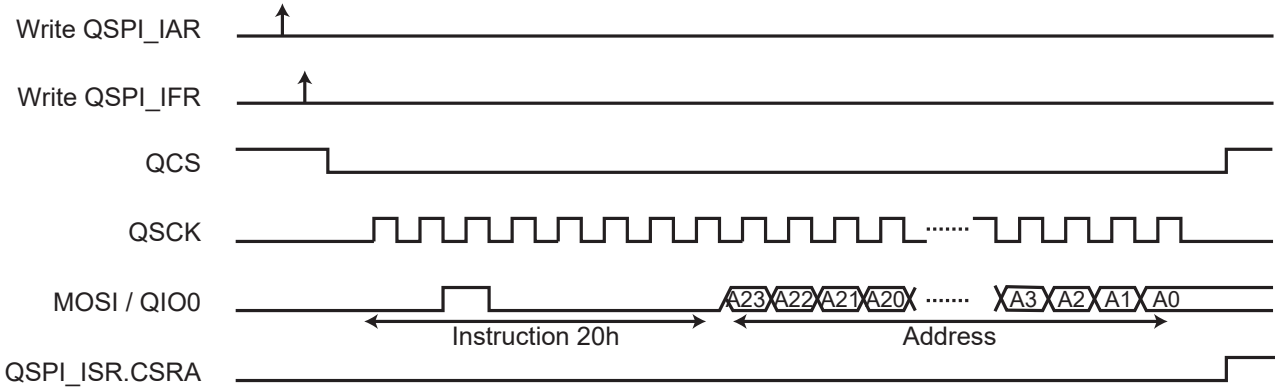
Example 3:

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 64.28. Instruction Transmission Waveform 3



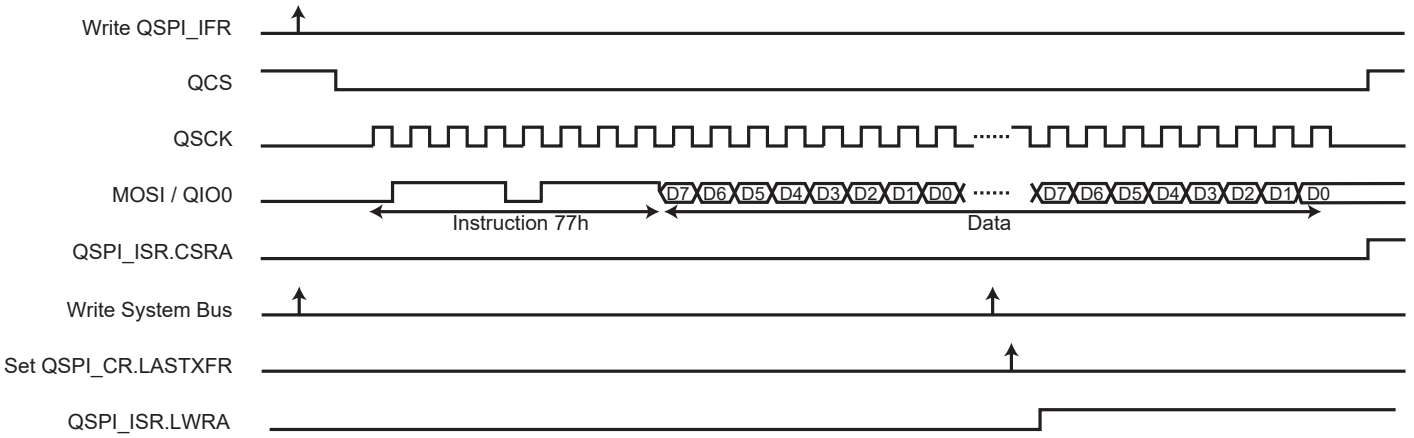
Example 4:

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 (0x60000000 - 0x80000000).
  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 64.29. Instruction Transmission Waveform 4



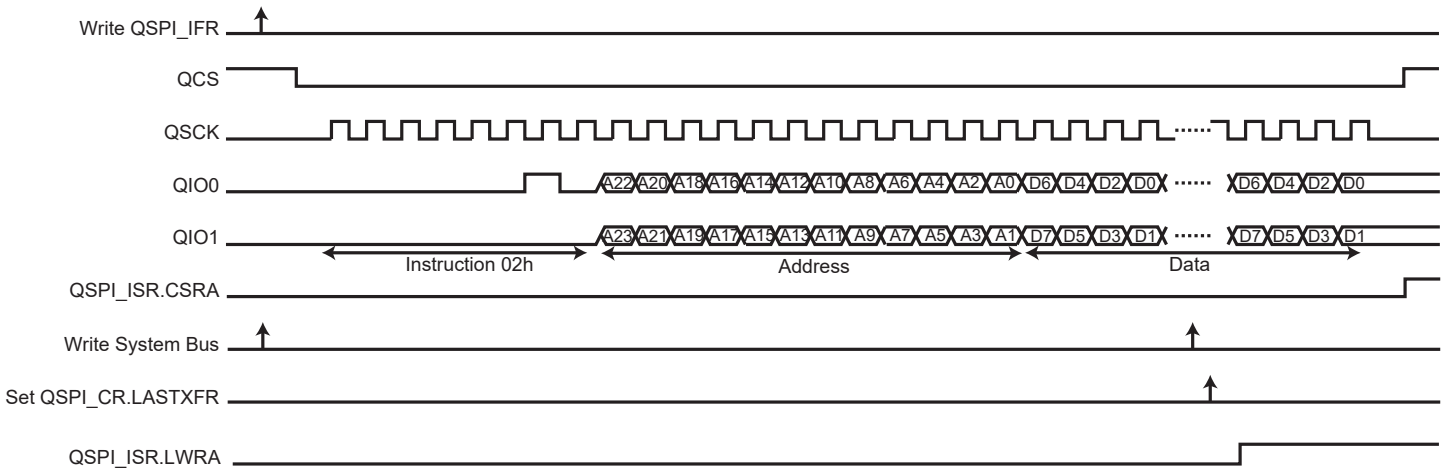
Example 5:

Instruction in Single-bit SPI, with 24-bit 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 (0x60000000 - 0x80000000).
  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 64.30. Instruction Transmission Waveform 5



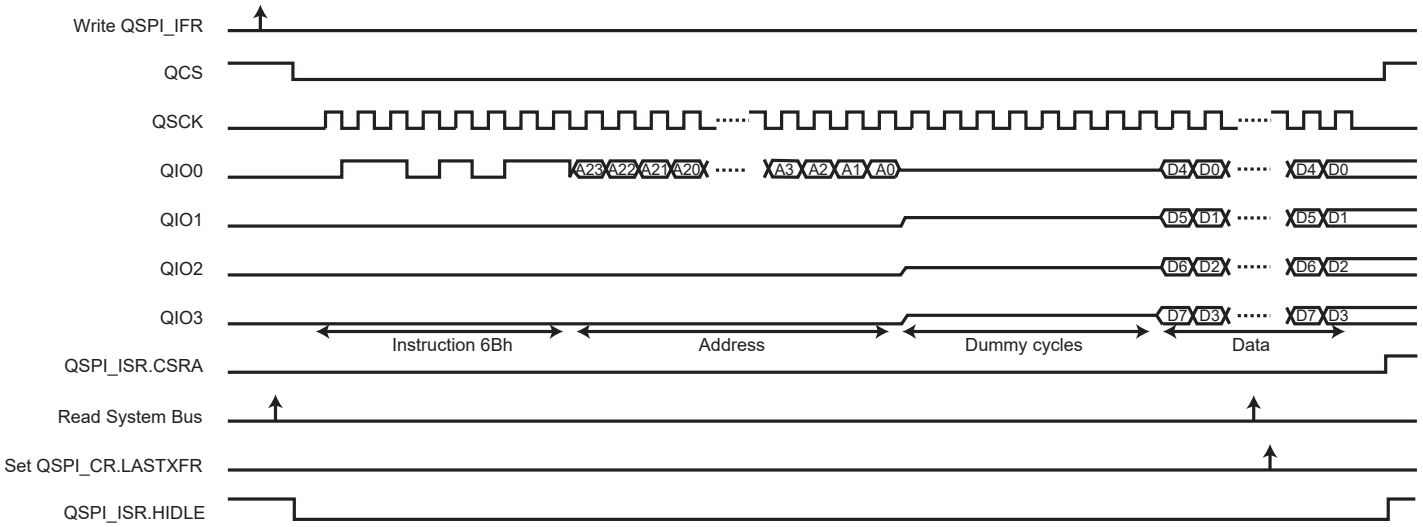
Example 6:

Instruction in Single-bit SPI, with 24-bit 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 (0x60000000 - 0x80000000).
  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=0 and QSPI_SR.HIDLE=1.
- Write a 1 to QSPI_CR.LASTXFR.
- If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA to rise.

Figure 64.31. Instruction Transmission Waveform 6



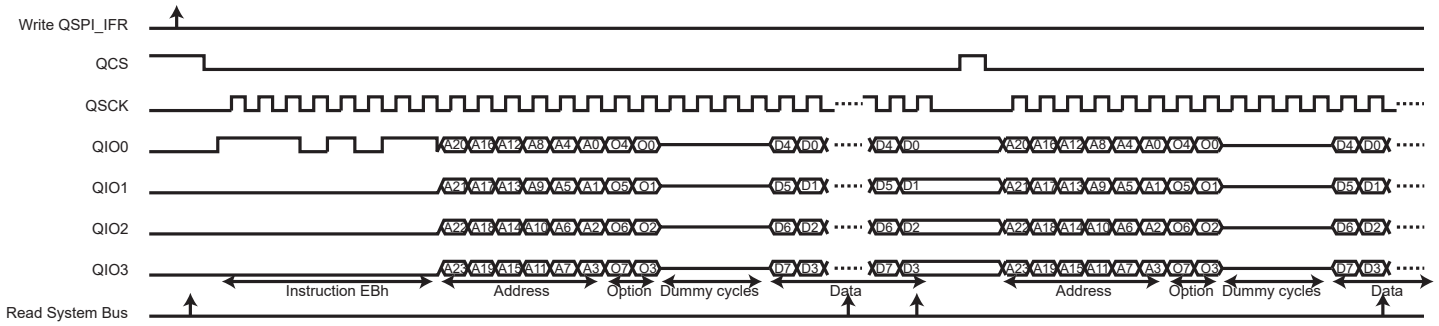
Example 7:

Instruction in Single-bit SPI, with 24-bit address and option in Quad SPI, with data read in Quad SPI, with four dummy cycles and continuous read.

Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB in QSPI_RICR.
- Write 0x0004_1BF4 in QSPI_IFR.
- Update configuration (see Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x60000000 - 0x80000000).
  The address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
- Write a 1 to QSPI_CR.LASTXFR.
- If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA to rise.

Figure 64.32. Instruction Transmission Waveform 7



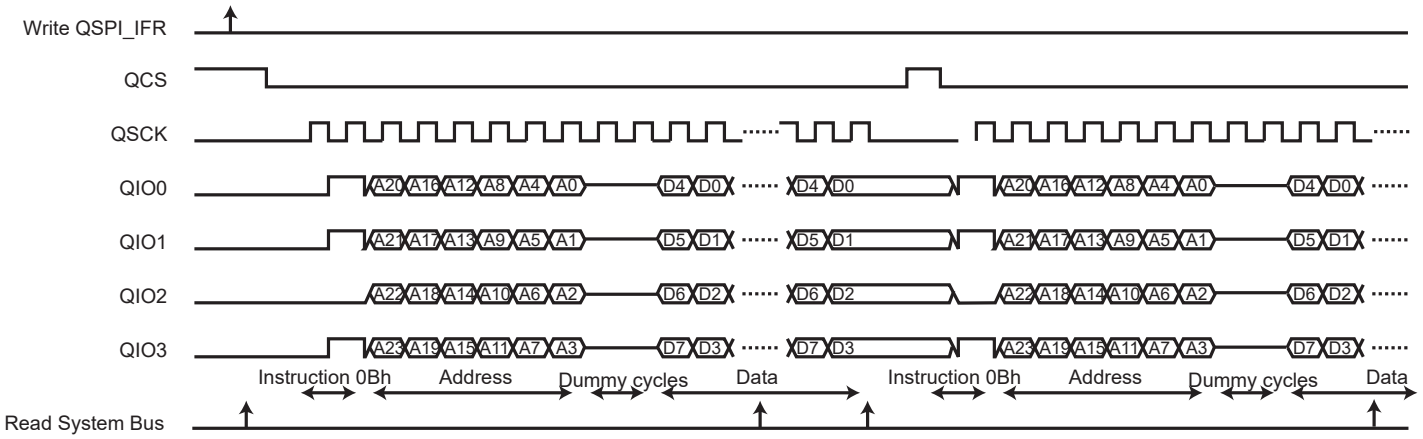
Example 8:

Instruction in Quad SPI, with 24-bit address in Quad SPI, without option, with data read in Quad SPI and two dummy cycles.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B in QSPI_RICR.
- Write 0x0002_08B6 in QSPI_IFR.
- Update configuration (see Updating the QSPI Configuration).
- Read data in the QSPI system bus memory space (0x60000000 - 0x80000000). The address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
- Write a 1 to QSPI_CR.LASTXFR.
- If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA to rise.

Figure 64.33. Instruction Transmission Waveform 8



Example 9:

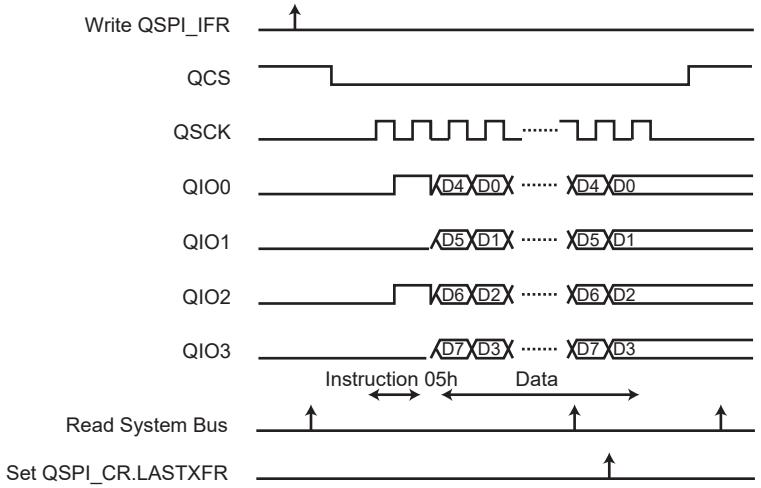
Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, without fetch.

Command: Read Status register (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 (0x60000000 - 0x80000000). Fetch is disabled.
- Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1.
- Write a 1 to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64.34. Instruction Transmission Waveform 9



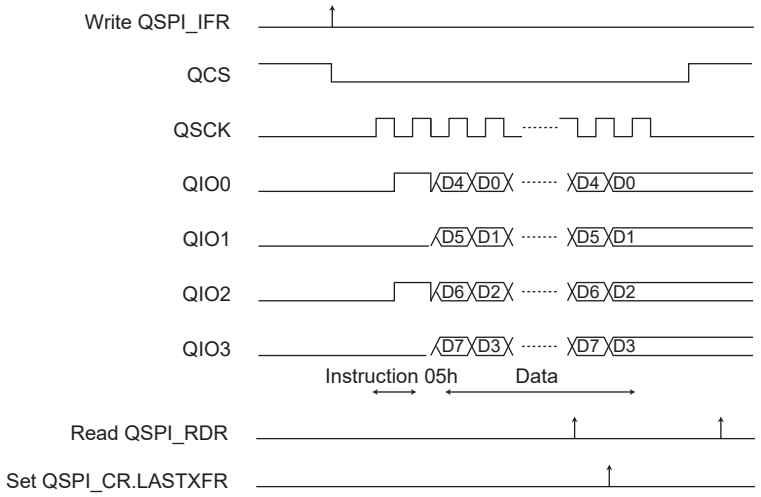
Example 10:

Instruction in Quad SPI, without address, without option, with data read in Quad SPI, without dummy cycles, and read launched through the peripheral bus.

Command: Read Status register (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 for flag RDRF and Read data in QSPI_RDR.
- Write a 1 to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.
- Read data in QSPI_RDR.

Figure 64.35. Instruction Transmission Waveform 10



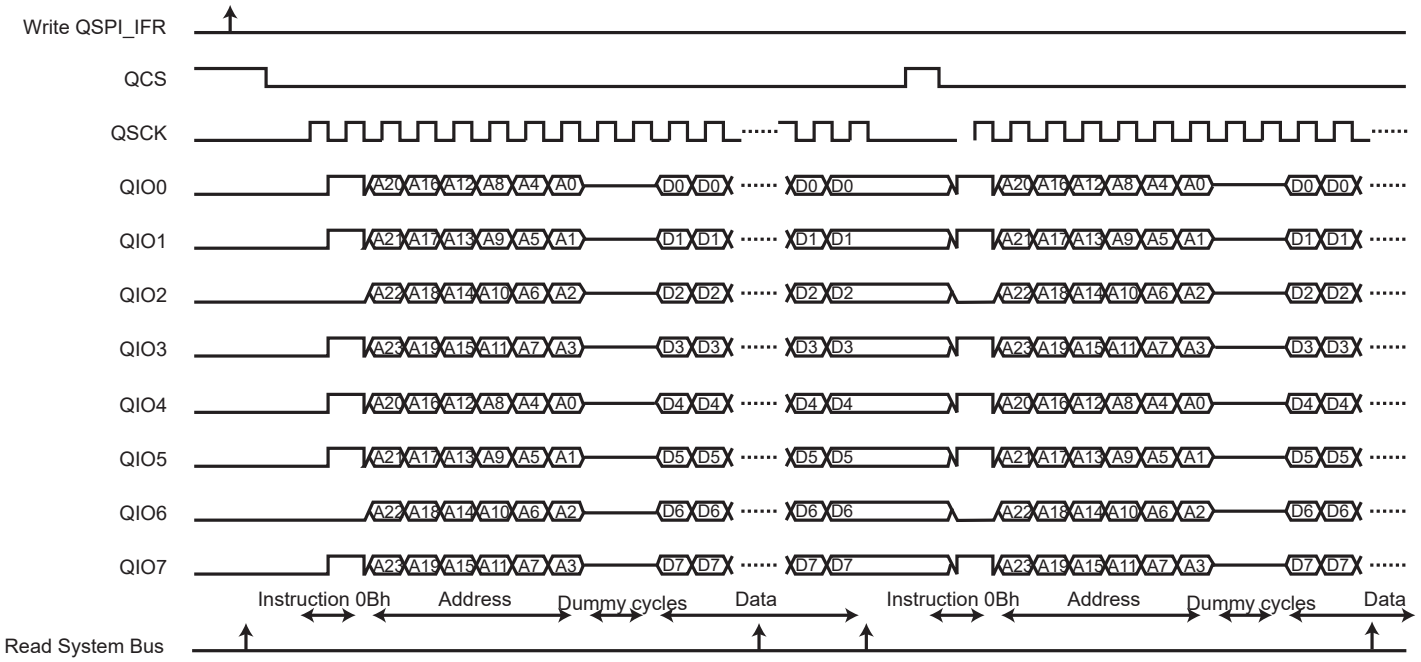
Example 11:

Instruction in Octal Twin-Quad SPI, with 24-bit address in Octal Twin-Quad SPI, without option, with data read in Octal Twin-Quad SPI and two dummy cycles.

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 (0x60000000 - 0x80000000). The address of the system bus read accesses is always used.
- Wait for QSPI_SR.RBUSY=0 and QSPI_SR.HIDLE=1'.
- Write a 1 to QSPI_CR.LASTXFR.
- If QSPI_SR.CSS=0, wait for QSPI_ISR.CSRA to rise.

Figure 64.36. Instruction Transmission Waveform 11



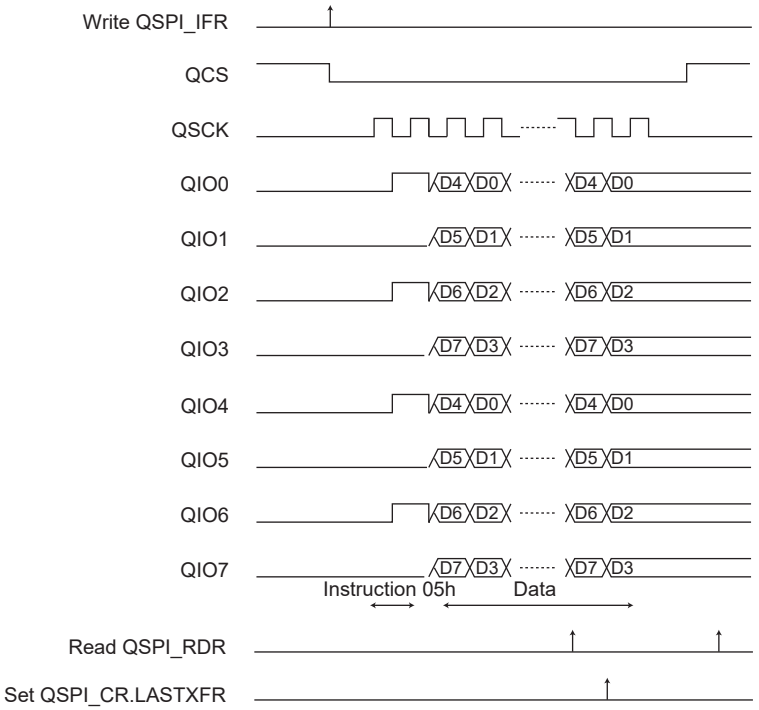
Example 12:

Instruction in Octal Twin-Quad SPI, without address, without option, with data read in Octal Twin-Quad SPI, without dummy cycles and read launched through the peripheral bus.

Command: Read Status register (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.
- Write a 1 to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.
- Read data in the QSPI_RDR register.

Figure 64.37. Instruction Transmission Waveform 12



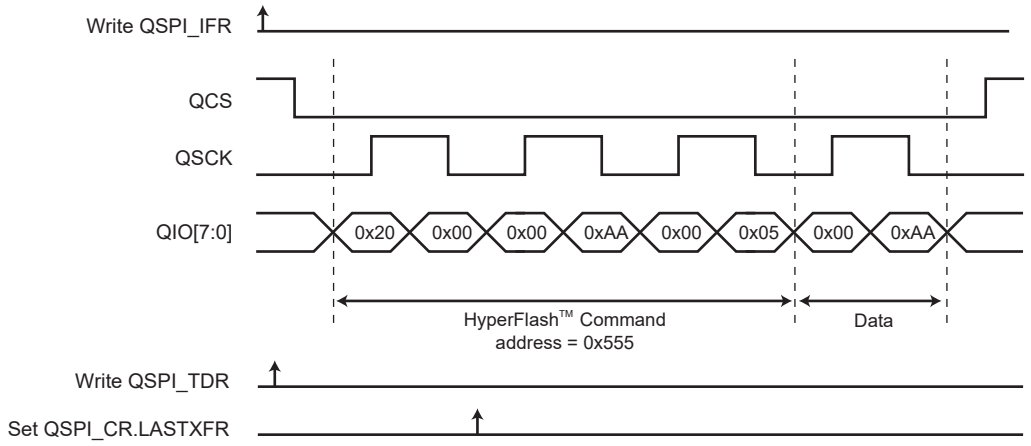
Example 13:

Classic HyperFlash command. Instruction in Octal DDR SPI, HyperFlash mode, without option, with data write, without dummy cycles and write launched through the peripheral bus.

Command: Data 0xAA at address 0x555

- Write 0x36C0_8099 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 64.38. Instruction Transmission Waveform 13



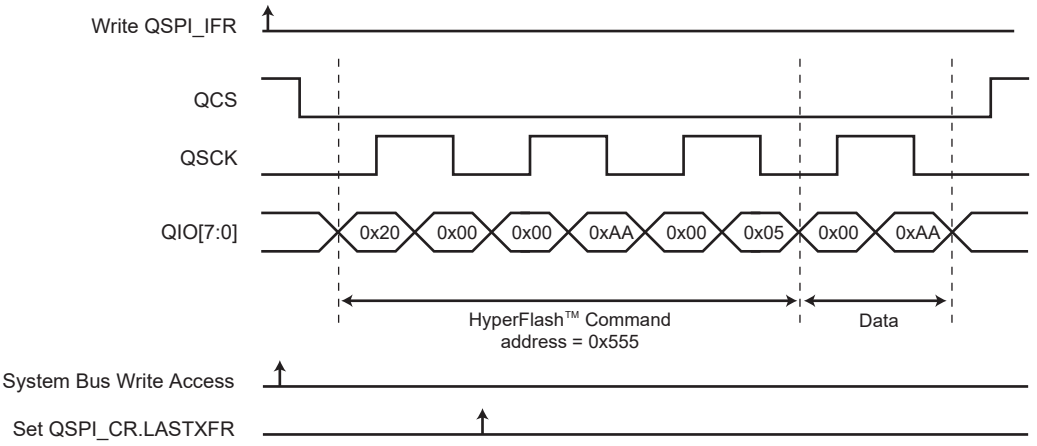
Example 14:

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_8099 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 (0x60000000 - 0x80000000).
- Wait for QSPI_ISR.LWRA to rise.
- Write a 1 to QSPI_CR.LASTXFR.
- Wait for QSPI_ISR.CSRA to rise.

Figure 64.39. Instruction Transmission Waveform 14



64.6.8.12.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.

See examples 11 and 12 in Instruction Frame Transmission Examples.

In this mode, only the DUAL_CMD, OCT_OUTPUT, OCT_IO and OCT_CMD configurations are supported by QSPI_IFR.WIDTH.

- DUAL_CMD: addresses both dies included in the Twin-Quad memory in single bit SPI for the Instruction, Address and Data phases (1-1-1).
- OCT_OUTPUT: addresses both dies included in the Twin-Quad memory in single bit SPI for the Instruction and Address phases, then in Quad SPI for the Data phase (1-1-4).
- OCT_IO: addresses both dies included in the Twin-Quad memory in single bit SPI for the Instruction phase, then in Quad SPI for the Address and Data phases (1-4-4).
- OCT_CMD: addresses both dies included in the Twin-Quad memory in Quad SPI for the Instruction, Address and Data phases (4-4-4).

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, 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 dies.

- 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 dies, 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 dies, 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 dies.

- 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 dies, 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 dies, 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]}.

For example, if data 0xB5 is to be written in both registers of both dies, 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]}.

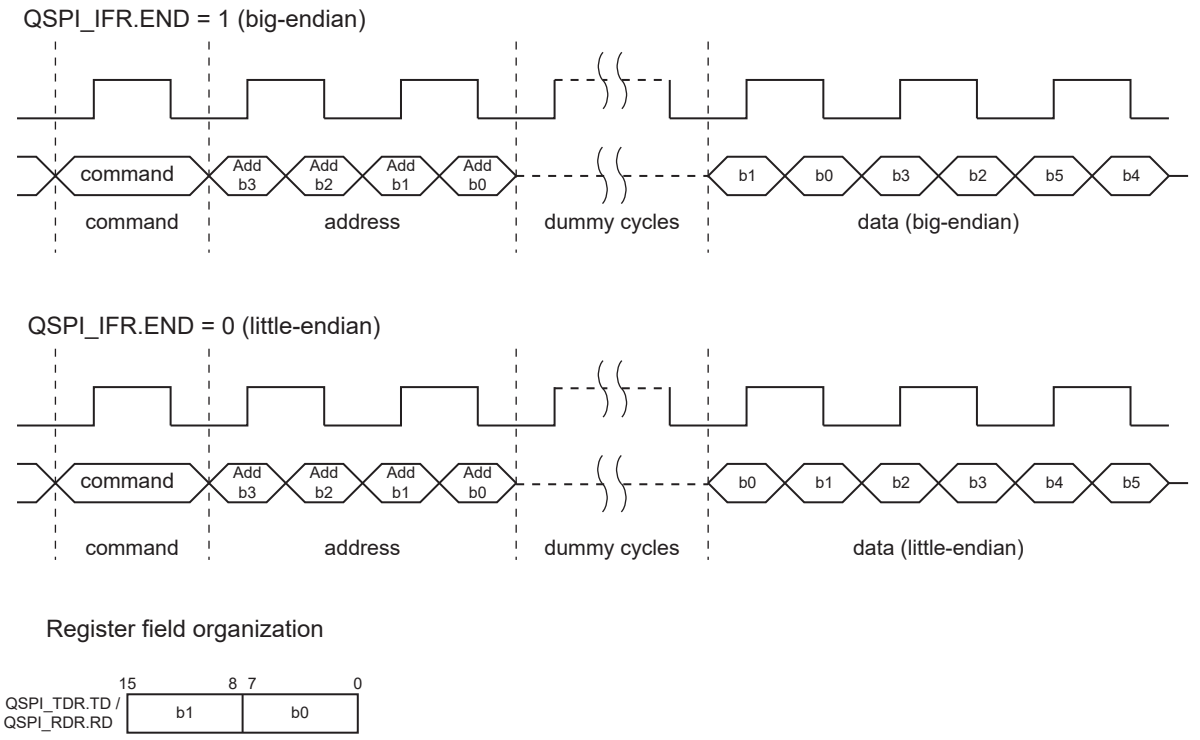
For example, if data 0xA7 is to be read in both registers of both dies, the halfword read in QSPI_RDR is 0xAA77.

64.6.8.13.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 data sheet.

Figure 64.40. Instruction Frame Endianness



64.6.8.14.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.

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 must be performed (refer to the memory data sheet 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.

64.6.8.15.HyperFlash Mode

The QSPI supports HyperFlash memories. To enable HyperFlash mode, set QSPI_IFR.PROTTYP to 3.

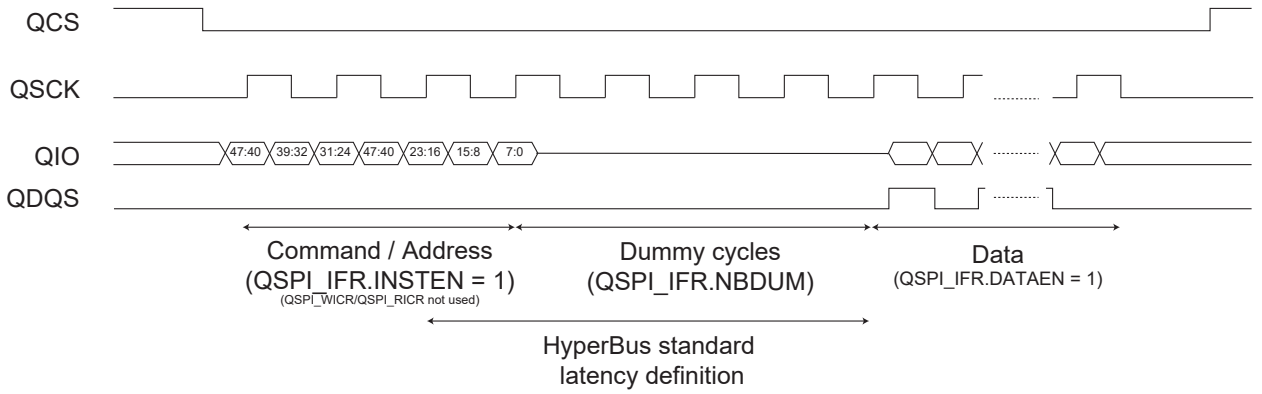
See examples 13 and 14 in Instruction Frame Transmission Examples.

HyperFlash memories use Octal DDR communication. Recommendations provided in Octal DDR Mode must be followed.

HyperFlash memories define a “command/address” field where the address of the transfer is included and merged in the command. For this reason, in this mode QSPI_IFR.ADDREN must be set to 0 and QSPI_IFR.INSTEN must be set to 1.

The HyperFlash “command/address” field merges the transfer address and the transfer characteristics (such as read/write, memory/register space, burst type, etc.) so that there is no specific opcode per transfer type. As a consequence, QSPI_WICR and QSPI_RICR are not used in this mode. See the following figure.

Figure 64.41. HyperBus Frame



Once HyperFlash mode is enabled, the procedure to access the memory is the same as for classic QSPI memories. See Instruction Frame Transmission (see Figure 64.14, Figure 64.15, Figure 64.18, Figure 64.20, Figure 64.22).

For the HyperFlash Write Buffer procedure, QSPI_IFR.HFWBEN must be set. When this bit is set, a new command is issued for each halfword written. In this mode, halfword accesses are mandatory. See Figure 64.23 and Figure 64.24.

Note:

1. In HyperFlash mode, some bits of the HyperFlash command are set automatically. For instance, the “Burst Type” bit of the HyperFlash command (bit 45) is always set to 1 (linear burst).
2. The HyperFlash standard latency definition counts the latency starting from the fourth byte of the instruction, whereas the QSPI counts the dummy cycles starting from the end of the instruction field. Therefore, the value to set in QSPI_IFR.NBDUM is the HyperFlash latency value minus 1.

64.6.8.16.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 the TCNTM field of the Timeout register (QSPI_TOUT), the flag QSPI_ISR.TOUT rises.

When the TOUT flag rises, any access waiting on the QSPI core is released with an error response so that the system bus is released. Any further access to the QSPI core receives an error response until the TOUT flag is cleared.

After a time-out error, the TOUT flag must be cleared 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.

64.6.9. Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories.

Scrambling is possible only if SMM =1.

If TFRTYP=1, on-the-fly scrambling via system bus accesses is possible (see Figure 64.19, Figure 64.21 and Figure 64.24 for specific flowcharts).

If TFRTYP=0 and SMRM=1, scrambling via peripheral bus accesses is possible for write accesses (see Figure 64.17 and Figure 64.23 for specific flowcharts).

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 (memory, for example).

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.

64.6.9.1. Clearing Scrambling Keys on a Tamper Event

On a tamper detection event, an immediate clear of the scrambling key (in the Scrambling Key register QSPI_SKR) is performed if QSPI_MR.TAMPCLR is set. For details about the tamper event source, refer to sections Safety and Security Features and/or Pinout.

64.6.10. Register Write Protection

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

If a write access to a write-protected register is detected, the WPVS flag in the WPEN bit in the 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 QSPI_WPSR.

The following registers can be write-protected when WPEN is set in QSPI_WPMR:

- QSPI Mode Register
- QSPI Serial Clock Register
- QSPI Scrambling Mode Register
- QSPI Scrambling Key Register
- QSPI Write Instruction Code Register
- QSPI Read Instruction Code Register
- QSPI Instruction Address Register
- QSPI Instruction Frame Register

- QSPI Refresh Register
- QSPI Write Access Counter Register
- QSPI Timeout Register

The following register 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

64.6.11. Peripheral Bus Access Errors

Some peripheral bus accesses (to the QSPI register interface) can result in errors if they generate an issue with the QSPI configuration.

Error flags considered as peripheral bus access errors are(1):

- QSPI_WPSR.WPVS
- QSPI_WPSR.SWE[2]
- QSPI_ISR.RBSYERR

In addition to the above error flags, any peripheral bus access error sets the QSPI_ISR.SECE flag.

Peripheral bus access errors are reported through interrupts (via QSPI_ISR.SECE) and the QSPI_WPSR flags.

Notes:

1. When a peripheral bus access error occurs, disable the QSPI and then check and/or reconfigure all registers.
2. QSPI_WPSR.SWETYP indicates the software error type when QSPI_WPSR.SWE is set. Note that SWETYP=UNDEF_DEVICE and SWETYP=UNEXP_DMACMD are not peripheral bus access errors.

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				QSPIDIS	QSPIEN
0x04	QSPI_MR	31:24	DLYCS[7:0]							
		23:16	DLYBCT[7:0]							
		15:8			AQICMEN		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	CSF	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x14	QSPI_IER	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE	CSF	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x18	QSPI_IDR	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE	CSF	CSR
		7:0					OVRES	TXEMPTY	TDRE	RDRF
0x1C	QSPI_IMR	31:24								
		23:16							TOUT	RFRSHD
		15:8	CSRA	CSFA	QITR	QITF	LWRA	INSTRE	CSF	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				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[2:0]			HFWBEN	DDRCMDEN	DQSEN	APBTFRTYP
		23:16	SMRM	END	NBDUM[5:0]					
		15:8	DDREN	CRM		TFRTYP	ADDRL[1:0]		OPTL[1:0]	
		7:0	DATAEN	OPTEN	ADDREN	INSTEN	WIDTH[3:0]			
0x3C	QSPI_RICR	31:24								
		23:16	RDOPT[7:0]							
		15:8	RDINST[15:8]							
		7:0	RDINST[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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 ... 0x63	Reserved									
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				QSPIDIS	QSPIEN
Access	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 are not affected by a software reset.

Value	Description
0	No effect.
1	Resets the QSPI. A software 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 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 set to 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.

64.7.2. QSPI Mode Register

Name: QSPI_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit 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
			AQICMEN		NBBITS[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
	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
This field 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 is written to 0, one GCLK period is inserted by default.
Otherwise, the following equation determines the delay:

- DLYCS=Minimum inactive × f_{GCLK}

Bits 23:16 – DLYBCT[7:0] Delay Between Consecutive Transfers

- SMM=0
  This field defines the delay between two consecutive transfers without releasing 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_{GCLK}) / 32
- SMM=1
  This field 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_{GCLK}

Bit 13 – AQICMEN QSPI Inter-Chip Communication Mode Enable All

Value	Name	Description
0	DISABLED	The QSPI Inter-Chip mode is disabled.
1	ENABLED	The QSPI Inter-Chip mode is enabled.

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 and QSPI_IFR.PROTYP is not selecting a RAM memory.

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	The DQS Delay cell is disabled.
1	ENABLED	The DQS Delay cell is enabled. The DQS Delay cell automatic refresh is triggered according to the QSPI_REFRESH configuration.

Bit 2 – WDRBT Wait for 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	CSF	CSR
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
					OVRES	TXEMPTY	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 17 – TOUT QSPI Time-out (cleared by writing QSPI_CR.RTOUT)

Value	Description
0	No QSPI time-out occurred.
1	At least one QSPI time-out occurred.

Bit 16 – RFRSHD Refresh Done (cleared on read)

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
See Device Selection Flags.

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
See Device Selection Flags.

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 (cleared on read)

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 (cleared on read)

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_ISR or NBWRA=0.
1	At least one last write access has been sent since the last read of QSPI_ISR.

Bit 10 – INSTRE Instruction End Status (cleared on read)

Value	Description
0	No instruction end has been detected since the last read of QSPI_ISR.
1	At least one instruction end has been detected since the last read of QSPI_ISR.

Bit 9 – CSF Chip Select Fall (cleared on read)
See Device Selection Flags.

Value	Description
0	No chip select rise has been detected since the last read of QSPI_ISR.
1	At least one chip select rise has been detected since the last read of QSPI_ISR.

Bit 8 – CSR Chip Select Rise (cleared on read)
See Device Selection Flags.

Value	Description
0	No chip select rise has been detected since the last read of QSPI_ISR.
1	At least one chip select rise has been detected since the last read of QSPI_ISR.

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)
TXEMPTY is set to zero when the QSPI is disabled or at reset. The QSPI enable command sets this bit to one.

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 is set to 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	CSF	CSR
Access	W	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 9 – CSF Chip Select Fall 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	CSF	CSR
Access	W	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 9 – CSF Chip Select Fall 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	CSF	CSR
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
					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 9 – CSF Chip Select Fall 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 the WPEN bit 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 is set to zero, the QCS valid to QSCK transition is half the QSCK clock period.
Otherwise, the following equation determines the delay:

- DLYBS = Delay Before QSCK × f_{GCLK}

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-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
				HIDLE	RBUSY	CSS	QSPIENS	SYNCBSY
Access				R	R	R	R	R
Reset				0	0	0	0	0

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 client bus interface has no activity.
1	The client 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	Allows access to any register.
1	Some register accesses must not be accessed. 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 the WPEN bit 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 the WPEN bit 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 the WPEN bit is cleared in the QSPI Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		PROTTYP[2:0]			HFWBEN	DDRCMDEN	DQSEN	APBTFRTYP
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
	SMRM	END	NBDUM[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
	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 30:28 – PROTTYP[2:0] Protocol Type

Value	Name	Description
0	STD_SPI	Standard (Q)SPI protocol (to be selected if no other value matches the device)
1	TWIN_QUAD	Twin-Quad protocol
2	OCTAFLASH	OctaFlash protocol or Octal DDR protocol with 16-bit instruction code
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. **Mandatory if the 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 the DQS signal.

Value	Description
1	Data from the memory are sampled with the DQS signal.

Bit 24 – APBTFRTYP Peripheral Bus Transfer Type

Value	Description
0	Register transfer to the memory is a write transfer. Used when TRFTYP is written to 0 and SMRM to 1.
1	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
See Instruction Frame Transmission for details.

Value	Description
0	Serial Memory registers are written via system bus access.
1	Serial Memory registers are written via peripheral bus access.

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 21:16 – NBDUM[5:0] Number Of Dummy Cycles
Defines the number of dummy cycles (also called read latency) required by the serial 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 system bus or the peripheral bus can be used to initiate the transfer (SMRM bit). • If the peripheral bus is used, the RDRF and TDRE flags help to control the frame. • Scrambling is possible only for write accesses and if the peripheral bus 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 system bus size accesses only are performed. • The internal optimization algorithm is enabled to minimize latency, and protocol specificities are handled automatically. • Seamless scrambling is possible. • Seamless handling of HyperFlash Write Buffer programming command (one command for each data) • Address shift is handled seamlessly (halfword memories, for example).

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 the Quad SPI protocol, requiring a minimum length of 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

This register can only be written if the WPEN bit 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
	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 the WPEN bit 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	Scrambling/unscrambling is disabled.
1	ENABLED	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 the WPEN bit 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 the WPEN bit 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 GCLK clock periods the delay between two refreshes of analog blocks. See Refresh Sequence.

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 the WPEN bit 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

64.7.19. QSPI Timeout Register

Name: QSPI_TOUT
Offset: 0x64
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit 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 GCLK clock periods when the connected client does not answer and before the TOUT flag is set.

64.7.20. 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)

64.7.21. 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. When SWE=1, indicates the faulty register address, if relevant.

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 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 V4.51, 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 and enhanced features.

The SDMMC is clocked by three asynchronous clocks (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 V4.51
- 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 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 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)
- 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

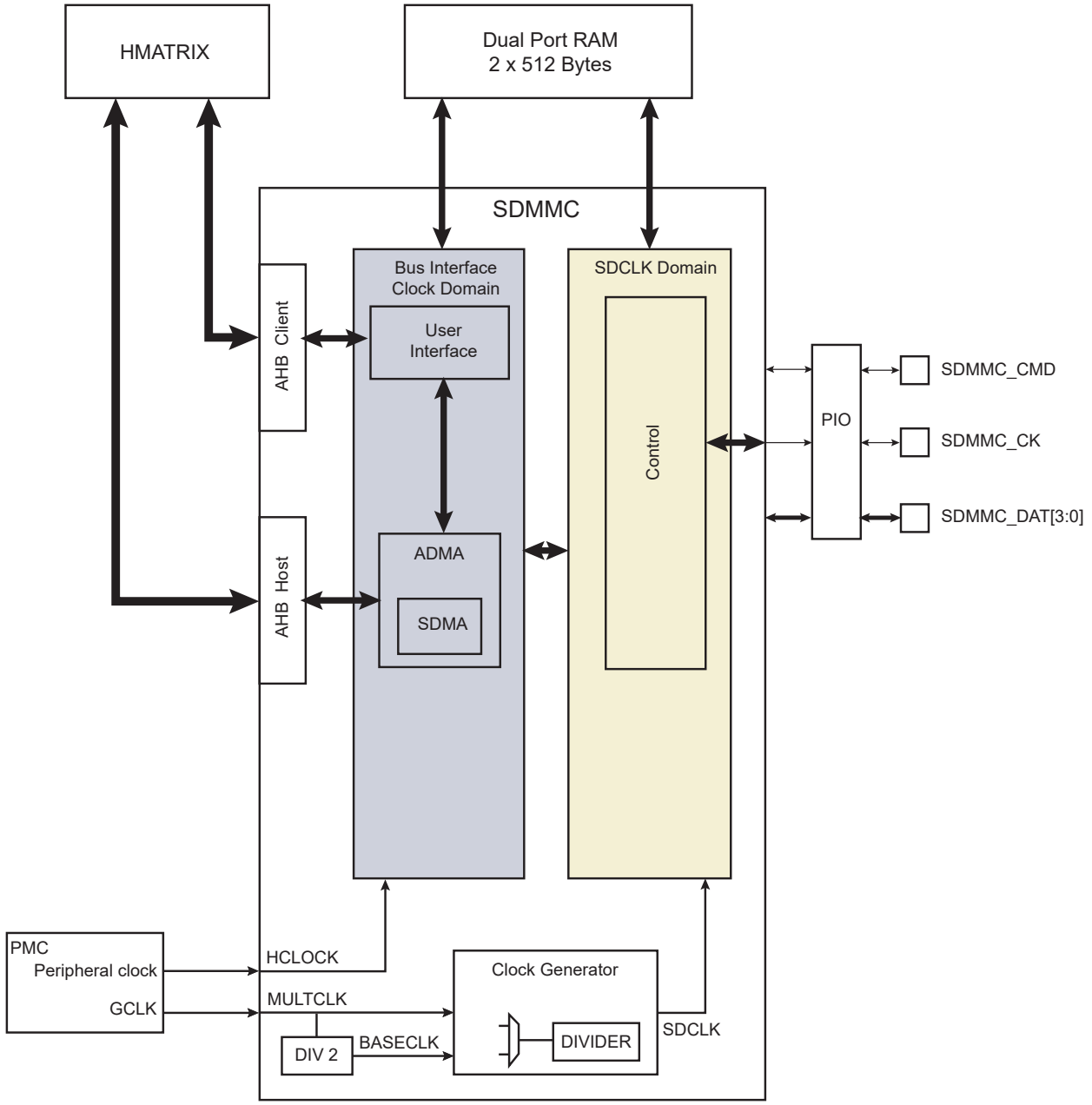
65.3. Reference Documents

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 4.51	http://www.jedec.org

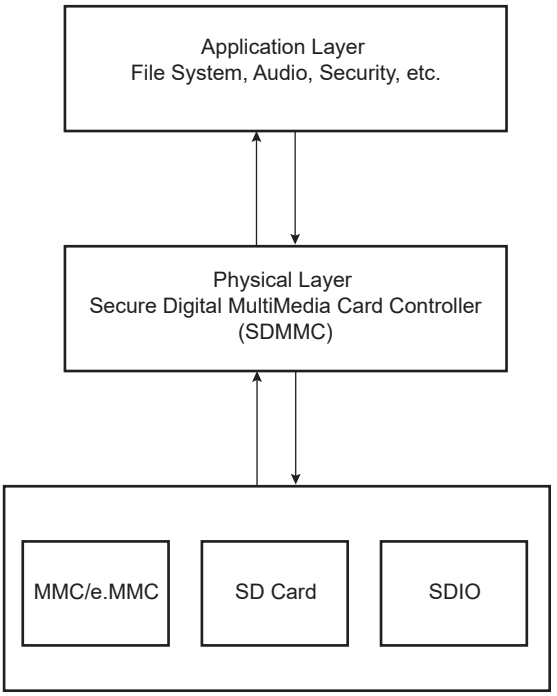
65.4. Block Diagram

Figure 65.1. SDMMC Block Diagram



65.5. Application Block Diagram

Figure 65.2. Application Block Diagram



65.6. Pin Name List

Table 65.2. I/O Lines Description for 8-bit Configuration

Pin Name	Pin Description	Type
SDMMC_CMD	SD Card/SDIO/e.MMC Command/Response Line	I/O
SDMMC_CK	SD Card/SDIO/e.MMC Clock Signal	Output
SDMMC_DAT[3:0]	SD Card/SDIO/e.MMC Data Lines	I/O

Note: When several SDMMCs are embedded in a product, SDMMC_CK refers to SDMMCx_CK, SDMMC_CMD to SDMMCx_CMD and SDMMC_DATy to SDMMCx_DATy.

65.7. Product Dependencies

65.7.1. 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.

65.7.2. 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.

65.7.3. 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.

65.8. 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.

65.9. 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.

65.9.1. 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 power-on 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”).

65.9.1.1. 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’.

65.9.1.2. 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'.

65.9.1.3.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 [1:0].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’.

65.10. 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]							
		15:8					BUFRDEN	BUFWREN	RTACT	WTACT
		7:0						DLACT	CMDINHD	CMDINHC
0x28	SDMMC_HC1R (SD_SDIO)	7:0		CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
0x28	SDMMC_HC1R (EMMC)	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 (EMMC)	7:0							CONTR	STPBGR
0x2B	SDMMC_WCR (SD_SDIO)	7:0								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
0x30	SDMMC_NISTR (SD_SDIO)	15:8	ERRINT							CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x30	SDMMC_NISTR (EMMC)	15:8	ERRINT	BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x32	SDMMC_EISTR (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x32	SDMMC_EISTR (EMMC)	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x34	SDMMC_NISTER (SD_SDIO)	15:8								CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x34	SDMMC_NISTER (EMMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x36	SDMMC_EISTER (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x36	SDMMC_EISTER (EMMC)	15:8				BOOTAE			ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x38	SDMMC_NISIER (SD_SDIO)	15:8								CINT
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x38	SDMMC_NISIER (EMMC)	15:8		BOOTAR						
		7:0			BRDRDY	BWRRDY	DMAINT	BLKGE	TRFC	CMDC
0x3A	SDMMC_EISIER (SD_SDIO)	15:8							ADMA	ACMD
		7:0	CURLIM	DATEND	DATCRC	DATTEO	CMDIDX	CMDEND	CMDCRC	CMDTEO
0x3A	SDMMC_EISIER (EMMC)	15:8				BOOTAE			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								
0x3E	SDMMC_HC2R (EMMC)	15:8	PVALEN							
		7:0								
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								
		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]							
0x5C ... 0x5F	Reserved									
0x60	SDMMC_PVR0	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x62	SDMMC_PVR1	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x64	SDMMC_PVR2	15:8						CLKGSEL	SDCLKFSEL[9:8]	
		7:0	SDCLKFSEL[7:0]							
0x66 ... 0xFB	Reserved									
0xFC	SDMMC_SISR	15:8								
		7:0							INTSSL[1:0]	
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			BOOTA	OPD	DDR		CMDTYP[1:0]	
0x0205	SDMMC_MC2R	7:0							ABOOT	SRESP
0x0206 ... 0x0207	Reserved									
0x0208	SDMMC_ACR	31:24								
		23:16								
		15:8								
		7:0							BMAX[1:0]	
0x020C	SDMMC_CC2R	31:24								
		23:16								
		15:8								
		7:0								FSDCLKD
0x0210 ... 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

65.10.1. 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: the system memory address for an 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: 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.

65.10.2. SDMMC Block Size Register

Name: SDMMC_BSR
Offset: 0x04
Reset: 0x0000
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
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
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.

65.10.3. 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.

65.10.4. 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” .

65.10.5. 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 Determining 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.

Value	Name	Description
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 Determining 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.

Value	Name	Description
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.

Value	Name	Description
0	DISABLED	DMA functionality is disabled.
1	ENABLED	DMA functionality is enabled.

65.10.6. 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.

Value	Name	Description
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.

Value	Name	Description
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

65.10.7. 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]

65.10.8. 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.

65.10.9. SDMMC Present State Register

Name: SDMMC_PSR
Offset: 0x24
Reset: 0x01F80000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								CMDLL
Access								R
Reset								1

Bit	23	22	21	20	19	18	17	16
	DATLL[3:0]							
Access	R	R	R	R				
Reset	1	1	1	1				

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 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.

65.10.10.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
		CARDDTL		DMASEL[1:0]		HSEN	DW	LEDCTRL
Access		R/W		R/W	R/W	R/W	R/W	R/W
Reset		0		0	0	0	0	0

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.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Value	Name	Description
0	1_BIT	1-bit mode.
1	4_BIT	4-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.

Value	Name	Description
0	OFF	LED off.
1	ON	LED on.

65.10.11.SDMMC Host Control 1 Register (e.MMC)

Name: SDMMC_HC1R (EMMC)
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.
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.
Note: If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

Value	Name	Description
0	1_BIT	1-bit mode.
1	4_BIT	4-bit mode.

65.10.12.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[3:0] (tri-state) and drives SDMMC_CK to low level.

65.10.13.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

65.10.14.SDMMC Block Gap Control Register (e.MMC)

Name: SDMMC_BGCR (EMMC)
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

65.10.15.SDMMC Wake-up 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
								WKENCINT
Access								R/W
Reset								0

Bit 0 – WKENCINT Wake-up Event Enable on Card Interrupt
This bit enables a wake-up event via Card Interrupt (CINT) in SDMMC_NISTR. This bit can be set to 1 if FN_WUS (Wake-up Support) in the CIS (Card Information Structure) is set to 1 in the SDIO card.
0 (DISABLED): Wake-up Event disabled.
1 (ENABLED): Wake-up Event enabled.

65.10.16.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): $f_{\mathrm{SDCLK}} = f_{\mathrm{MULTCLK}}/(\mathrm{DIV} + 1)$

When HSEN is set in SDMMC_HC1R, or DDR is set in SDMMC_MC1R, the clock divider (DIV) must be non-zero. 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.
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 wake-up 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.

Value	Description
0	The internal clock stops.
1	The internal clock oscillates.

65.10.17.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.

65.10.18.SDMMC Software Reset Register

Name: SDMMC_SRR
Offset: 0x2F
Reset: 0x00000000
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. 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 Wake-up 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 Capabilities Control Register (except KEY)

Value	Description
0	Work
1	Reset

65.10.19.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
			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 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 wake-up. 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 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.
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.

65.10.20.SDMMC Normal Interrupt Status Register (e.MMC)

Name: SDMMC_NISTR (EMMC)
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.
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
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.

65.10.21.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
							ADMA	ACMD
Access							R/W	R/W
Reset							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 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.
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

65.10.22.SDMMC Error Interrupt Status Register (e.MMC)

Name: SDMMC_EISTR (EMMC)
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			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 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 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.

Value	Description
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

65.10.23.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
			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 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 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.

65.10.24.SDMMC Normal Interrupt Status Enable Register (e.MMC)

Name: SDMMC_NISTER (EMMC)
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.

65.10.25.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
							ADMA	ACMD
Access							R/W	R/W
Reset							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 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.

65.10.26.SDMMC Error Interrupt Status Enable Register (e.MMC)

Name: SDMMC_EISTER (EMMC)
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			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 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 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.

65.10.27.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
			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 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 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.

65.10.28.SDMMC Normal Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_NISIER (EMMC)
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.

65.10.29.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
							ADMA	ACMD
Access							R/W	R/W
Reset							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 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.

65.10.30.SDMMC Error Interrupt Signal Enable Register (e.MMC)

Name: SDMMC_EISIER (EMMC)
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			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 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 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.

65.10.31.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.

65.10.32.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
Access								
Reset								

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_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, 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

65.10.33.SDMMC Host Control 2 Register (e.MMC)

Name: SDMMC_HC2R (EMMC)
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
Access								
Reset								

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_CCR.SDCLKFSEL and SDMMC_CCR.CLKGSEL are set by the user.
If this bit is set to 1, 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.

65.10.34.SDMMC Capabilities 0 Register

Name: SDMMC_CA0R
Offset: 0x40
Reset: 0x27E832B2
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 the 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
	SLTYPE[1:0]		ASINTSUP	SB64SUP		V18VSUP	V30VSUP	V33VSUP
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset	0	0	1	0		1	1	1

Bit	23	22	21	20	19	18	17	16
	SRSUP	SDMASUP	HSSUP		ADMA2SUP	ED8SUP	MAXBLKL[1:0]	
Access	R	R	R		R	R	R	R
Reset	1	1	1		1	0	0	0

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	0	0	1	1	0	0	1	0

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	1		1	1	0	0	1	0

Bits 31:30 – SLTYPE[1:0] Slot Type
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

Value	Description
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 (read-only)

Indicates whether the SDMMC supports the Suspend/Resume functionality. If 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 (read-only)

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 (read-only)

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 (read-only)

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 (read-only)

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 (read-only)

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

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

Indicates 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

Shows the timeout clock frequency (TEOCLK) used to detect Data Timeout Error.
If 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: $f_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{MHz}}$

65.10.35.SDMMC Capabilities 1 Register

Name: SDMMC_CA1R
Offset: 0x44
Reset: 0x00010070
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 the 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
Access								
Reset								

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		0	0	0

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.
$f_{\mathrm{MULTCLK}} = f_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT} + 1)$

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.

65.10.36.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.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR33}V$

65.10.37.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.

65.10.38.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.

65.10.39.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

65.10.40.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.

65.10.41.SDMMC Preset Value Register

Name: SDMMC_PVRx
Offset: 0x60 + x*0x02 [x=0..2]
Reset: 0x0000
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	HSEN (SDMMC_HC1R)
Default Speed	0
High Speed	1

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

When Preset Value Enable (PVALEN) in SDMMC_HC2R is set to 1, 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
						CLKGSEL	SDCLKFSEL[9:8]	
Access						R/W	R/W	R/W
Reset						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

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.

65.10.42.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[1:0]	
Access							R	R
Reset							0	0

Bits 1:0 – INTSSL[1:0] Interrupt Signal for Each Slot
These status bits indicate the logical OR of Interrupt Signals and Wake-up Signal for each SDMMC instance in the product (INTSSL[x] corresponds to SDMMCx instance in the product).

65.10.43.SDMMC Host Controller Version Register

Name: SDMMC_HCVR
Offset: 0xFE
Reset: 0x00001802
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	0	1	1	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

65.10.44.SDMMC Additional Present State Register

Name: SDMMC_APSR
Offset: 0x200
Reset: –
Property: Read-only

Register reset value:

Instance	Reset Value
SDMMC0, SDMMC1	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
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.

65.10.45.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
			BOOTA	OPD	DDR		CMDTYP[1:0]	
Access			R/W	R/W	R/W		R/W	R/W
Reset			0	0	0		0	0

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.

65.10.46.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.

65.10.47.SDMMC AHB Control Register

Name: SDMMC_ACR
Offset: 0x208
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								

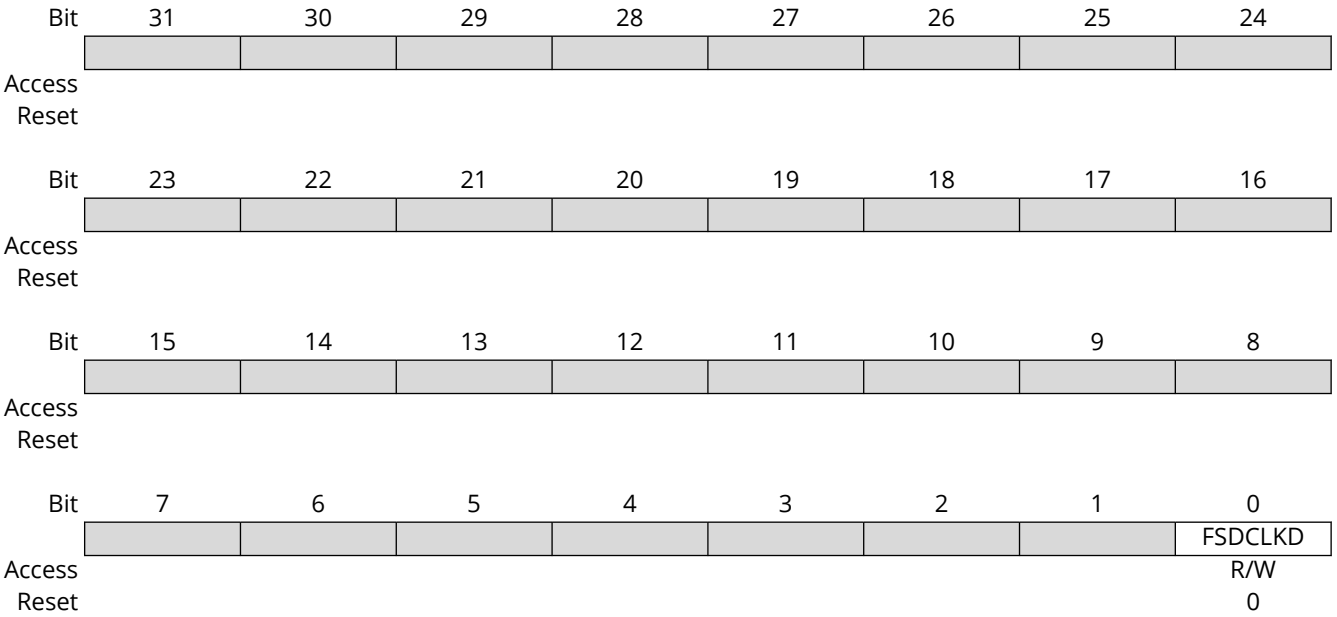
Bit	7	6	5	4	3	2	1	0
							BMAX[1:0]	
Access							R/W	R/W
Reset							0	0

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.

65.10.48.SDMMC Clock Control 2 Register

Name: SDMMC_CC2R
Offset: 0x20C
Reset: 0x00000000
Property: Read/Write



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.

65.10.49.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.

65.10.50.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.

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.

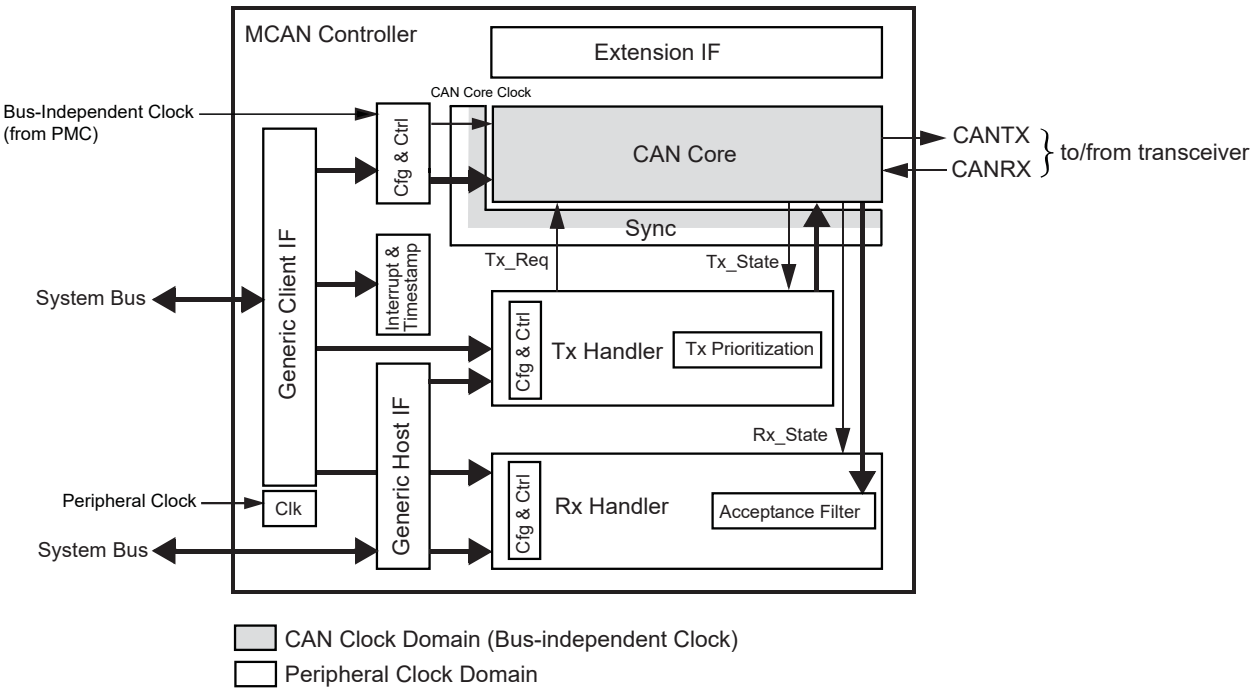
Acceptance filtering is implemented by a combination of up to 128 filter elements, where each element can be configured as a range, as a bit mask, or as a dedicated ID filter.

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. To achieve these frequencies, PMC GCLK must select the UPLLCK (480 MHz) as source clock and divide by 24,12, or 6. 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.

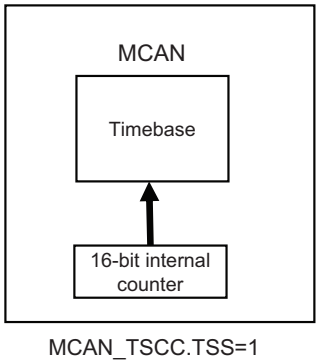
The MSBs [bits 31:16] of the CAN Message RAM for CAN0 and CAN1 are configured at 0x00300000.

66.4.5. Timestamping

66.4.5.1. Internal 16-bit Timestamp Generation

Set MCAN_TSCC.TSS to 1 to use an internal 16-bit timestamp. See the following figure.

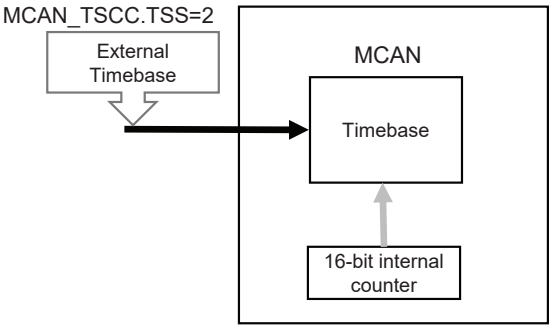
Figure 66.2. 16-Bit Timestamping with Internal Counter



66.4.5.2. External 16-bit Timestamp Generation

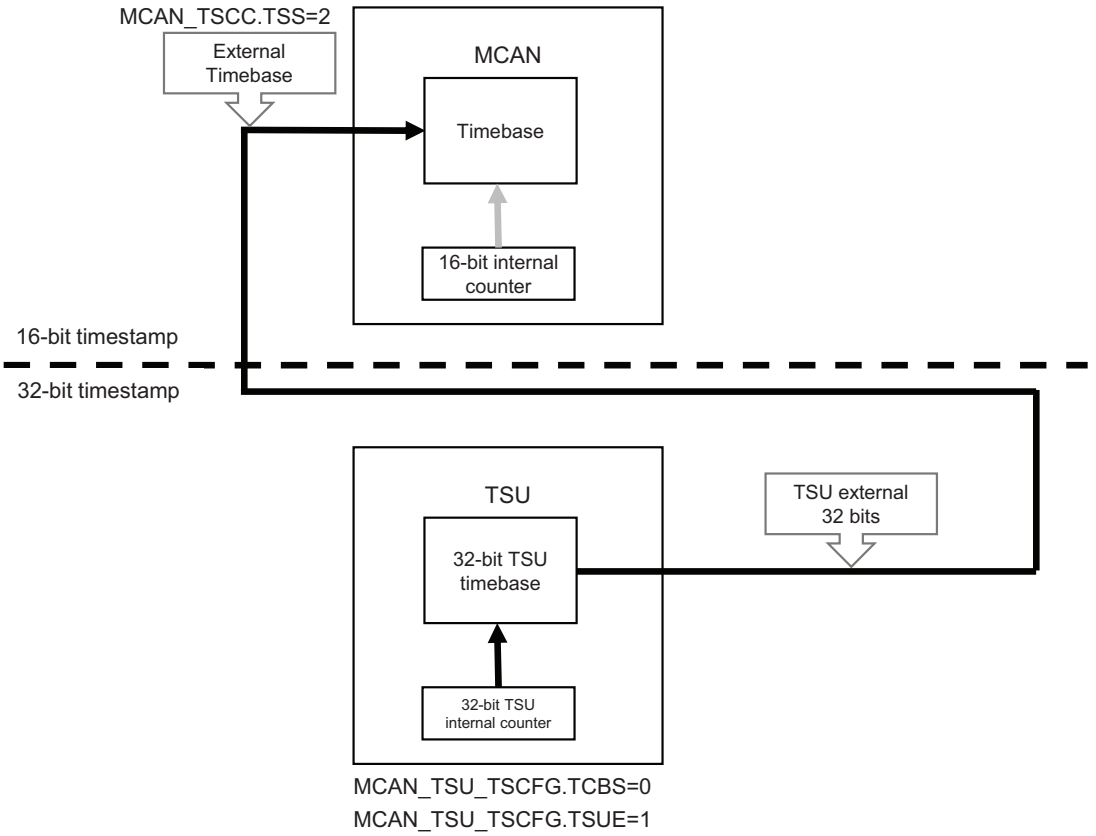
An external timebase is provided by TSU when MCAN_TSCC.TSS is set to 2. See the following figure.

Figure 66.3. 16-Bit Timestamping with External Timebase



Timestamping uses the value of the TSU counter. In this case, both the register in MCAN (external timebase) and the register in TSU must be programmed to configure the counter. See the following figure.

Figure 66.4. 16-Bit and 32-Bit Timestamping with External TSU

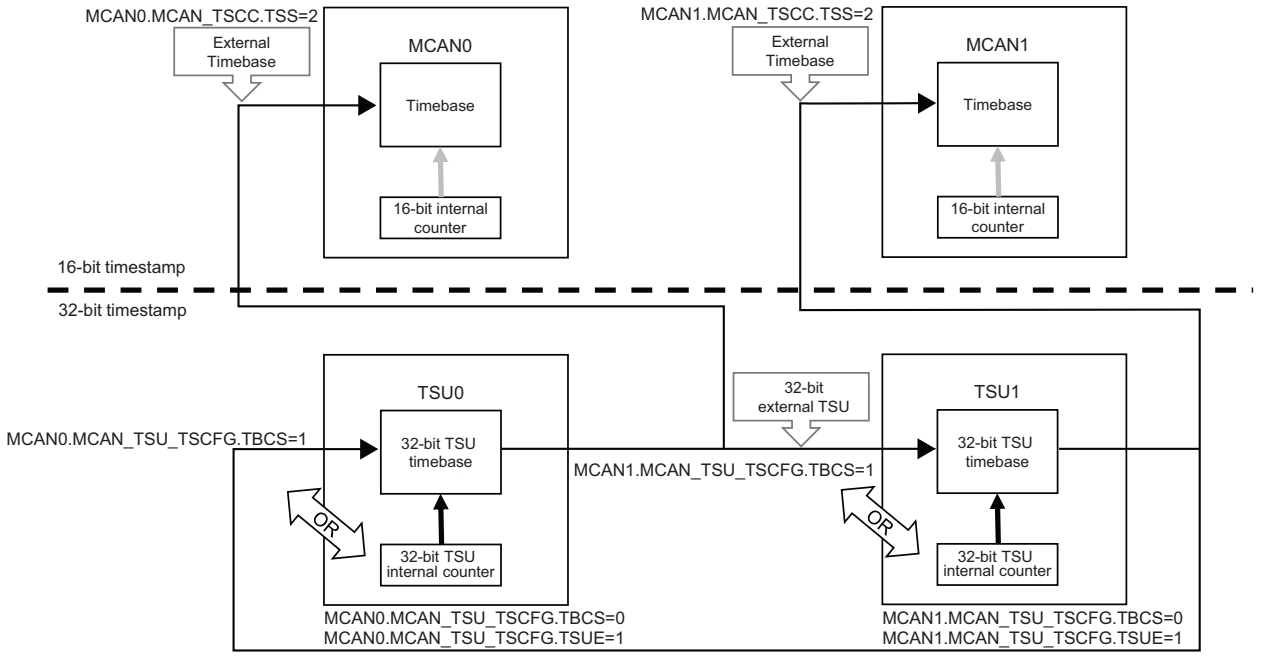


66.4.5.3. Either Internal or External 32-bit Timestamp Generation

Either the TSU internal timebase or the TSU external timebase can be selected for timestamping, by setting MCAN_TSU_TSCFG.TBCS to 0 or MCAN_TSU_TSCFG.TBCS to 1, respectively.

For each MCAN, MCAN_TSU_TSCFG.TSUE must be set to 1. See the following figure.

Figure 66.5. 16-Bit and 32-Bit Timestamping with Chained External TSU



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 start-up 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 Data 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 (for example, 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:

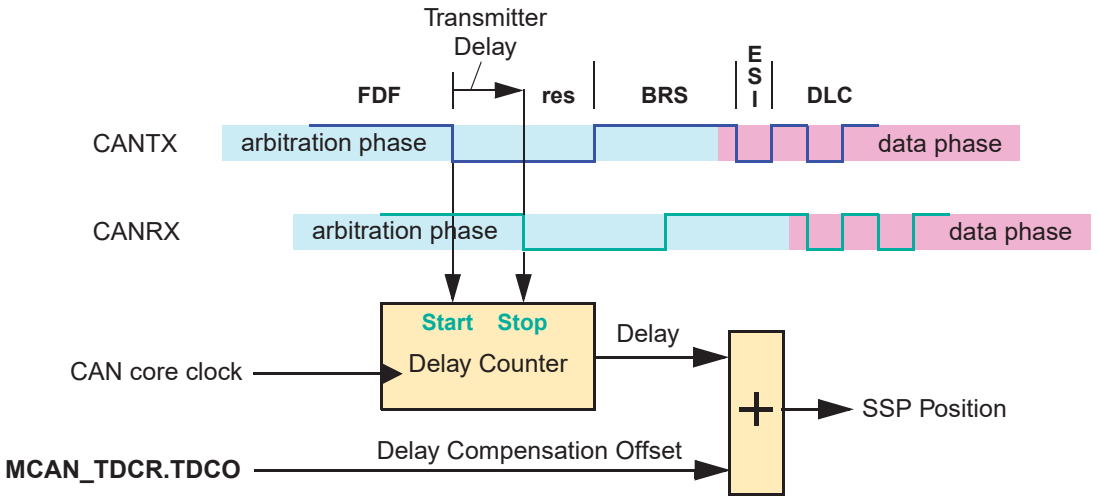
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANTX to CANRX and the configured delay compensation offset MCAN_TDCR.TDCO has to be less or equal 127 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 66.6. 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.

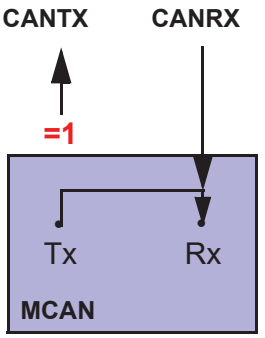
Note: The Restricted Operation Mode must not be combined with the Loop Back mode (internal or external).

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 66.7. 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

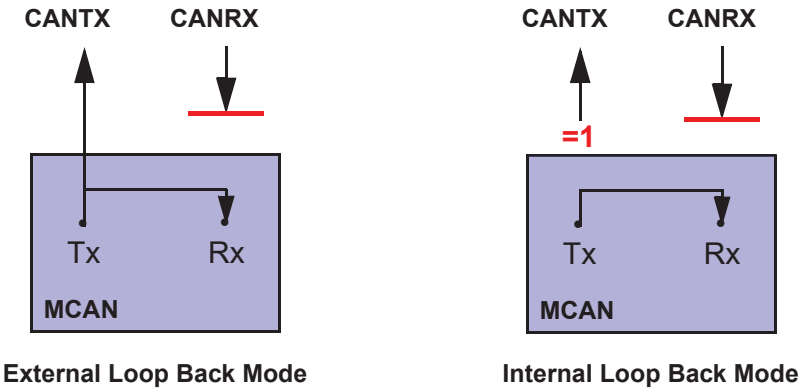
pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CANTX and CANRX to the MCAN in External Loop Back mode.

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 66.8. Pin Control in Loop Back Modes



66.5.2. Timestamp Generation

- **Internal Timestamp Generation**

For internal timestamp generation, the MCAN supplies a 16-bit wrap-around counter. A prescaler (MCAN_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 (SOF) reception/transmission, the counter value is captured and stored in the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

MCAN_TSCC.TSS can be used to capture the external 16-bit timebase vector input of the MCAN as timestamp instead of the internal 16-bit counter.

- **Timestamp Generation Using an External TSU**

A timestamping unit (TSU) for generation of 32-bit timestamps according to CiA 603 is connected to the MCAN TSU interface. External timestamping is enabled when MCAN_CCCR.UTSU = 1.

To filter for Sync messages, a Standard or Extended Filter element must be configured with the Sync message ID and bit SSYNC resp. ESYNC set to 1.

At the end of frame (EOF) reception/transmission, the timestamp is captured by the TSU. The number of the TSU Timestamp register which holds the captured timestamp is sent to the MCAN by a timestamp pointer (TSP) and stored in the related Rx Buffer/Rx FIFO element (R1B.RXTSP[2:0]) resp. Tx Event FIFO element (E1B.TXTSP[2:0]).

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', for example, 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 (for example, 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, for example, 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.

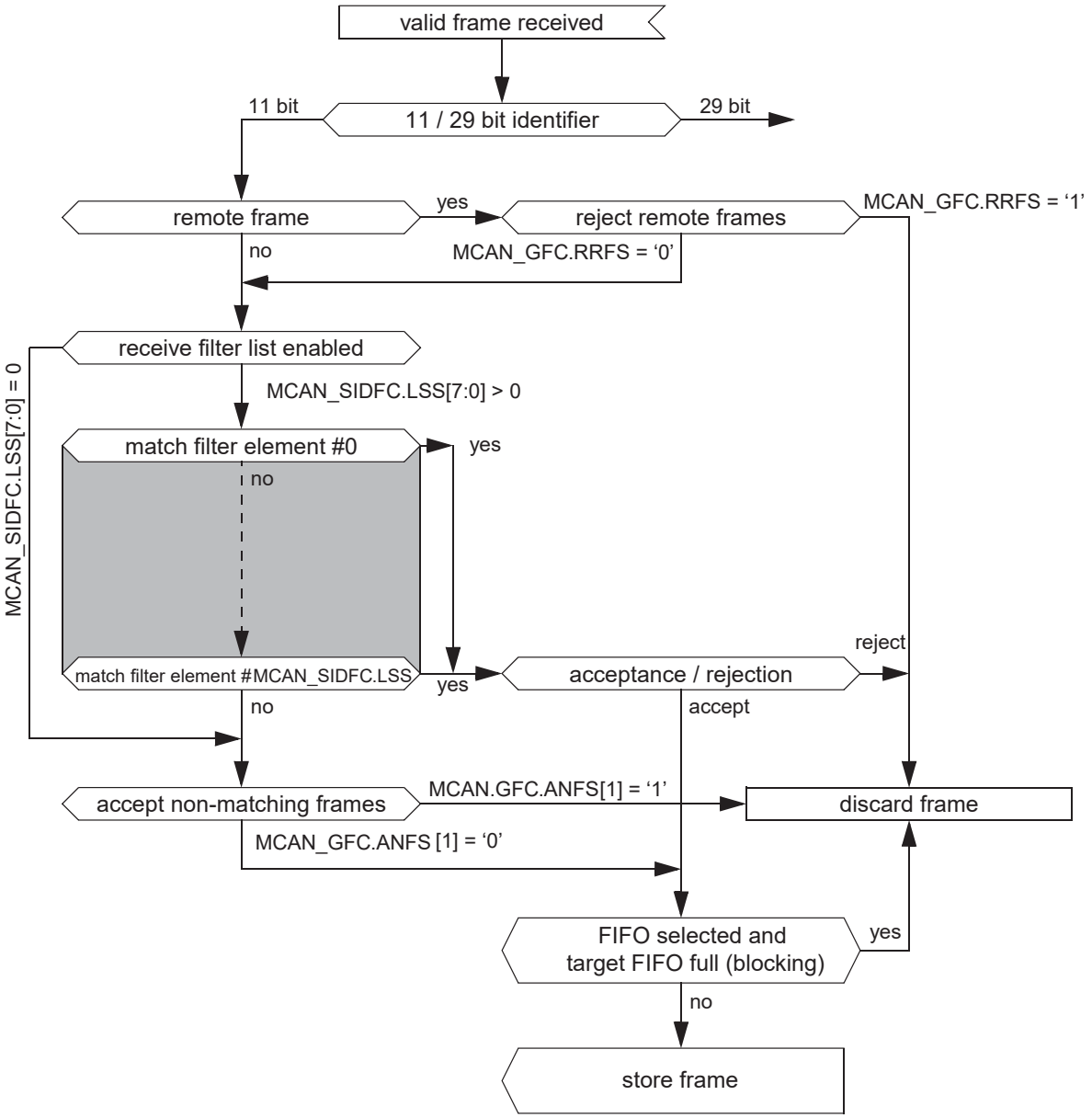
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

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 66.9. Standard Message ID Filter Path



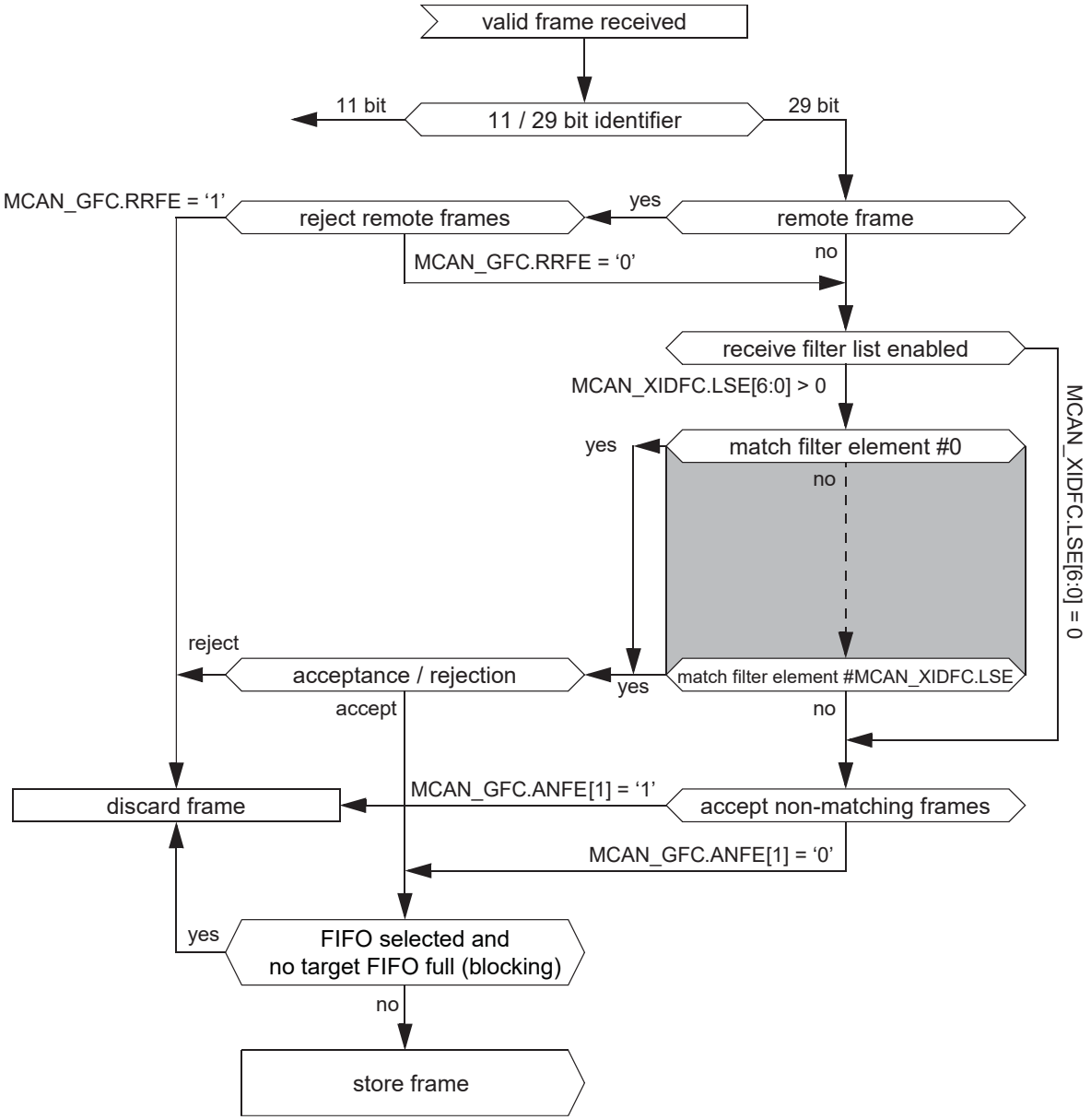
Extended Message ID Filtering

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 66.10. Extended Message ID Filter Path



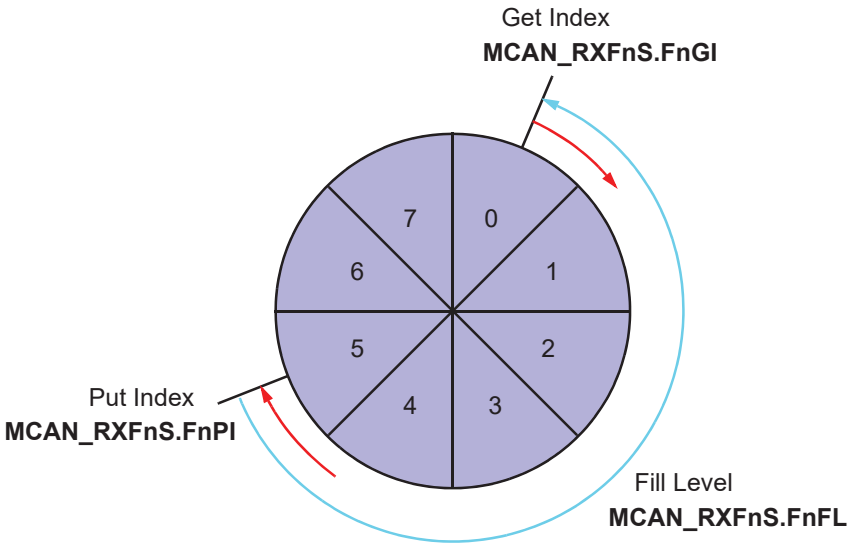
66.5.4.2.Rx FIFOs

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 66.11. 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.

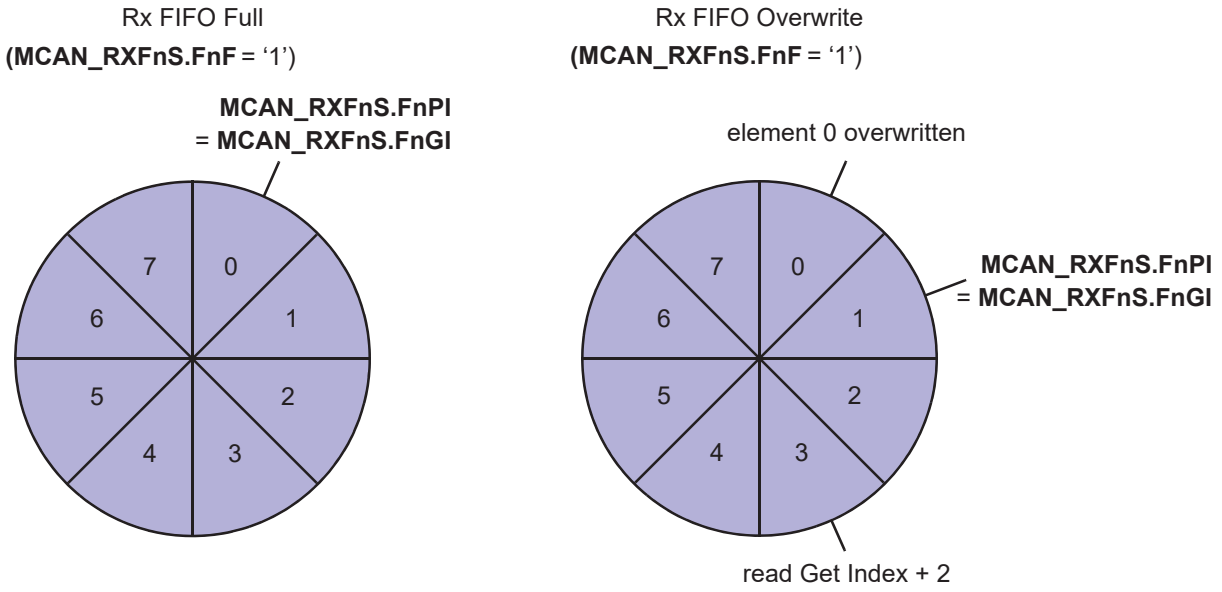
66.5.4.2.2.Rx FIFO Overwrite Mode

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 66.12. 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 (for example, #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

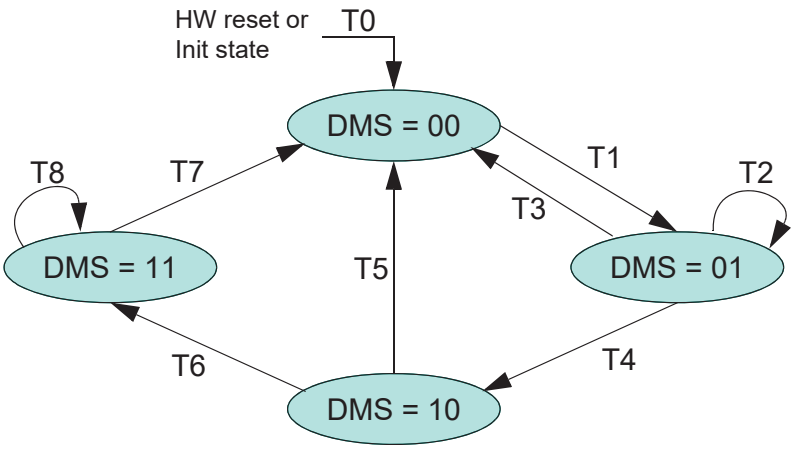
Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID debug message A	1	11 1101
1	ID debug message B	2	11 1110
2	ID debug message C	3	11 1111

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 66.13. 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, for example, 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.

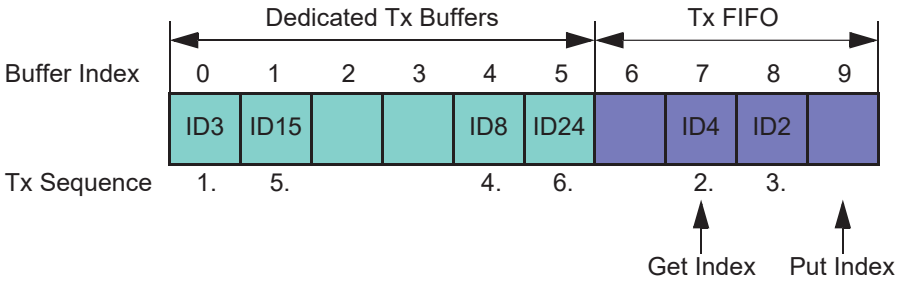
The application may use register MCAN_TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

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 66.14. Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO



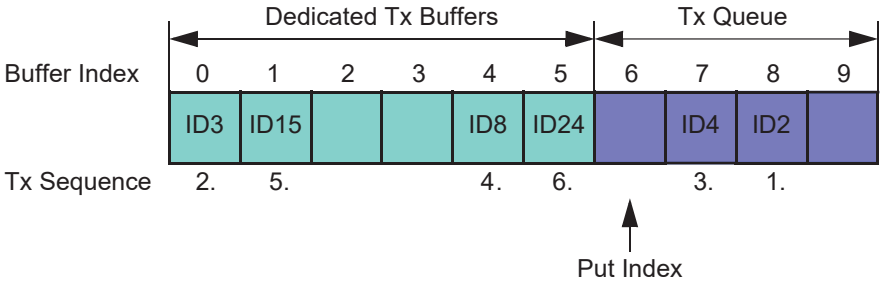
Tx prioritization:

- 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 66.15. 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.

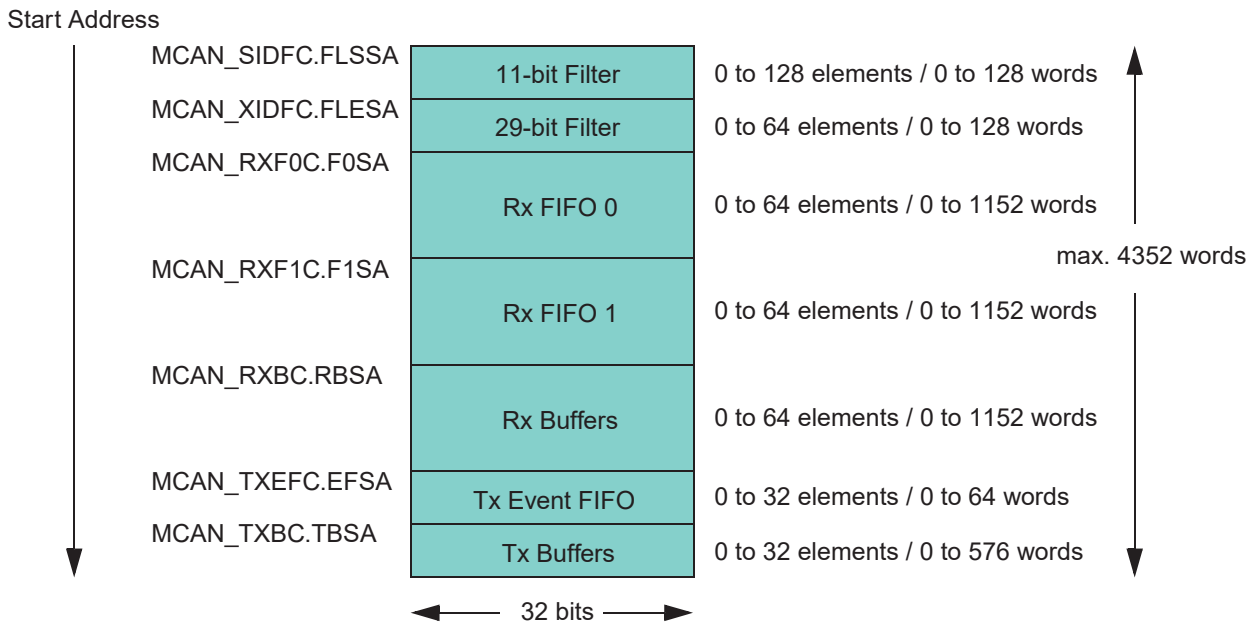
66.5.7. Message RAM

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 66.16. 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.

R1A: When no TSU is used (CCCR.UTSU = '0'), R1A.RXTS[15:0] holds the 16-bit timestamp generated by the MCAN internal timestamping logic.

R1B: When a TSU is used (CCCR.UTSU = '1') and when bit SSYNC/ESYNC of the matching filter element is set, R1B.TSC = '1' and R1B.RXTSP[3:0] holds the number of the TSU Timestamp register which holds the 32-bit timestamp captured by the TSU. Else R1B.TSC = '0' and R1B.RXTSP[3:0] is not valid.

Table 66.7. Rx Buffer and 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
R0	ESI	XTD	RTR	ID[28:0]																												
R1A	ANMF	FIDX[6:0]							–		FDF	BRS	DLC[3:0]			RXTS[15:0]																
R1B	ANMF	FIDX[6:0]							–		FDF	BRS	DLC[3:0]			–												TSC	RXTSP[3:0]			
R2	DB3[7:0]								DB2[7:0]							DB1[7:0]												DB0[7:0]				
R3	DB7[7:0]								DB6[7:0]							DB5[7:0]												DB4[7:0]				
...	..								...							...												...				
Rn	DBm[7:0]								DBm-1[7:0]DBM[7:0]							DBm-2[7:0]DBM[7:0]												DBm-3[7:0]DBM[7:0]				

- **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].

• **R1A/B 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.

• **R1A/B 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.

• **R1A/B Bit 21 FDF:** FD Format

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

• **R1A/B 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.

• **R1A/B 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.

• **R1A/B 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.

• **R1B Bits 3:0 RXTSP[3:0]: Rx Timestamp Pointer**

Number of the TSU Timestamp register (TS0..15) where the related timestamp is stored.

- **R1B Bit 4 TSC: Timestamp Captured**

0: No timestamp captured.

1: Timestamp captured and stored in TSU Timestamp register referenced by R1B.RXTSP.

- **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

	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
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC	TFCE	FDF	BRS	DLC[3:0]				MM[15:8]								reserved							
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
...	..								...								...								...							
Tn	DBm[7:0]								DBm-1[7:0]DBM[7:0]								DBm-2[7:0]DBM[7:0]								DBm-3[7:0]DBM[7:0]							

- **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 22 TSCE:** Timestamp Capture Enable for TSU

Only available when CCCR.UTSU = '1'. When this bit is set and the message is transmitted, a pulse is generated to signal the transmission of a Sync message to the Timestamping Unit (TSU) connected to the MCAN.

0: Timestamp Capture disabled

1: Timestamp Capture enabled

• **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.

• **T1 Bits 15:8 MM[15:8]:** Message Marker

High byte of Wide Message Marker, written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status. Available only when CCCR.WMM = '1' or when CCCR.UTSU = '1'.

- **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.

E1A: When CCCR.WMM = '0' and no TSU is used (CCCR.UTSU = '0'), E1A.TXTS[15:0] holds the 16-bit timestamp generated by the MCAN internal timestamping logic.

E1B: When 16-bit Message Markers are enabled (CCCR.WMM ='1') or when CCCR.UTSU = '1', E1B.MM[15:8] holds the upper 8 bit of the Wide Message Marker. When a TSU is used (CCCR.UTSU = '1') and when bit TSCE of the related Tx Buffer element is set, E1B.TSC = '1' and E1B.TXTSP[3:0] holds the number of the TSU Timestamp register which holds the 32-bit timestamp captured by the TSU. Else E1B.TSC = '0' and E1B.TXTSP[3:0] is not valid.

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]																												
E1A	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]			TXTS[15:0]																
E1B	MM[7:0]								ET[1:0]		FDF	BRS	DLC[3:0]			MM[15:8]							–				TSC	TXTSP[3:0]			

- **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].

• **E1A/B Bits 31:24 MM[7:0]:** Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

• **E1A/B 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

• **E1A/B Bit 21 FDF:** FD Format

0: Standard frame format.

1: CAN FD frame format (new DLC-coding and CRC).

• **E1A/B Bit 20 BRS:** Bit Rate Switch

0: Frame transmitted without bit rate switching.

1: Frame transmitted with bit rate switching.

• **E1A/B 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

• **E1A/B 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.

• **E1B Bit 4 TSC: Timestamp Captured**

0: No timestamp captured

1: Timestamp captured and stored in TSU Timestamp register referenced by E1B.TXTSP

• **E1B Bits 3:0 TXTSP[3:0]: Tx Timestamp Pointer**

Number of the TSU Timestamp register (TS0..15) where the related timestamp is stored.

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]											SSYNC	–				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

Standard Message ID Filter Element (continued)	
Value	**Description**
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.

• **Bit 15 SSYNC: Standard Sync Message**

Only evaluated when CCCR.UTSU = ‘1’. When this bit is set and a matching message is received, a pulse is generated to signal the reception of a Sync message to the Timestamping Unit (TSU) connected to the MCAN.

Value	Description
0	Timestamping for the matching Sync message disabled
1	Timestamping for the matching Sync message enabled

• **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]		ESYNC	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 Bit 29 ESYNC: Extended Sync Message**

Only evaluated when CCCR.UTSU = ‘1’. When this bit is set and a matching message is received, a pulse is generated to signal the reception of a Sync message to the Timestamping Unit (TSU) connected to the MCAN.

Value	Description
0	Timestamping for the matching Sync message disabled
1	Timestamping for the matching Sync message enabled

• **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			SVAL	TXBNS[4:0]				
		15:8			PVAL	TXBNP[4:0]				
		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	WMM	UTSU	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]							
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]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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
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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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
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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
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]				
0xFC ... 0x0163	Reserved									
0x0164	MCAN_TSU_TSCFG	31:24								
		23:16								
		15:8	TBPRE[7:0]							
		7:0						SCP	TBCS	TSUE
0x0168	MCAN_TSU_TSS1	31:24	TSL[15:8]							
		23:16	TSL[7:0]							
		15:8	TSN[15:8]							
		7:0	TSN[7:0]							
0x016C	MCAN_TSU_TSS2	31:24								
		23:16								
		15:8	ITBG[1:0]		NTSG[1:0]					
		7:0					TSP[3:0]			
0x0170	MCAN_TSU_TS0	31:24	TS[31:24]							
		23:16	TS[23:16]							
		15:8	TS[15:8]							
		7:0	TS[7:0]							
...										
0x01AC	MCAN_TSU_TS15	31:24	TS[31:24]							
		23:16	TS[23:16]							
		15:8	TS[15:8]							
		7:0	TS[7:0]							
0x01B0	MCAN_TSU_ATB	31:24	TB[31:24]							
		23:16	TB[23:16]							
		15:8	TB[15:8]							
		7:0	TB[7:0]							

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

To enable write access to the Test register, set 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 bit 7, MCAN_TEST.RX, is undefined. Bits 4 to 6 read 0. Bit 7 reads 1 when operating in CAN environments.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SVAL	TXBNS[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
			PVAL	TXBNP[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
	RX	TX[1:0]		LBCK				
Access	R	R/W	R/W	R/W				
Reset	x	0	0	0				

Bit 21 – SVAL Started Valid
Monitors the actual value of pin CANRX.
The reset value for this bit is undefined.

Value	Description
0 (DISABLED)	TXBNS value is not valid.
1 (ENABLED)	TXBNS value is valid.

Bits 20:16 – TXBNS[4:0] TX Buffer Number Started
In TX buffer, number of messages whose transmission was started last. Valid when SVAL is set. Valid values are 0 to 31.

Bit 13 – PVAL Prepared Valid
In TX buffer, number of messages whose transmission was started last. Valid when SVAL is set. Valid values are 0 to 31.

Value	Description
0 (DISABLED)	TXBNP value is not valid.
1 (ENABLED)	TXBNP value is valid.

Bits 12:8 – TXBNP[4:0] TX Buffer Number Prepared

Five digits. In TX buffer, number of messages that are ready for transmission. Valid when PVAL is set. Valid values are 0 to 31.

Bit 7 – RX Receive Pin

Monitors the actual value of pin CANRX.
The reset value for this bit is undefined. The bit reads 1 when operating in CAN environments.

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	WMM	UTSU	BRSE	FDOE
Access	R/W	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
	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 11 – WMM Wide Message Marker (write protection)
0 (DISABLED): 8-bit message marker is used.
1 (ENABLED): 16-bit message marker is used, replacing 16-bit timestamps in TX Event FIFO.

Bit 10 – UTSU Use Timestamping Unit (write protection)
When UTSU is set, 16-bit Wide Message Marker (WMM) is also enabled regardless of the value of WMM.
0 (DISABLED): Internal timestamping.
1 (ENABLED): External timestamping by TSU.

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.

Value	Name	Description
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

Value	Name	Description
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.

66.6.46. MCAN TSU Timestamp Configuration

Name: MCAN_TSU_TSCFG
Offset: 0x164
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
	TBPRE[7:0]							
Access	R/W	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
						SCP	TBCS	TSUE
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 15:8 – TBPRE[7:0] Timebase Prescaler

The value by which the oscillator frequency is divided to generate the timebase counter clock. Valid values for the timebase prescaler are 0 to 255. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Affects only the TSU internal timebase.

Bit 2 – SCP Select Capturing Position

Value	Description
0	Timestamp captured at EOF
1	Timestamp captured at SOF

Bit 1 – TBCS Timebase Counter Select

Value	Description
0	Timestamp value captured from internal timebase counter (32 bits)
1	Timestamp value captured from external timebase counter (32 bits)

Bit 0 – TSUE Timestamp Unit Enable

Value	Description
0	TSU disabled
1	TSU enabled

66.6.47. MCAN TSU Timestamp Status 1

Name: MCAN_TSU_TSS1
Offset: 0x168
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TSL[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
	TSL[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
	TSN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

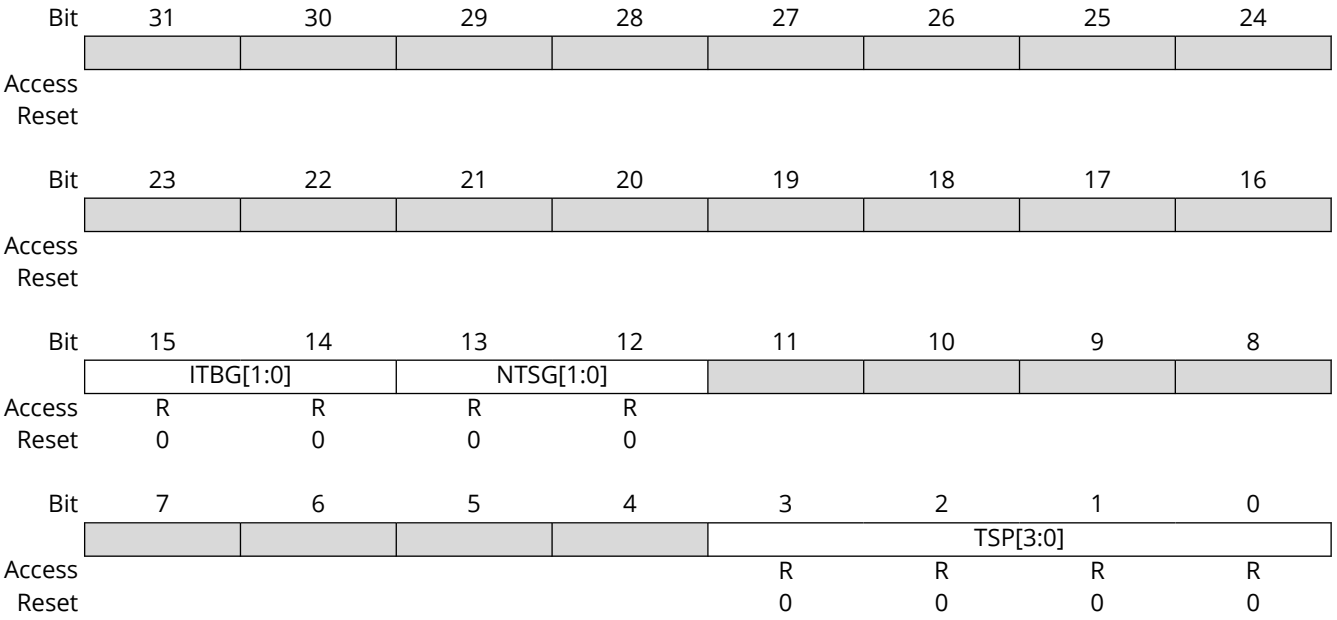
Bit	7	6	5	4	3	2	1	0
	TSN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – TSL[15:0] Timestamp Lost
Each timestamp register (TS0-TS15) is assigned one bit. The bits are set when the timestamp stored in the related timestamp register was overwritten before it was read. Reading a timestamp register resets the related bit.

Bits 15:0 – TSN[15:0] Timestamp New
Each timestamp register (TS0-TS15) is assigned one bit. The bits are set when a timestamp was stored in the related timestamp register. Reading a timestamp register resets the related bit.

66.6.48. MCAN TSU Timestamp Status 2

Name: MCAN_TSU_TSS2
Offset: 0x16C
Reset: 0x000A0000
Property: Read-only



Bits 15:14 – ITBG[1:0] Internal Timebase and SOF Select Generic
Constant value 11 (SOF option, internal timebase)

Bits 13:12 – NTSG[1:0] Number of Timestamps Generic
Constant value 10 (16 timestamp registers TS0-TS15).

Bits 3:0 – TSP[3:0] Timestamp Pointer
The timestamp pointer is incremented by one each time a timestamp is captured, from its minimum value (0) to 15.

66.6.49. MCAN TSU Timestamp

Name: MCAN_TSU_TSx
Offset: 0x0170 + x*0x04 [x=0..15]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TS[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
	TS[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
	TS[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
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS

66.6.50. MCAN TSU Actual Timebase

Name: MCAN_TSU_ATB
Offset: 0x1B0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	TB[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
	TB[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
	TB[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
	TB[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TB[31:0] Timebase for Timestamp Generation

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
- Register Write Protection
- Safety/Security Reports

67.3. Block Diagram

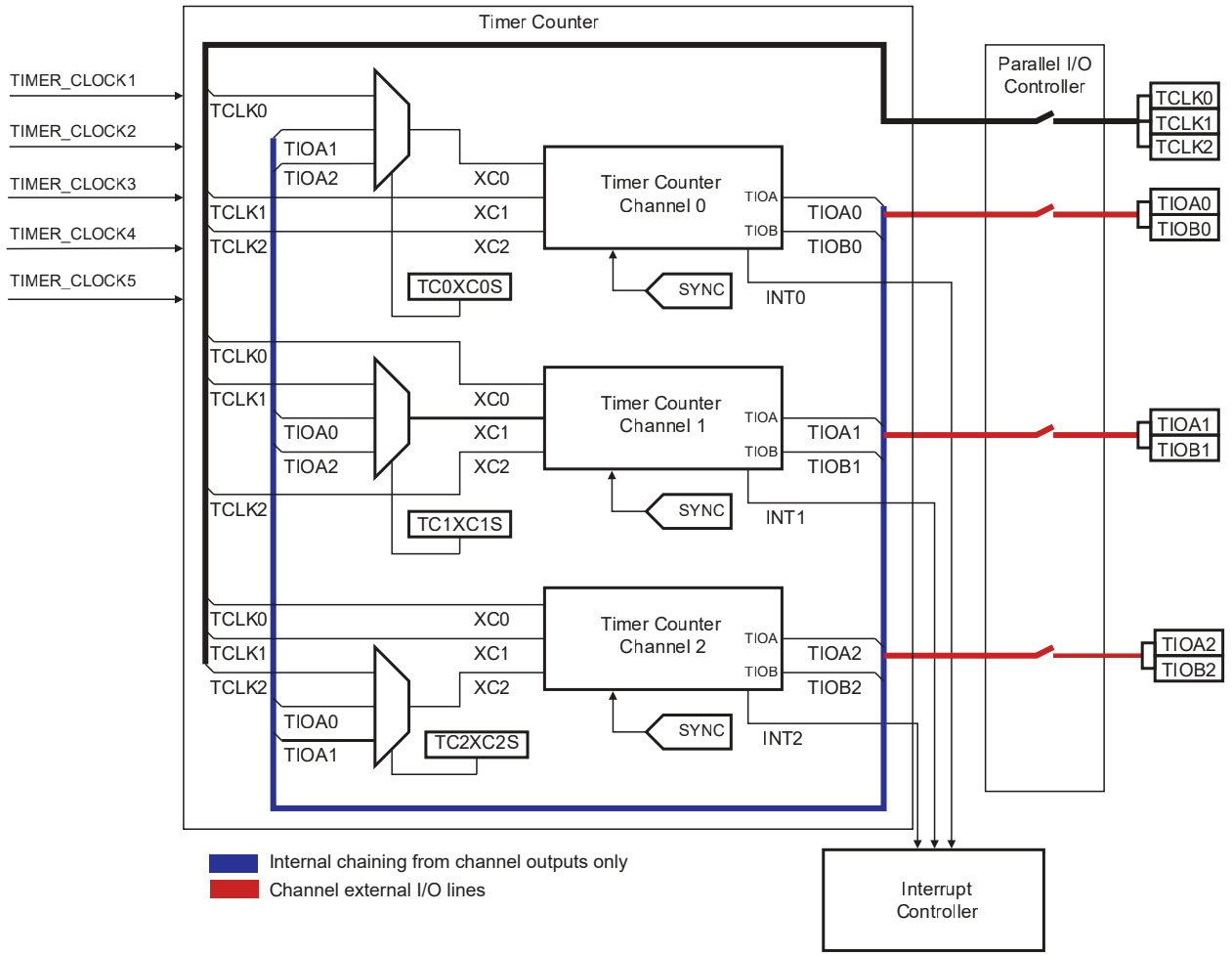
Table 67.1. Timer Counter Clock Assignment

Name	Definition
TIMER_CLOCK1	GCLK [17], GCLK[45]
TIMER_CLOCK2	MCK/8
TIMER_CLOCK3	MCK/32
TIMER_CLOCK4	MCK/128
TIMER_CLOCK5 (1)	MD_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



Notes:

1. The above figure provides pin names of a first instance of a Timer Counter module (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external I/O pins of a second Timer Counter module (i.e., instance TC1).
2. QDEC connections are detailed in Figure 67.17.

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
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 (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external I/O pins of a second Timer Counter block (i.e., instance TC1).

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.

67.5.3. Interrupt Sources

The TC has an interrupt line 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.

67.6. 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 [17], GCLK[45], MCK/8, MCK/32, MCK/128, MD_SLCK

This selection is made by the TCCLKS bits in the Channel Mode register (TC_CMRx).

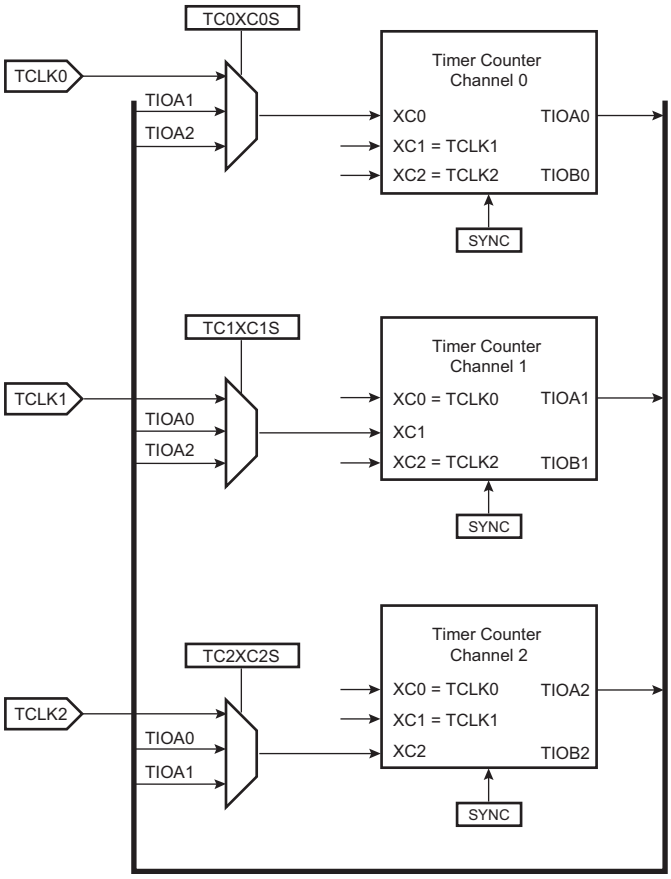
The selected clock can be inverted with TC_CMRx.CLKI. This allows counting on the opposite edges of the clock.

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:

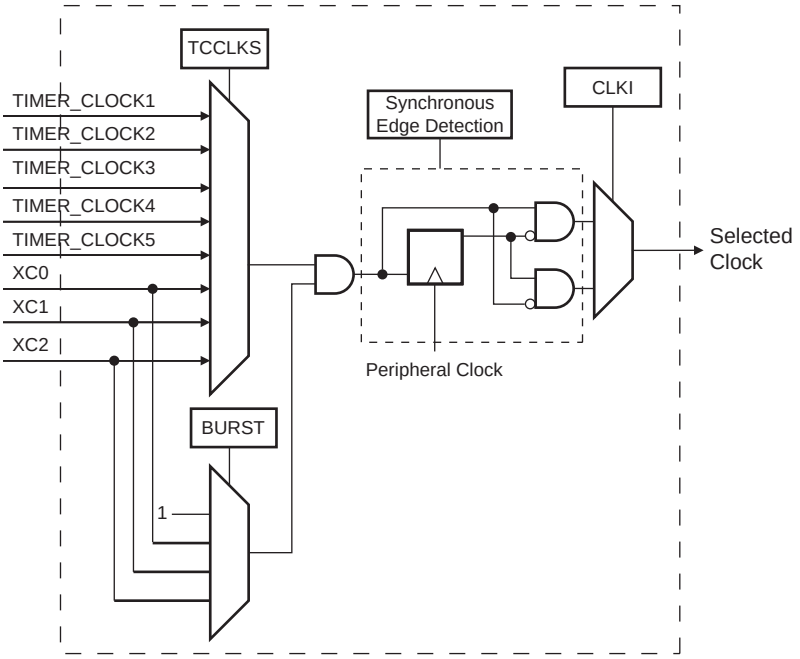
1. In Waveform mode, to chain two timers, it is mandatory to initialize some parameters:
   - Configure TIOx outputs to 1 or 0 by writing the required value to TC_CMRx.ASWTRG.
   - Bit TC_BCR.SYNC must be written to 1 to start the channels at the same time.
2. In all cases, if an external clock or asynchronous 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



Note: The above figure provides pin names of a first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, “TCLK3-TCLK5”, "TIOA3-TIOA5” and "TIOB3-TIOB5” are the external I/O pins of a second Timer Counter block (i.e., instance TC1).

Figure 67.3. Clock Selection

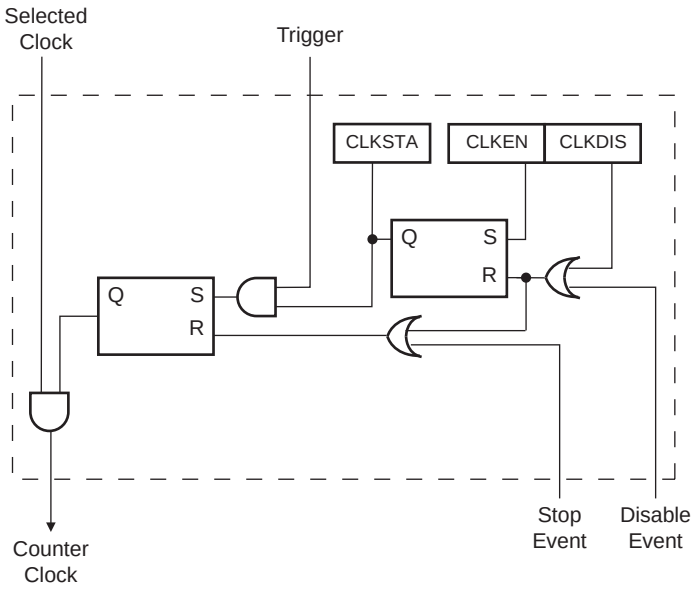


67.6.4. Clock Control

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

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.

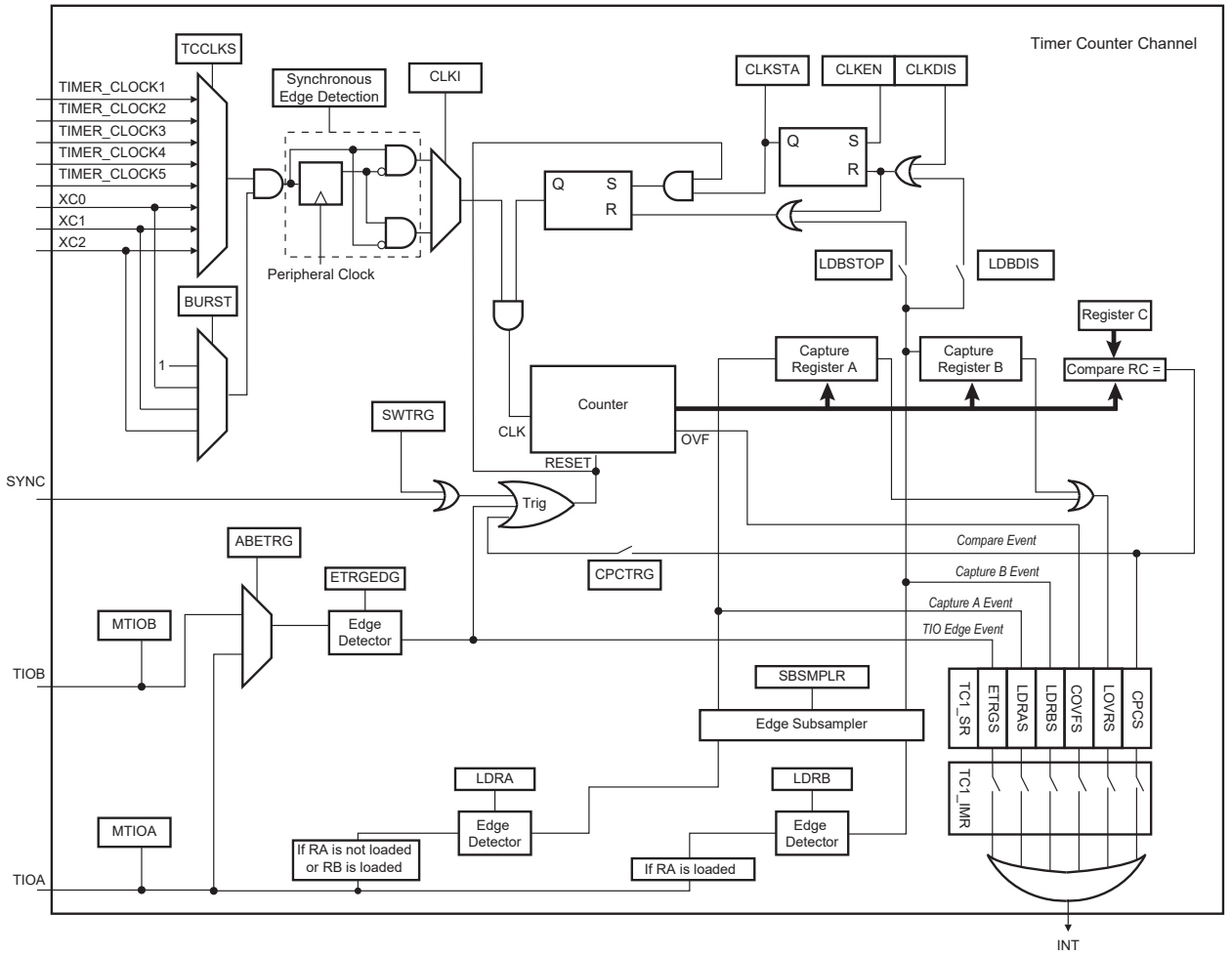
If an external trigger event is used, the duration of the pulses must be longer than the peripheral clock period to be detected.

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 67.5. 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.

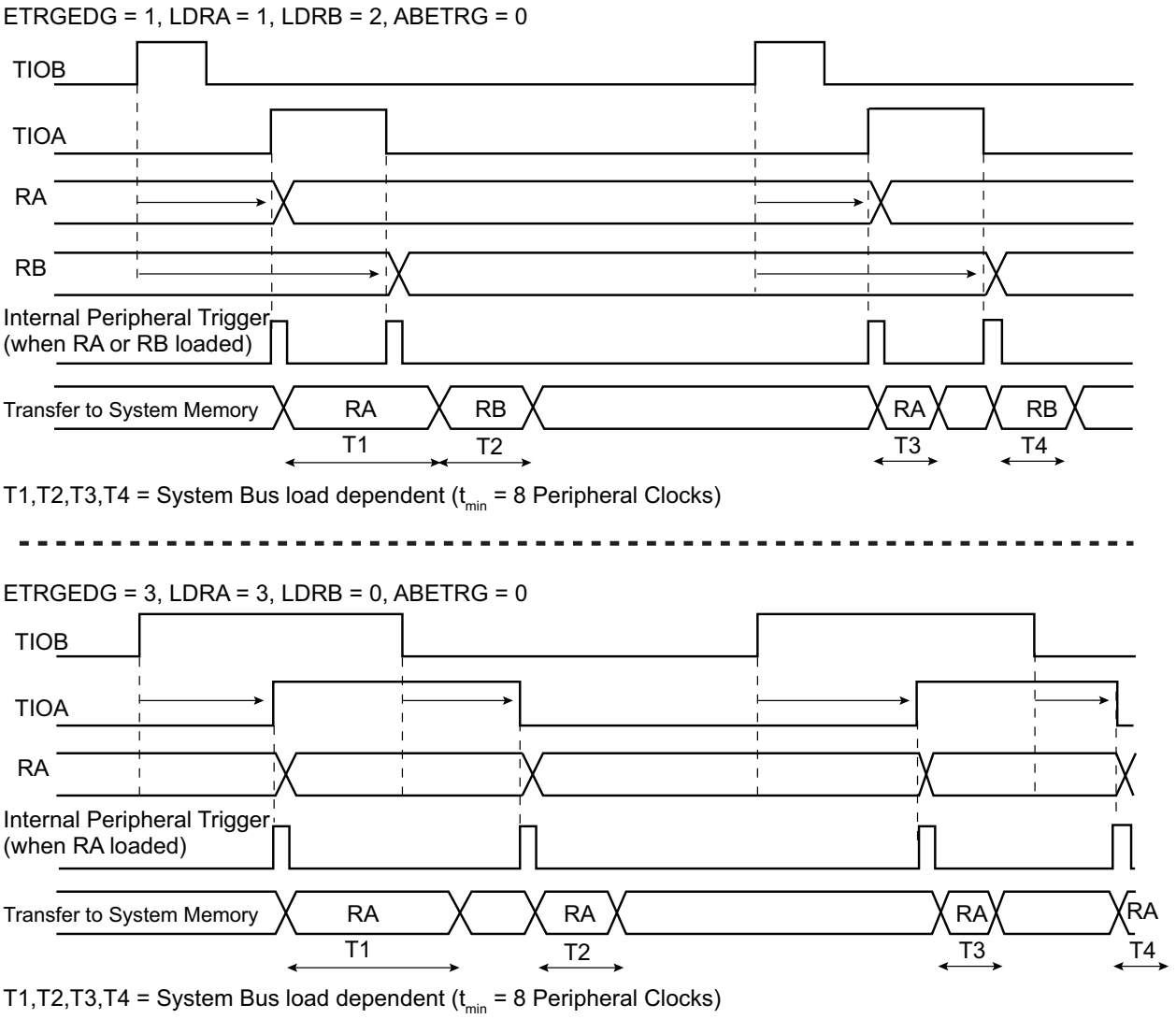
When DMA is used, the Register AB (TC_RAB) address must be configured as source address of the transfer. TC_RAB provides the next unread value from TC_RA and TC_RB. It may be read by the DMA after a request has been triggered upon loading TC_RA or TC_RB.

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 67.6. 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.

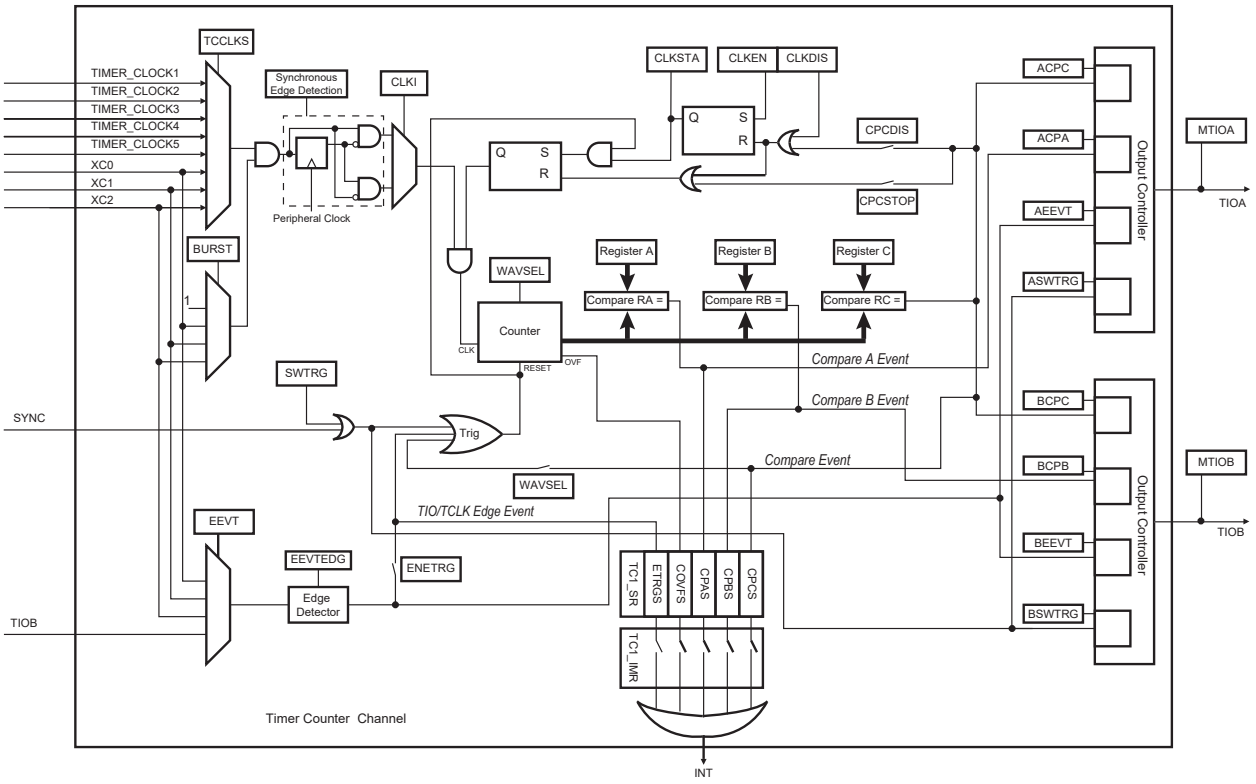
67.6.12. Waveform Selection

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 67.7. 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 67.8. WAVSEL = 00 without Trigger

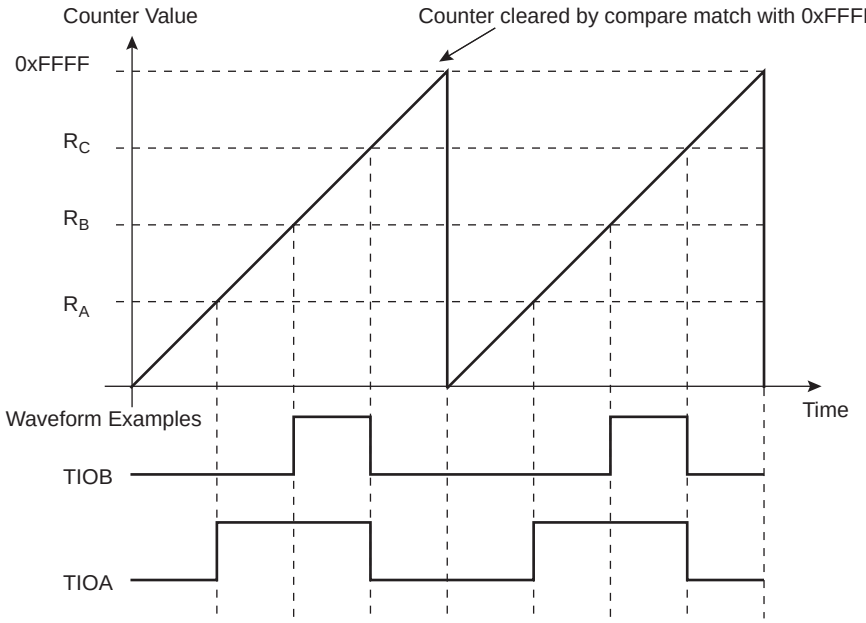
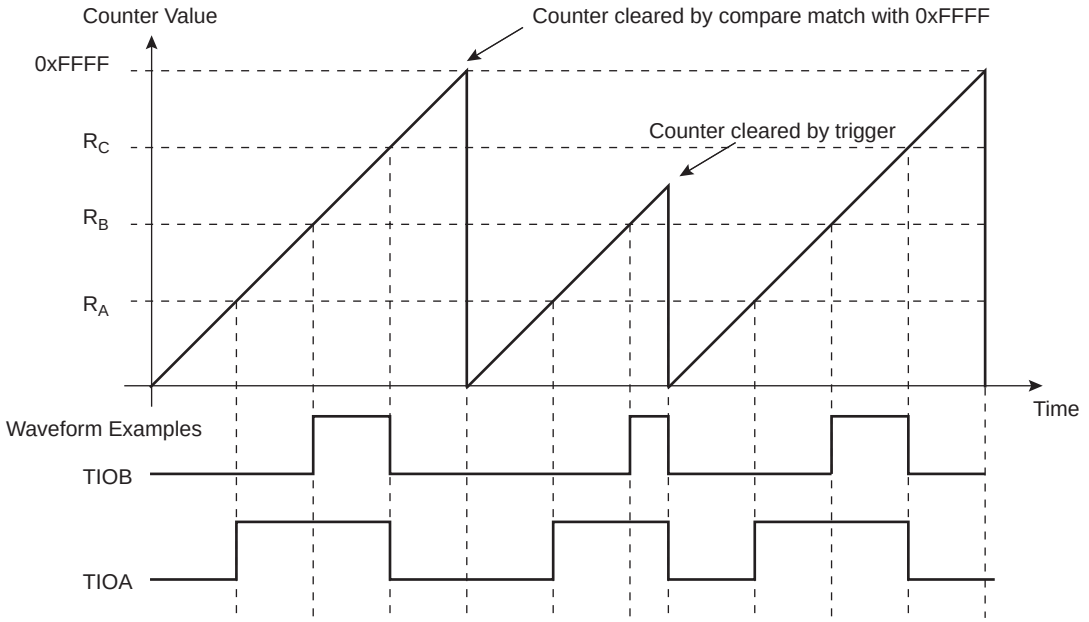


Figure 67.9. WAVSEL = 00 with Trigger



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 67.10. WAVSEL = 10 without Trigger

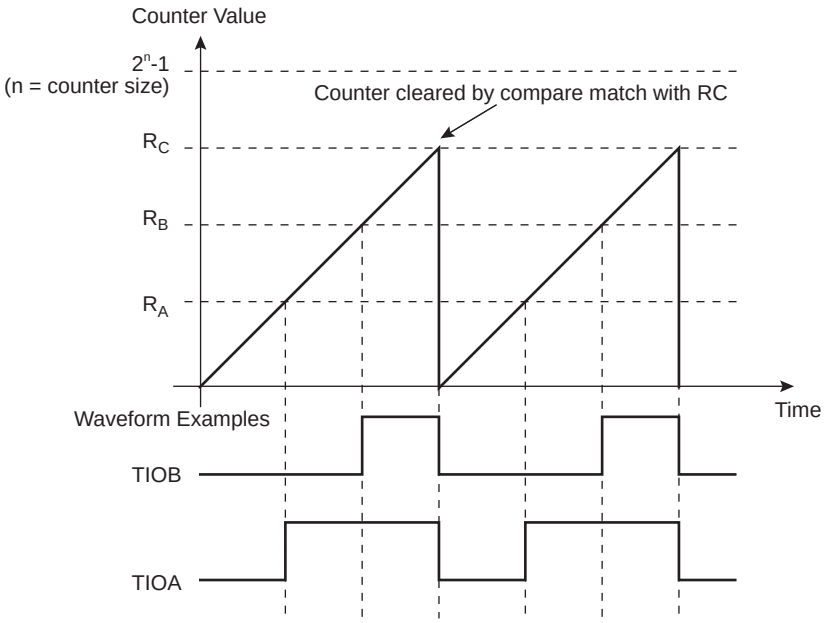
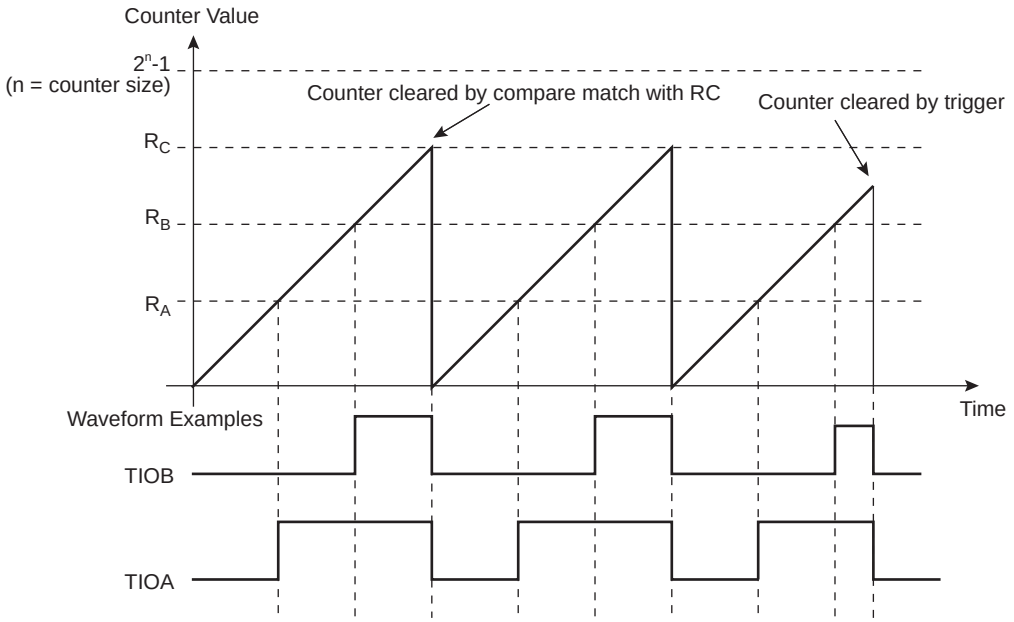


Figure 67.11. WAVSEL = 10 with Trigger



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 67.12. WAVSEL = 01 without Trigger

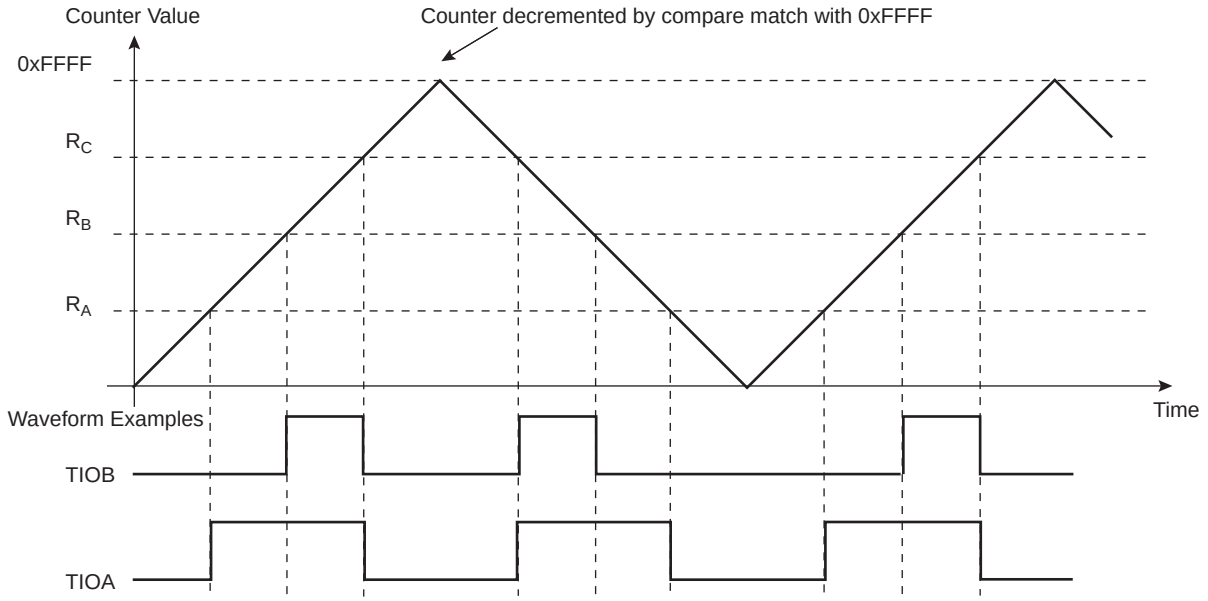
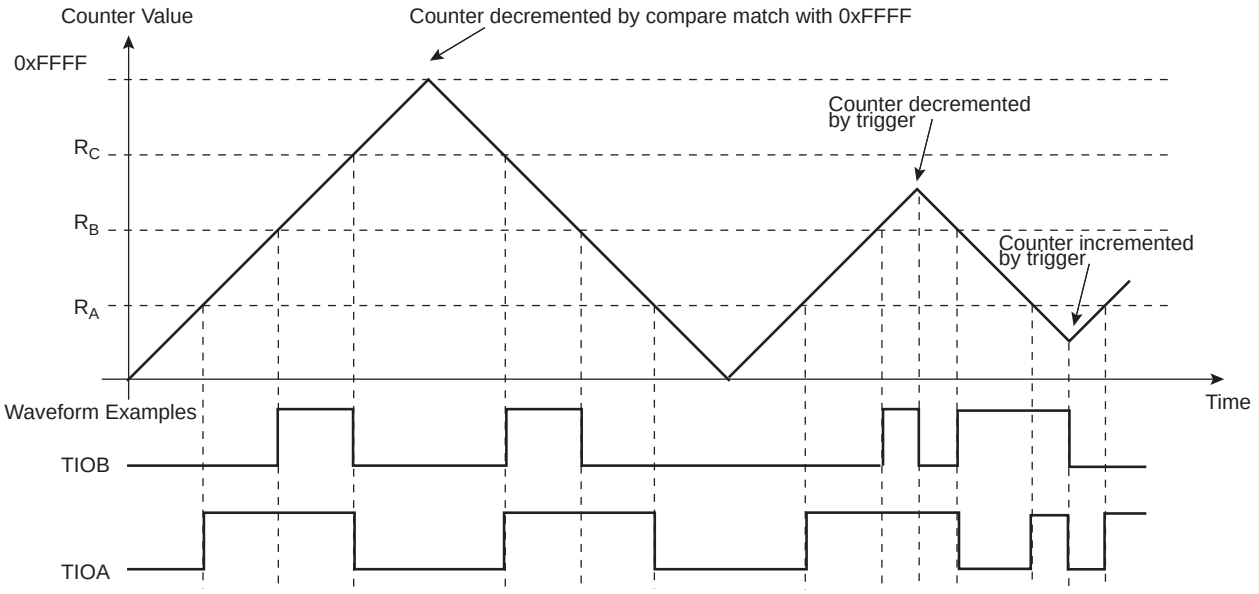


Figure 67.13. WAVSEL = 01 with Trigger



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 67.14. WAVSEL = 11 without Trigger

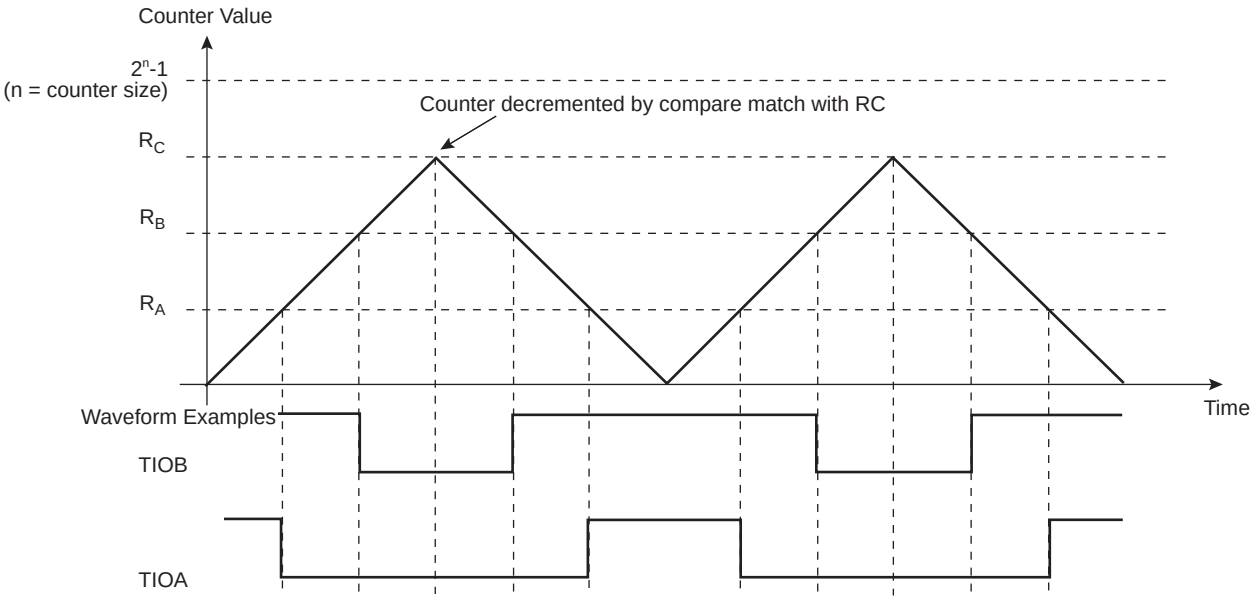
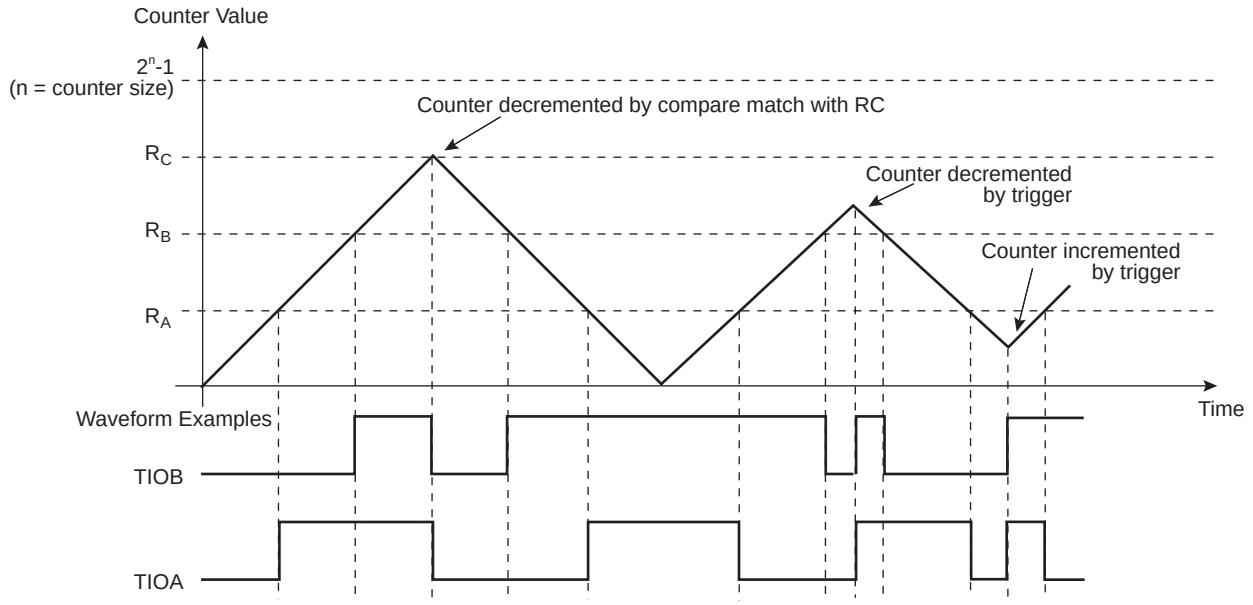


Figure 67.15. WAVSEL = 11 with Trigger



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

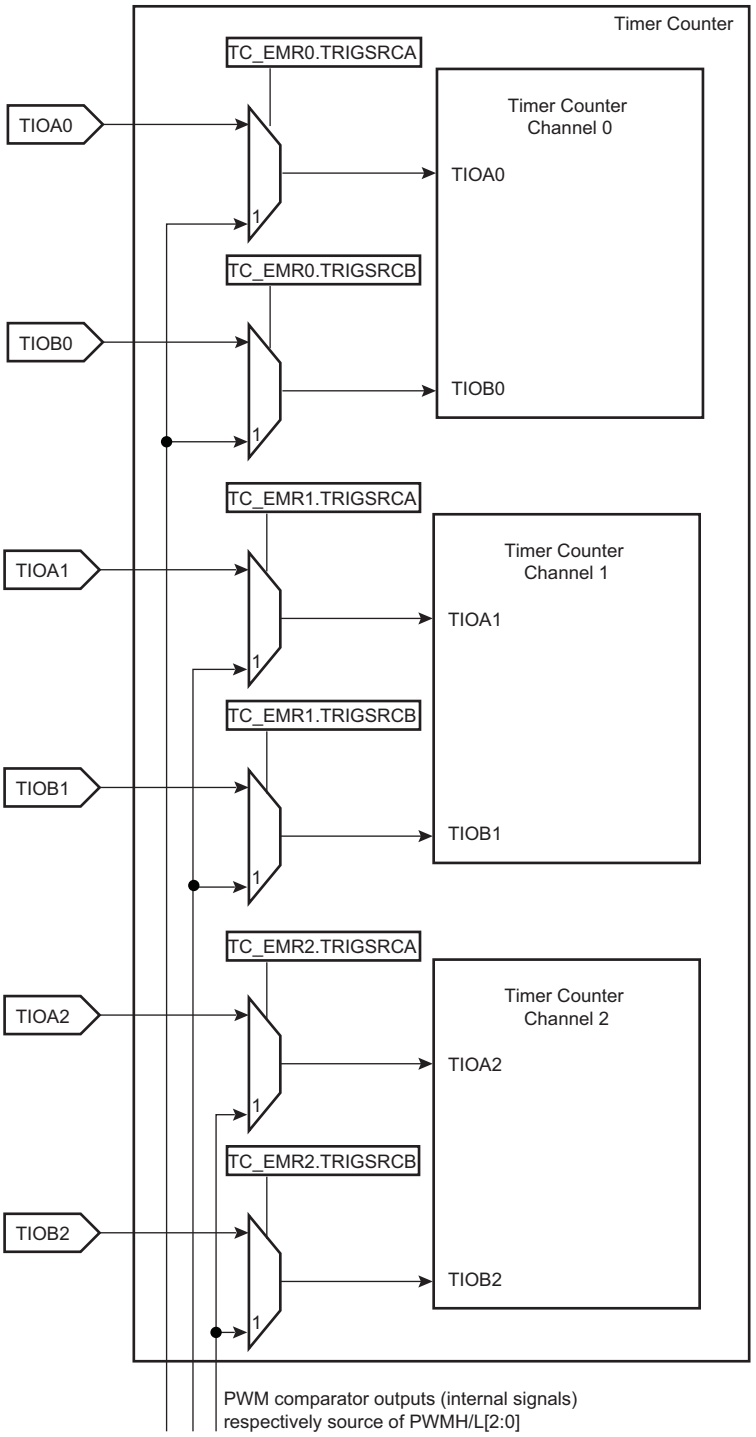
The inputs TIOAx/TIOBx can be bypassed, and thus channel trigger/capture events can be directly driven by the independent PWM module.

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 67.16. Synchronization with PWM



Note: This figure provides pin names of the first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, "TIOA3-TIOA5" and "TIOB3-TIOB5" are the external IO pins of a second Timer Counter block (i.e., instance TC1).

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.

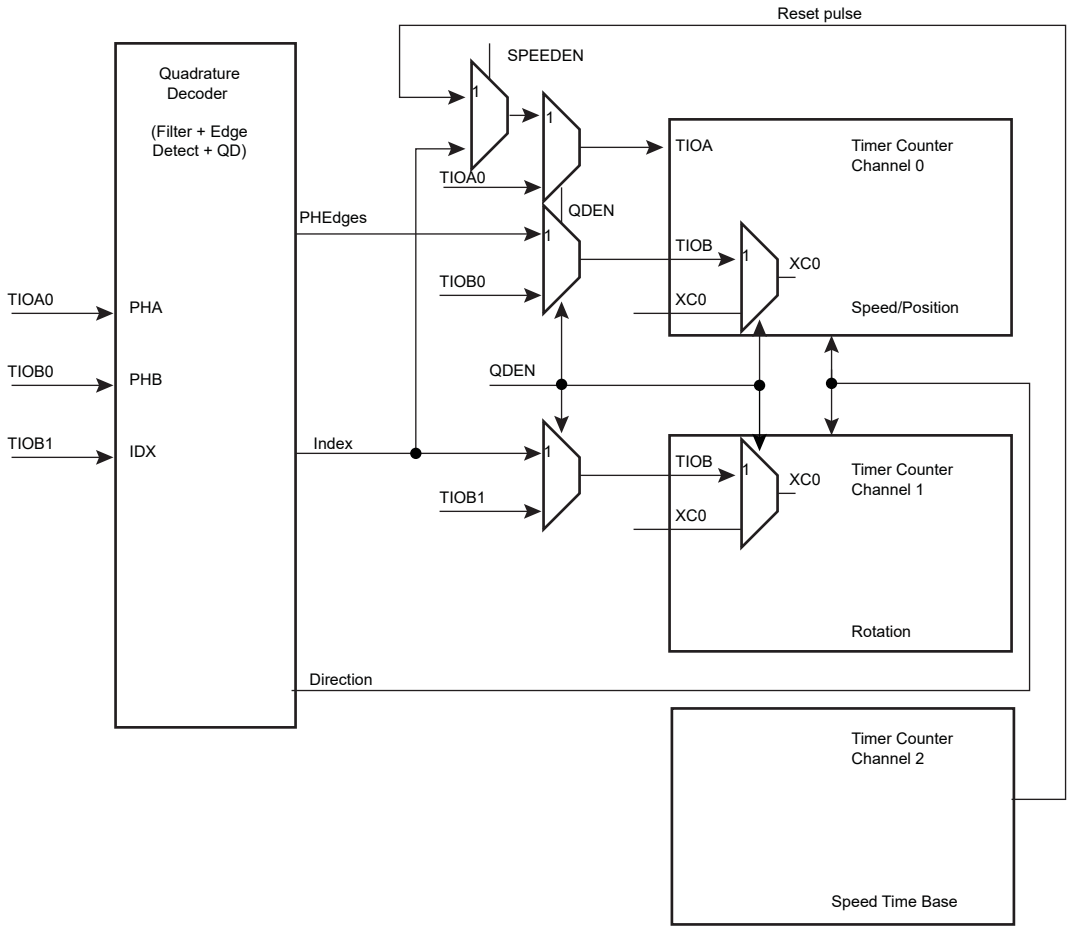
In Speed mode, position cannot be measured but revolution can be measured.

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 67.17. Predefined Connection of the Quadrature Decoder with Timer Counters



Note: This figure provides pin names of the first instance of a Timer Counter block (i.e., instance TC0). For any subsequent instances, the signal numbering increments. For example, "TIOA3-TIOA5" and "TIOB3-TIOB5" are the external IO pins of a second Timer Counter block (i.e., instance TC1).

67.6.16.2.Input Preprocessing

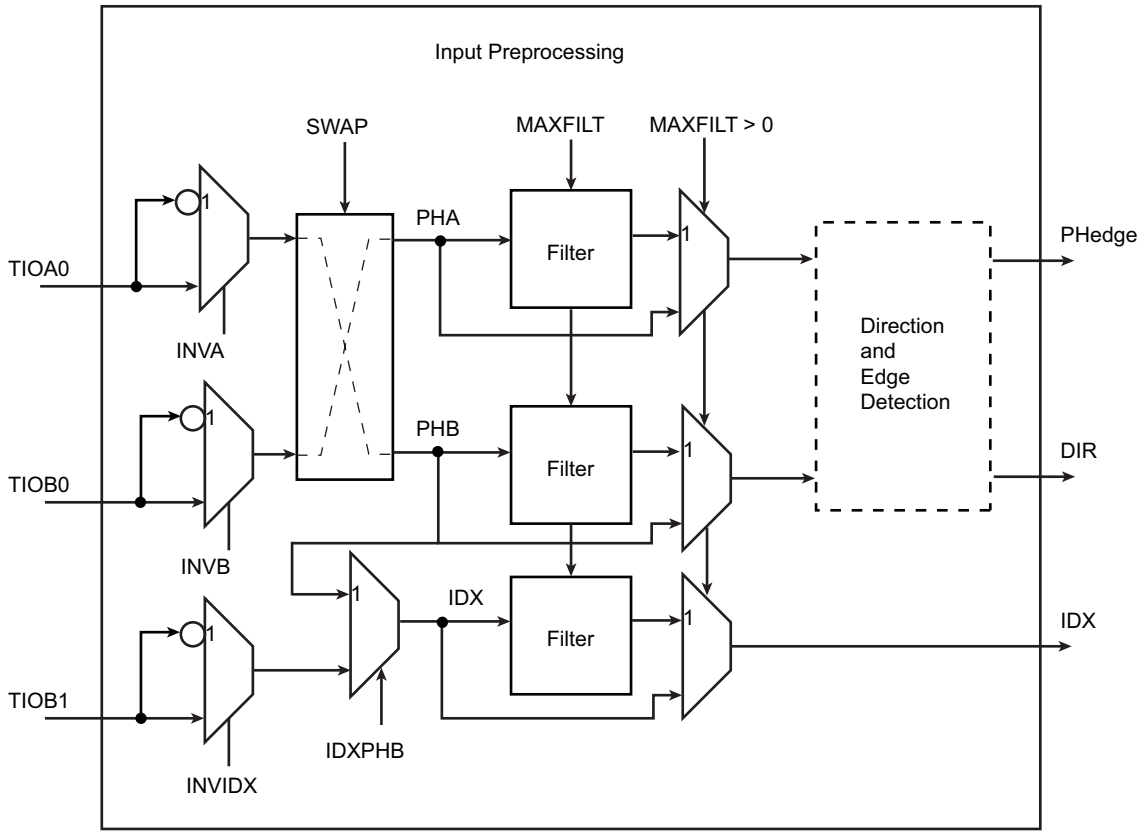
Input preprocessing consists of capabilities to take into account rotary sensor factors such as polarities and phase definition followed by configurable digital filtering.

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.

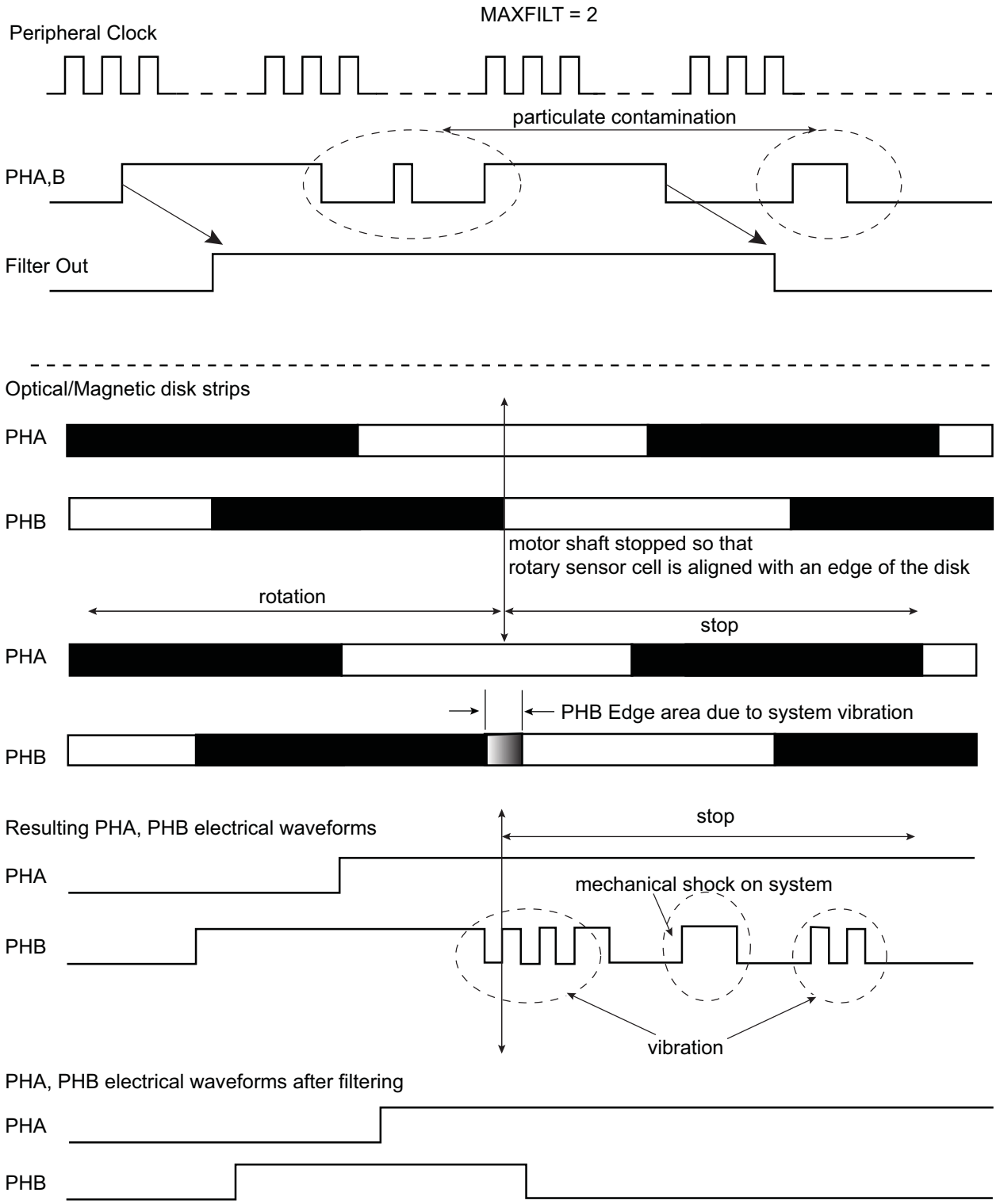
Figure 67.18. Input Stage



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 67.19. Filtering Examples



67.6.16.3.Direction Status and Change Detection

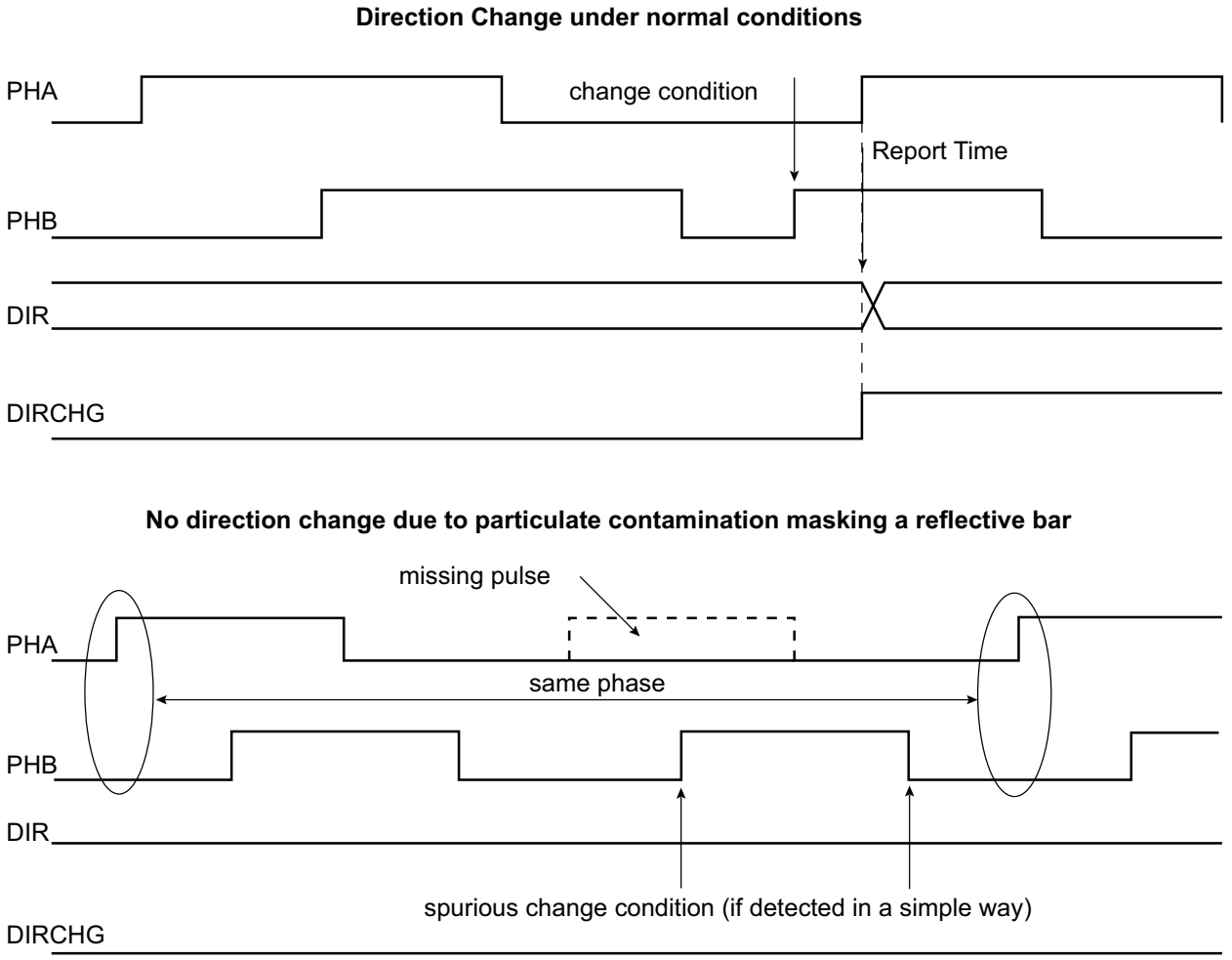
After filtering, the quadrature signals are analyzed to extract the rotation direction and edges of the two quadrature signals detected in order to be counted by TC logic downstream.

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.

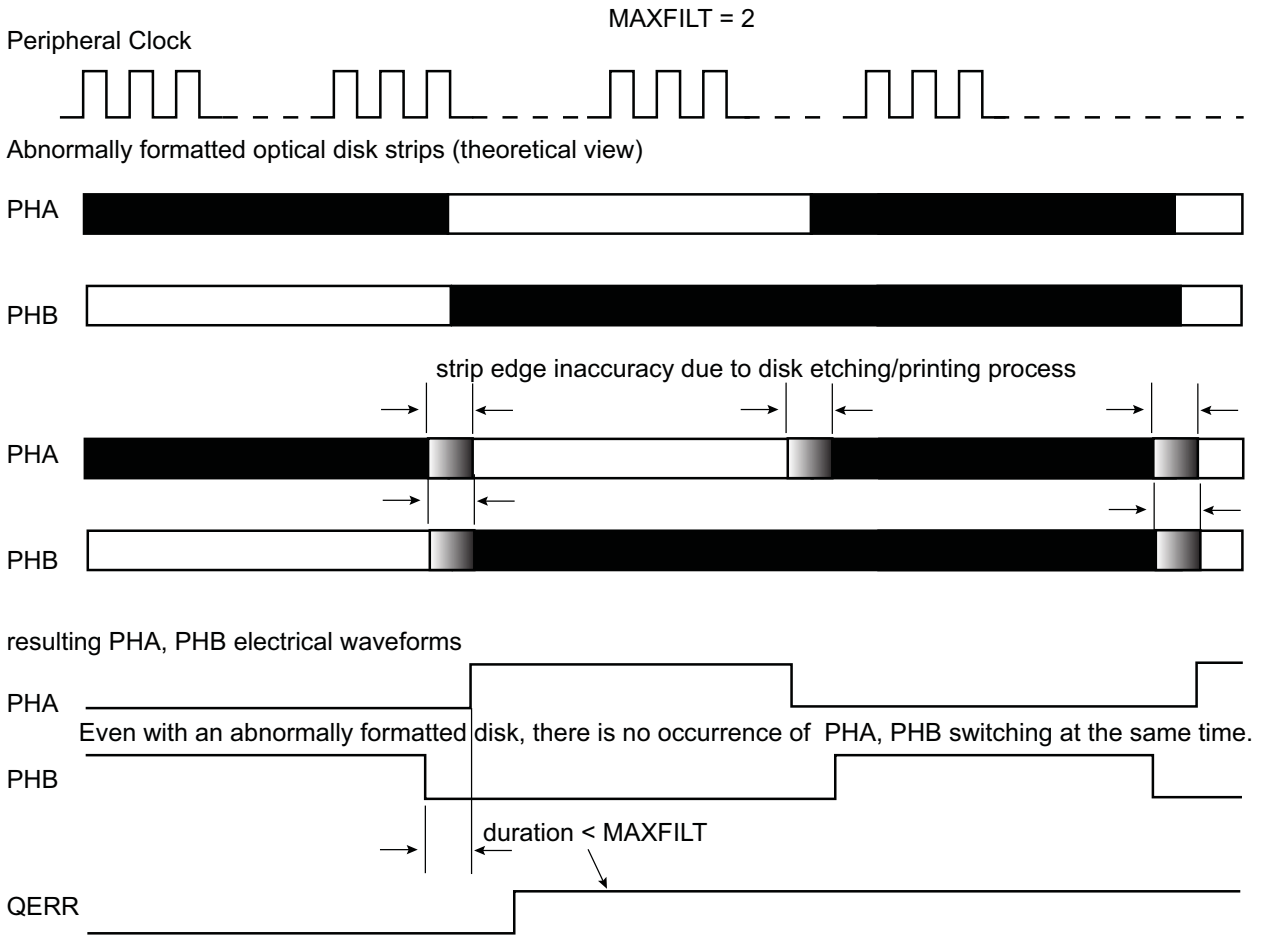
Figure 67.20. Rotation Change Detection



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 67.21. 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 (ex: if the 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. For example, 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.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

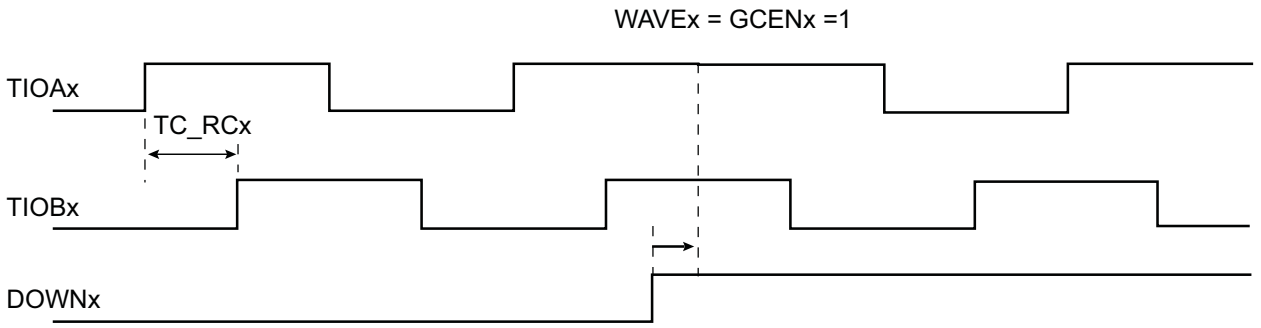
Each channel can be independently configured to generate a 2-bit Gray count waveform on corresponding TIOAx, TIOBx outputs by means of TC_SMMRx.GCEN.

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 67.22. 2-bit Gray Up/Down Counter



67.6.18. 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 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

67.6.19. 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]	

Register Summary (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

Register Summary (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

Register Summary (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							MAXFILT[5:4]	
		23:16	MAXFILT[3:0]						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		QERR	DIRCHG	IDX
0xCC	TC_QIDR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD0	TC_QIMR	31:24								
		23:16								
		15:8								
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD4	TC_QISR	31:24								
		23:16								
		15:8								DIR
		7:0	FMP	FIDX	FPHB	FPHA		QERR	DIRCHG	IDX
0xD8 ... 0xDB	Reserved									
0xDC	TC_QSR	31:24								
		23:16								
		15:8								DIR
		7:0								

Register Summary (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 [17], GCLK[45] 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)

Value	Name	Description
4	TIMER_CLOCK5	Clock selected: internal MD_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.

Value	Description
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 [17], GCLK[45] 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 MD_SLCK clock signal (from PMC)

Value	Name	Description
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.

Value	Description
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

Bit	31	30	29	28	27	26	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.TIOA0, TC1.TIOA2: The trigger/capture input A is driven internally by PWMx. For TC1.TIOA1: The trigger/capture input A 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
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.
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.

The External Clock Signal x Selection (TCxXCxS) bit field mentions pin names of the first Timer Counter module (TC0). For any subsequent instances, the signal numbering increments. For example, "TCLK3-TCLK5", "TIOA3-TIOA5" and "TIOB3-TIOB5" are the external I/O pins of the second Timer Counter module (TC1).

Bit	31	30	29	28	27	26	25	24
							MAXFILT[5:4]	
Access							R/W	R/W
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	MAXFILT[3:0]						IDXPHB	SWAP
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
	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 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 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 (XC2) Selection

See Clock Selection for more details.

Value	Name	Description
0	TCLK2	Signal connected to XC2: TCLK2
1	–	Reserved
2	TIOA0	Signal connected to XC2: internal TIOA0 for chaining
3	TIOA1	Signal connected to XC2: internal TIOA1 for chaining

Bits 3:2 – TC1XC1S[1:0] External Clock Signal 1 (XC1) Selection
See Clock Selection for more details.

Value	Name	Description
0	TCLK1	Signal connected to XC1: TCLK1
1	–	Reserved
2	TIOA0	Signal connected to XC1: internal TIOA0 for chaining
3	TIOA2	Signal connected to XC1: internal TIOA2 for chaining

Bits 1:0 – TC0XC0S[1:0] External Clock Signal 0 (XC0) Selection
See Clock Selection for more details.

Value	Name	Description
0	TCLK0	Signal connected to XC0: TCLK0
1	–	Reserved
2	TIOA1	Signal connected to XC0: internal TIOA1 for chaining
3	TIOA2	Signal connected to XC0: internal TIOA2 for chaining

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		QERR	DIRCHG	IDX
Access	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 2 – QERR Quadrature Error

Value	Description
0	No effect.
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		QERR	DIRCHG	IDX
Access	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 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		QERR	DIRCHG	IDX
Access	R	R	R	R		R	R	R
Reset	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 2 – QERR Quadrature Error

Value	Description
0	The interrupt on quadrature error is disabled.
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		QERR	DIRCHG	IDX
Access	R	R	R	R		R	R	R
Reset	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 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 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
Access								DIR
Reset								R
								0

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bit 8 – DIR Direction
Returns an image of the current rotation direction.

67.7.24. 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) controls several channels independently. Each channel controls one square output waveform. Characteristics of the output waveform such as period, duty cycle and polarity are configurable 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.

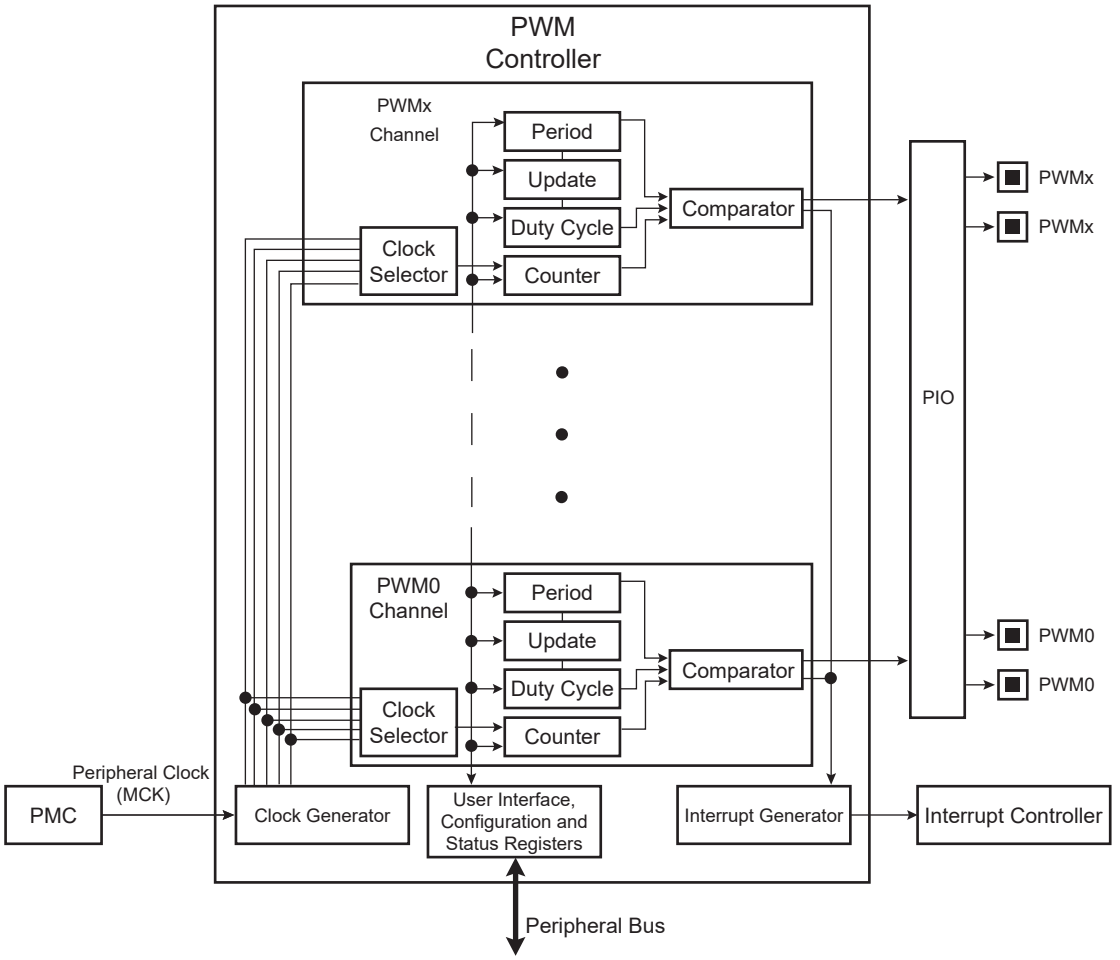
Channels can be synchronized, to generate non overlapped waveforms. All channels integrate a double buffering system in order to prevent an unexpected output waveform while modifying the period or the duty cycle.

68.2. Embedded Characteristics

- 4 Channels
- One 16-bit Counter Per Channel
- 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 Enable Disable Command for Each Channel
  - Independent Clock Selection for Each Channel
  - Independent Period and Duty Cycle for Each Channel
  - Double Buffering of Period or Duty Cycle for Each Channel
  - Programmable Selection of The Output Waveform Polarity for Each Channel
  - Programmable Center or Left Aligned Output Waveform for Each Channel Block Diagram

68.3. Block Diagram

Figure 68.1. PWM Block Diagram



68.4. I/O Lines Description

Each channel outputs one waveform on one external I/O line.

Table 68.1. I/O Line Description

Name	Description	Type
PWMx	PWM Waveform Output for channel x	Output

68.5. Product Dependencies

68.5.1. I/O Lines

The pins used for interfacing the PWM controller may be 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 controller 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 will be assigned to PWM outputs.

68.5.2. Power Management

The PWM controller 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 controller will resume its operations where it left off.

All the PWM registers except the Channel Duty Cycle (PWM_CDTYx) and Channel Period (PWM_CPRDx) registers can be read without the PWM peripheral clock enabled. All the registers can be written without the peripheral clock enabled.

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. Note that it is not recommended to use the PWM interrupt line in Edge Sensitive mode.

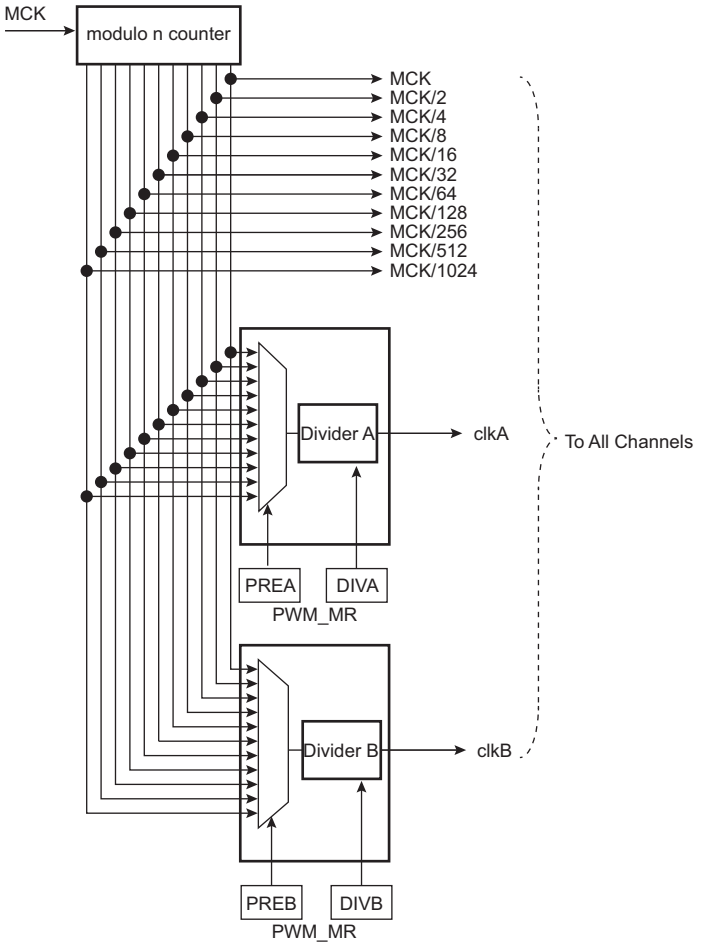
68.6. Functional Description

The PWM controller is primarily composed of a clock generator module and 4 channels.

- Clocked by the system clock, MCK, 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



CAUTION Before using the PWM controller, the programmer must first enable the PWM clock in the Power Management Controller (PMC).

The PWM peripheral clock, MCK, 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 in three blocks:

- a modulo n counter which provides 11 clocks: f_{MCK}, $f_{MCK}/2$, $f_{MCK}/4$, $f_{MCK}/8$, $f_{MCK}/16$, $f_{MCK}/32$, $f_{MCK}/64$, $f_{MCK}/128$, $f_{MCK}/256$, $f_{MCK}/512$, $f_{MCK}/1024$
- 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 Mode register (PWM_MR). The resulting clock clkA (clkB) is the clock selected divided by DIVA (DIVB) field value in PWM_MR.

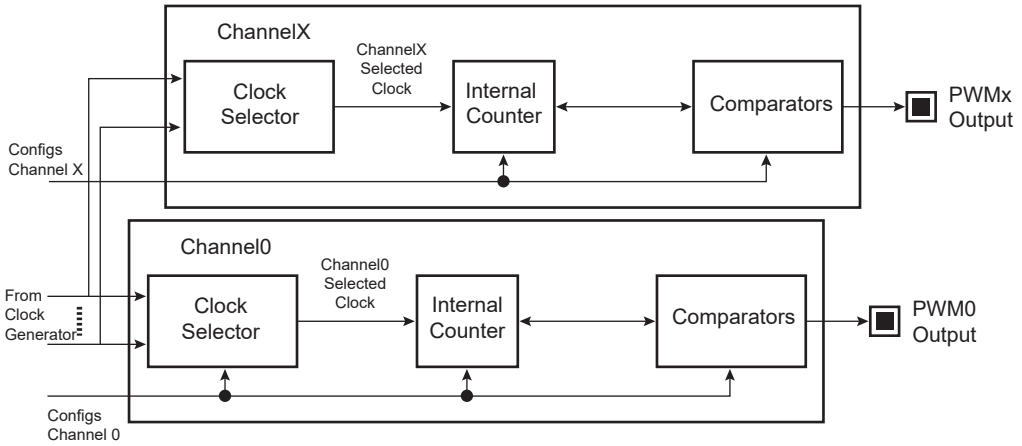
After a reset of the PWM controller, DIVA (DIVB) and PREA (PREB) in PWM_MR are set to 0. This implies that after reset clkA (clkB) are turned off.

At reset, all clocks provided by the modulo n counter are turned off except clock “clk”. This situation is also true when the PWM peripheral clock is turned off through the Power Management Controller.

68.6.2. PWM Channel

68.6.2.1. Channel Processing Overview

Figure 68.3. Channel Processing Overview



Each of the 4 channels is composed of three blocks:

- A clock selector which selects one of the clocks provided by the clock generator described in the section "PWM Clock Generator".
- An internal counter clocked by the output of the clock selector. This internal counter is incremented or decremented according to the channel configuration and comparators events. The size of the internal counter is 16 bits.
- A comparator used to generate events according to the internal counter value. It also computes the PWMx output waveform according to the configuration.

68.6.2.2. Waveform Properties

The different properties of output waveforms are:

- the ***internal clock selection***. The internal 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 Channel Mode register (PWM_CMRx) This field is reset at 0.
- ***the waveform period***. This channel parameter is defined in the PWM_CPRDx.CPRD field.
  - If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
    - By using the peripheral clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024), the resulting period formula will be:

      $$\frac{(X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

    - By using a peripheral clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

      $$\frac{(X*\mathrm{CPRD}*\mathrm{DIVA})}{\mathrm{MCK}} \text{ or } \frac{(X*\mathrm{CPRD}*\mathrm{DIVB})}{\mathrm{MCK}}$$

  - If the waveform is center-aligned then the output waveform period depends on the counter source clock and can be calculated:
    - By using the peripheral clock divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

      $$\frac{(2 \times X \times \text{CPRD})}{\text{MCK}}$$

    - By using a peripheral clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

      $$\frac{(2*X*\text{CPRD}*\text{DIVA})}{\text{MCK}} \text{ or } \frac{(2*X*\text{CPRD}*\text{DIVB})}{\text{MCK}}$$

- ***the waveform duty cycle***. This channel parameter is defined in the PWM_CDTYx.CDTY field. If the waveform is left-aligned, then:

  $$\text{duty cycle} = \frac{(\text{PWM\_period} - (\text{CDTY} \times \text{SRC\_period}))}{\text{PWM\_period}}$$

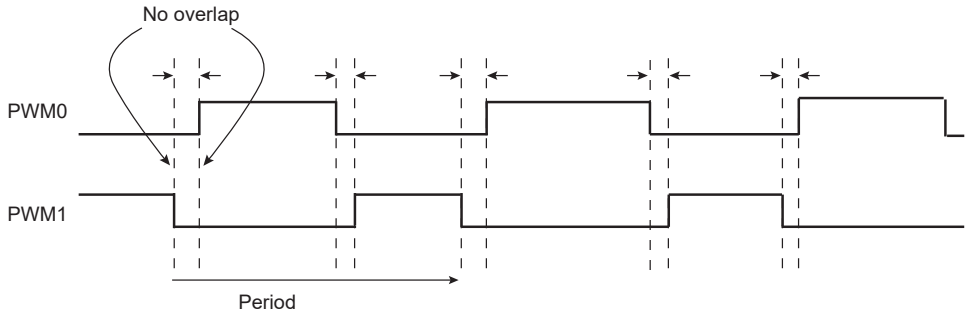
  If the waveform is center-aligned, then:

  $$\text{duty cycle} = \frac{\left(\left(\frac{PWM\_period}{2}\right) - \left(\text{CDTY} \times \text{SRC\_period}\right)\right)}{\left(\frac{PWM\_period}{2}\right)}$$

  where:

  "PWM_period" is the PWM output period and "SRC_period" is the PWM source clock period. "SRC_period" depends on the peripheral clock period and on PWM_CMRx.CPRE, PWM_CLK.(PREA, PREB, DIVA, DIVB), PWM_CPRDx.CPRD and PWM_CMRx.CPRE as per previous formulas.
- ***the waveform polarity***. At the beginning of the period, the signal can be at high or low level. This property is defined in the PWM_CMRx.CPOL field. By default the signal starts by a low level.
- ***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 PWM_CMRx.CALG field. The default mode is left-aligned.

Figure 68.4. Non Overlapped Center-aligned Waveforms



Note: See the figure below for a detailed description of center-aligned waveforms.

When center-aligned, the internal channel counter increases up to CPRD and decreases down to 0. This ends the period.

When left-aligned, the internal 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:

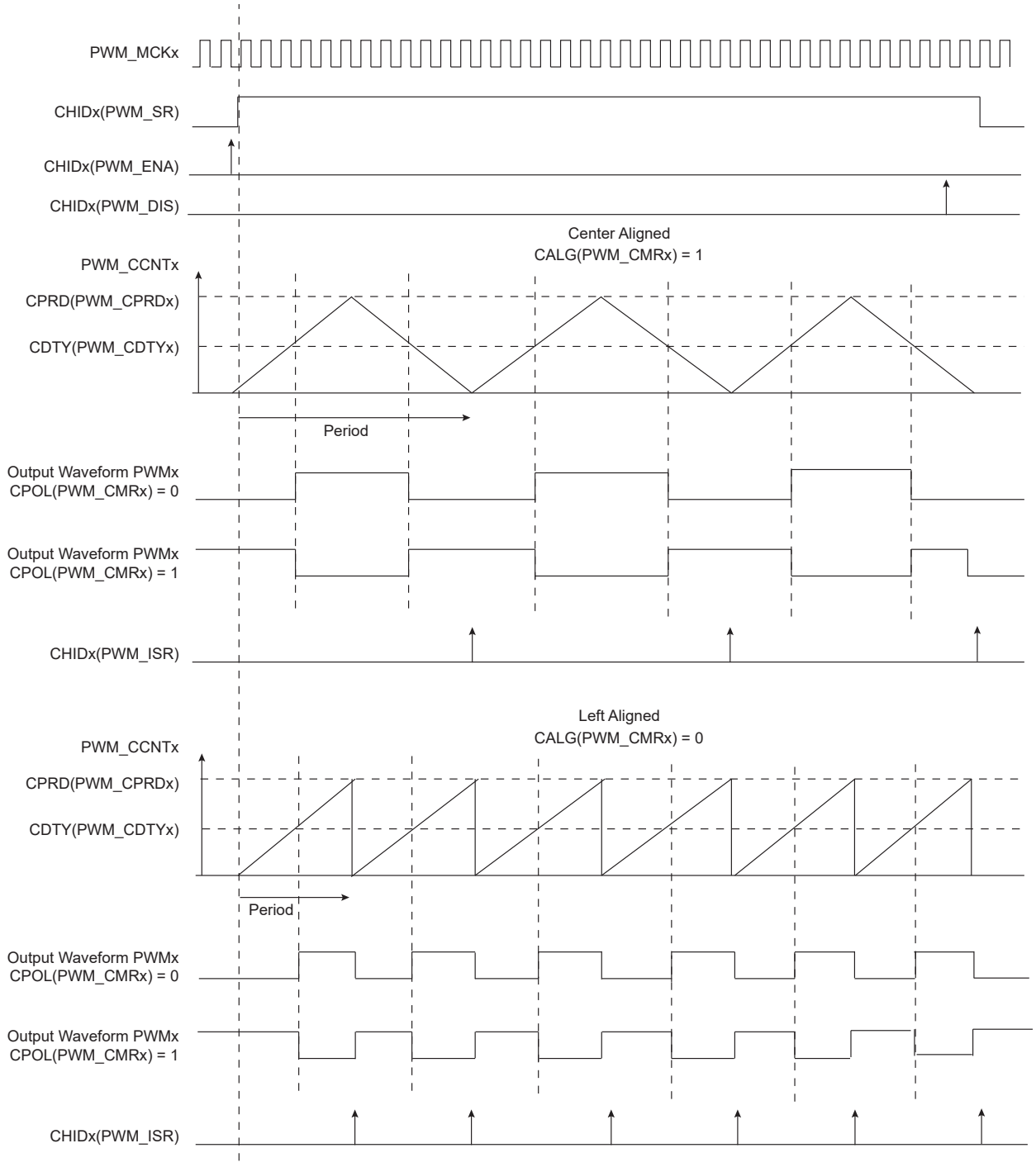
- CDTY = CPRD and CPOL = 0
- CDTY = 0 and CPOL = 1

Waveforms are fixed at 1 (once the channel is enabled) when:

- CDTY = 0 and CPOL = 0
- CDTY = CPRD and CPOL = 1

The waveform polarity must be set before enabling the channel. This immediately affects the channel output level. Changes on channel polarity are not taken into account while the channel is enabled.

Figure 68.5. Waveform Properties



68.6.3. PWM Controller Operations

68.6.3.1. Initialization

Before enabling the output channel, this channel must have been configured by the software application:

- Configuration of the clock generator if DIVA and DIVB are required
- Selection of the clock for each channel (PWM_CMRx.CPRE)

- Configuration of the waveform alignment for each channel (PWM_CMRx.CALG)
- Configuration of the period for each channel (PWM_CPRDx.CPRD). Writing in PWM_CPRDx is possible while the channel is disabled. After validation of the channel, the user must use the Channel Update register (PWM_CUPDx) to update PWM_CPRDx as explained below.
- Configuration of the duty cycle for each channel (PWM_CDTYx.CDTY). Writing in PWM_CDTYx is possible while the channel is disabled. After validation of the channel, the user must use PWM_CUPDx to update PWM_CDTYx as explained below.
- Configuration of the output waveform polarity for each channel (PWM_CMRx.CPOL)
- Enable Interrupts (set CHIDx in the Interrupt Enable register (PWM_IER))
- Enable the PWM channel (set CHIDx in the Enable register (PWM_ENA))

It is possible to synchronize different channels by enabling them at the same time by means of simultaneously setting several CHIDx bits in PWM_ENA.

- In such a situation, all channels may have the same clock selector configuration and the same period specified.

68.6.3.2. Source Clock Selection Criteria

The large number of source clocks can make selection difficult. The relationship between the value in the Channel Period register (PWM_CPRDx) and the Channel Duty Cycle register (PWM_CDTYx) can help the user in choosing. The event number written in PWM_CPRDx gives the PWM accuracy. The Duty Cycle quantum cannot be lower than 1/PWM_CPRDx value. The higher the value of PWM_CPRDx, the greater the PWM accuracy.

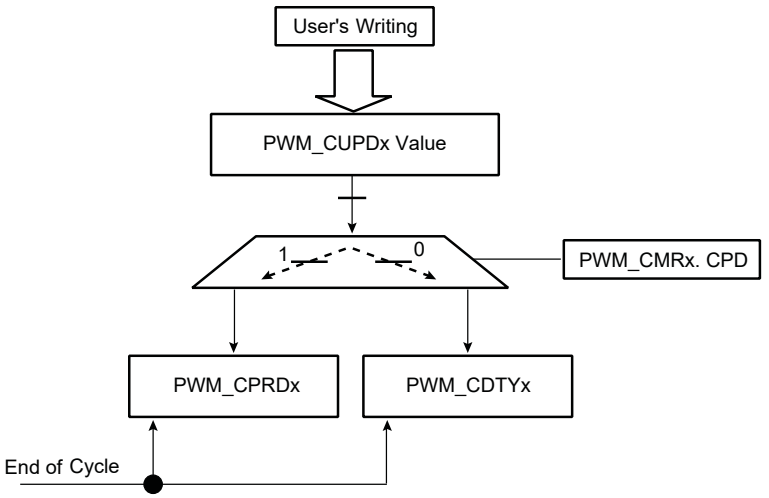
For example, if the user sets 15 (in decimal) in PWM_CPRDx, the user is able to set a value between 1 up to 14 in PWM_CDTYx. The resulting duty cycle quantum cannot be lower than 1/15 of the PWM period.

68.6.3.3. Changing the Duty Cycle or the Period

It is possible to modulate the output waveform duty cycle or period.

To prevent unexpected output waveform, the user must use PWM_CUPDx to change waveform parameters while the channel is still enabled. The user can write a new period value or duty cycle value in PWM_CUPDx. This register holds the new value until the end of the current cycle and updates the value for the next cycle. Depending on the PWM_CMRx.CPD field, PWM_CUPDx either updates PWM_CPRDx or PWM_CDTYx. Note that even if the update register is used, the period must not be smaller than the duty cycle.

Figure 68.6. Synchronized Period or Duty Cycle Update



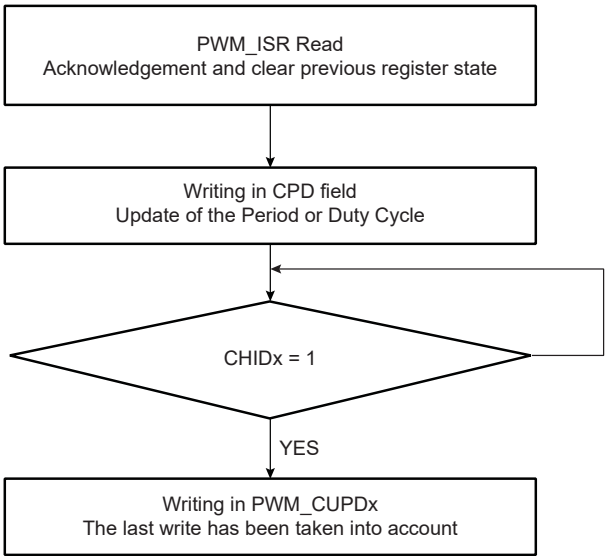
To prevent overwriting the PWM_CUPDx by software, the user can use status events in order to synchronize his software. Two methods are possible. In both, the user must enable the dedicated interrupt in PWM_IER at PWM controller level.

The first method (polling method) consists of reading the relevant status bit in the Interrupt Status register (PWM_ISR) according to the enabled channel(s). See the figure below.

The second method uses an Interrupt Service Routine associated with the PWM channel.

Note: Reading PWM_ISR automatically clears the CHIDx flags.

Figure 68.7. Polling Method



Note: Polarity and alignment can be modified only when the channel is disabled.

68.7. Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	PWM_MR	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_IER	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x14	PWM_IDR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x18	PWM_IMR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x1C	PWM_ISR	31:24								
		23:16								
		15:8								
		7:0					CHID3	CHID2	CHID1	CHID0
0x20 ... 0x01FF	Reserved									
0x0200	PWM_CMR0	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0204	PWM_CDTY0	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0208	PWM_CPRD0	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x020C	PWM_CCNT0	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0210	PWM_CUPD0	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							
0x0214 ... 0x021F	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0220	PWM_CMR1	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0224	PWM_CDTY1	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0228	PWM_CPRD1	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x022C	PWM_CCNT1	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0230	PWM_CUPD1	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							
0x0234 ... 0x023F	Reserved									
0x0240	PWM_CMR2	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0244	PWM_CDTY2	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0248	PWM_CPRD2	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							
0x024C	PWM_CCNT2	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0250	PWM_CUPD2	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							
0x0254 ... 0x025F	Reserved									
0x0260	PWM_CMR3	31:24								
		23:16								
		15:8						CPD	CPOL	CALG
		7:0					CPRE[3:0]			
0x0264	PWM_CDTY3	31:24	CDTY[31:24]							
		23:16	CDTY[23:16]							
		15:8	CDTY[15:8]							
		7:0	CDTY[7:0]							
0x0268	PWM_CPRD3	31:24	CPRD[31:24]							
		23:16	CPRD[23:16]							
		15:8	CPRD[15:8]							
		7:0	CPRD[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x026C	PWM_CCNT3	31:24	CNT[31:24]							
		23:16	CNT[23:16]							
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0270	PWM_CUPD3	31:24	CUPD[31:24]							
		23:16	CUPD[23:16]							
		15:8	CUPD[15:8]							
		7:0	CUPD[7:0]							

68.7.1. PWM Mode Register

Name: PWM_MR
Offset: 0x00
Reset: 0x00000000
Property: Read/Write

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
0000	MCK	Peripheral clock
0001	MCKDIV2	Peripheral clock divided by 2
0010	MCKDIV4	Peripheral clock divided by 4
0011	MCKDIV8	Peripheral clock divided by 8
0100	MCKDIV16	Peripheral clock divided by 16
0101	MCKDIV32	Peripheral clock divided by 32
0110	MCKDIV64	Peripheral clock divided by 64
0111	MCKDIV128	Peripheral clock divided by 128
1000	MCKDIV256	Peripheral clock divided by 256
1001	MCKDIV512	Peripheral clock divided by 512
1010	MCKDIV1024	Peripheral clock divided by 1024
Other	–	Reserved

Bits 23:16 – DIVB[7:0] CLKB Divide Factor

Value	Name	Description
0	CLK_OFF	The CLKB internal source clock is turned off. The PWMx output is stuck when PWM_CMRx.CPRE=CLKB.
1	CLK_DIV1	CLKB is the clock selected by PREB.
2–255	–	CLKB is the clock selected by PREB, divided by the DIVB factor.

Bits 11:8 – PREA[3:0] CLKA Source Clock Selection

Value	Name	Description
0000	MCK	Peripheral clock
0001	MCKDIV2	Peripheral clock divided by 2
0010	MCKDIV4	Peripheral clock divided by 4
0011	MCKDIV8	Peripheral clock divided by 8
0100	MCKDIV16	Peripheral clock divided by 16
0101	MCKDIV32	Peripheral clock divided by 32
0110	MCKDIV64	Peripheral clock divided by 64
0111	MCKDIV128	Peripheral clock divided by 128
1000	MCKDIV256	Peripheral clock divided by 256
1001	MCKDIV512	Peripheral clock divided by 512
1010	MCKDIV1024	Peripheral clock divided by 1024
Other	–	Reserved

Bits 7:0 – DIVA[7:0] CLKA Divide Factor

Value	Name	Description
0	CLK_OFF	The CLKA internal source clock is turned off. The PWMx output is stuck when PWM_CMRx.CPRE=CLKA.
1	CLK_DIV1	CLKA is the clock selected by PREA.
2–255	–	CLKA is the clock selected by PREA, divided by the 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

Bit	31	30	29	28	27	26	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

Name: PWM_IER
Offset: 0x10
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 interrupt for PWM channel x.

68.7.6. PWM Interrupt Disable Register

Name: PWM_IDR
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
					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 interrupt for PWM channel x.

68.7.7. PWM Interrupt Mask Register

Name: PWM_IMR
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
					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	Interrupt for PWM channel x is disabled.
1	Interrupt for PWM channel x is enabled.

68.7.8. PWM Interrupt Status Register

Name: PWM_ISR
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Note: Reading PWM_ISR automatically clears the CHIDx 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
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	No new channel period has been achieved since the last read of PWM_ISR.
1	At least one new channel period has been achieved since the last read of PWM_ISR.

68.7.9. PWM Channel Mode Register x

Name: PWM_CMRx
Offset: 0x0200 + x*0x20 [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
						CPD	CPOL	CALG
Access						R/W	R/W	R/W
Reset						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 10 – CPD Channel Update Period

Value	Description
0	Writing to PWM_CUPDx will modify the duty cycle at the next period start event.
1	Writing to PWM_CUPDx will modify the period at the next period start event.

Bit 9 – CPOL Channel Polarity

Value	Description
0	The output waveform starts at a low level.
1	The output waveform 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
0000	MCK	Peripheral clock
0001	MCKDIV2	Peripheral clock divided by 2
0010	MCKDIV4	Peripheral clock divided by 4
0011	MCKDIV8	Peripheral clock divided by 8
0100	MCKDIV16	Peripheral clock divided by 16
0101	MCKDIV32	Peripheral clock divided by 32
0110	MCKDIV64	Peripheral clock divided by 64

Value	Name	Description
0111	MCKDIV128	Peripheral clock divided by 128
1000	MCKDIV256	Peripheral clock divided by 256
1001	MCKDIV512	Peripheral clock divided by 512
1010	MCKDIV1024	Peripheral clock divided by 1024
1011	CLKA	Clock A
1100	CLKB	Clock B
Other	–	Reserved

68.7.10. PWM Channel Duty Cycle Register x

Name: PWM_CDTYx
Offset: 0x0204 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Only the first 16 bits (internal channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
	CDTY[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
	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 31:0 – CDTY[31:0] Channel Duty Cycle
Defines the waveform duty cycle. This value must be defined between 0 and CPRD (PWM_CPRx).

68.7.11. PWM Channel Period Register

Name: PWM_CPRDx
Offset: 0x0208 + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read/Write

Only the first 16 bits (internal channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
	CPRD[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
	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 31:0 – CPRD[31:0] Channel Period

If the waveform is left-aligned, then the output waveform period depends on the counter source clock and can be calculated:
– By using the peripheral clock (MCK) divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

$$\frac{(X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

– By using a peripheral clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(\mathrm{CPRD} \times \mathrm{DIVA})}{\mathrm{MCK}} \text{ or } \frac{(\mathrm{CPRD} \times \mathrm{DIVB})}{\mathrm{MCK}}$$

If the waveform is center-aligned, then the output waveform period depends on the counter source clock and can be calculated:
– By using the peripheral clock divided by an X given prescaler value (with X being 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024). The resulting period formula will be:

$$\frac{(2 \times X \times \mathrm{CPRD})}{\mathrm{MCK}}$$

– By using a peripheral clock divided by one of both DIVA or DIVB divider, the formula becomes, respectively:

$$\frac{(2 \times \mathrm{CPRD} \times \mathrm{DIVA})}{\mathrm{MCK}} \text{ or } \frac{(2 \times \mathrm{CPRD} \times \mathrm{DIVB})}{\mathrm{MCK}}$$

68.7.12. PWM Channel Counter Register x

Name: PWM_CCNTx
Offset: 0x020C + x*0x20 [x=0..3]
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	CNT[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
	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 31:0 – CNT[31:0] Channel Counter Register
Internal counter value. This register is reset when:

- the channel is enabled (PWM_ENA.CHIDx = 1)
- the counter reaches CPRD value defined in PWM_CPRDx if the waveform is left aligned.

68.7.13. PWM Channel Update Register x

Name: PWM_CUPDx
Offset: 0x0210 + x*0x20 [x=0..3]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	CUPD[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	23	22	21	20	19	18	17	16
	CUPD[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	15	14	13	12	11	10	9	8
	CUPD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit	7	6	5	4	3	2	1	0
	CUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 31:0 – CUPD[31:0] Channel Update Register

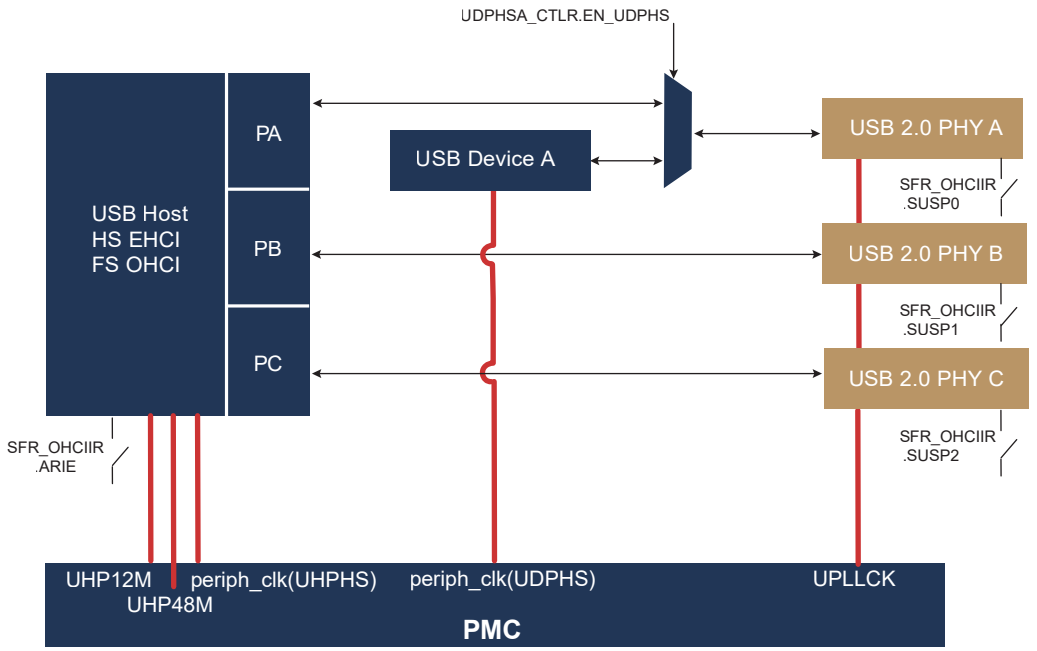
This register acts as a double buffer for the period or the duty cycle. This prevents an unexpected waveform when modifying the waveform period or duty cycle.
Only the first 16 bits (internal channel counter size) are significant.
When PWM_CMRx.CPD = 0, the duty cycle (CDTY of PWM_CDTYx) is updated with the CUPD value at the beginning of the next period.
When PWM_CMRx.CPD = 1, the period (CPRD of PWM_CPRDx) is updated with the CUPD value at the beginning of the next period.

USB Subsystem

69. Overview

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)
- 1x high-speed device shared with PA and PB (UDPHS)
- 3x USB 2.0 PHYs

69.3. Product Dependencies

69.3.1. Clocks

To generate USB frequencies with high accuracy, the device embeds a dedicated USB PLL.

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.

69.3.4. I/Os

USB I/Os are dedicated I/Os powered by VDDIN33/GND.

69.4. Special Functions in SFR

- SFR_OHCIICR.SUSPx controls suspend for each port. These bits must be set to reduce power consumption on VDDIN33 in Low-power mode.
- SFR_OHCIICR.ARIE enables OHCI asynchronous resume interrupt.
- SFR_OHCIISR.RISx indicates the Port Resume status of each port.

70. USB Device High Speed Port (UDPHS)

70.1. Description

The USB Device High Speed Port (UDPHS) is compliant with the Universal Serial Bus (USB), rev 2.0 High Speed device specification.

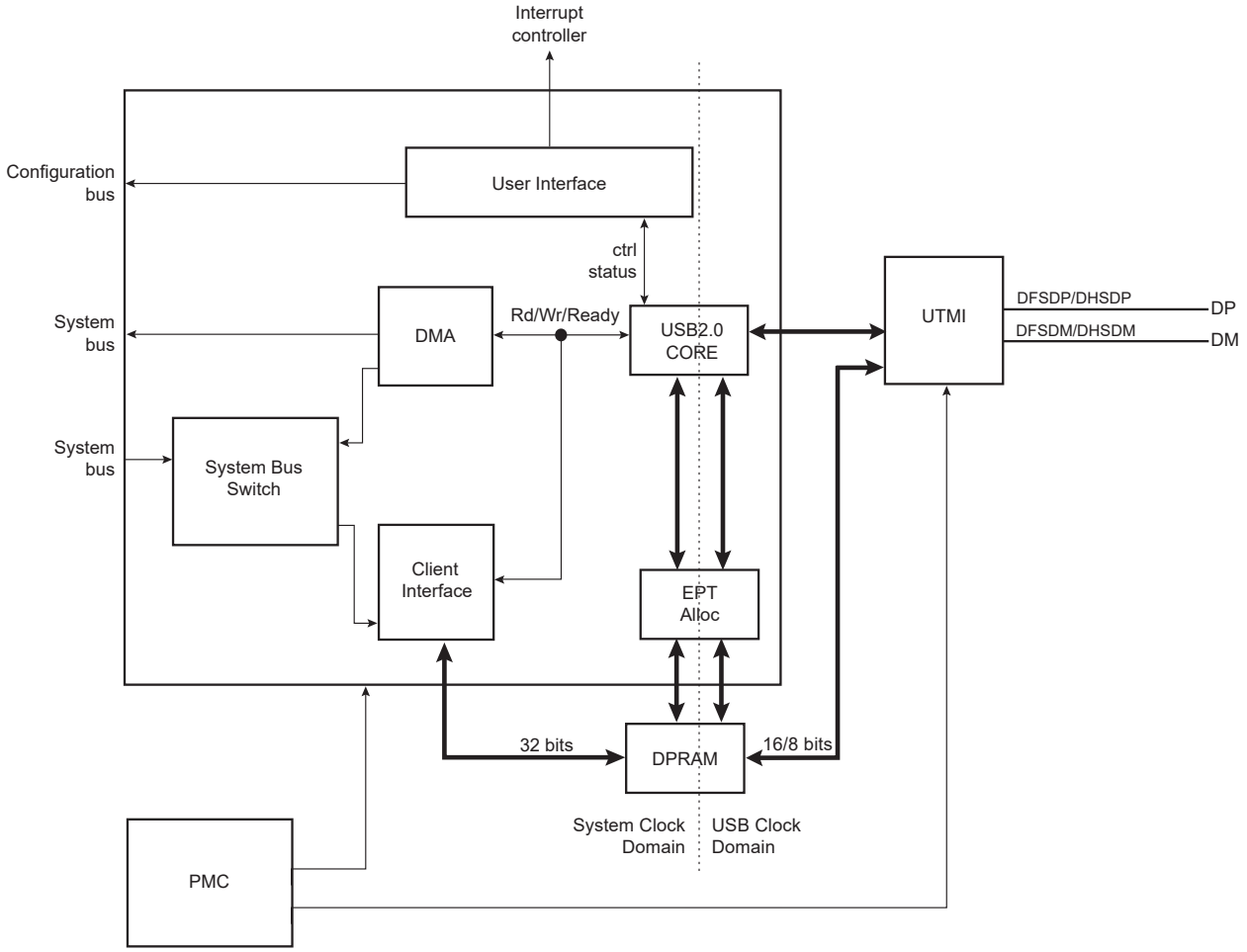
Each endpoint can be configured in one of several USB transfer types. It can be associated with one, two or three banks of a Dual-port RAM used to store the current data payload. If two or three banks are used, one DPR bank is read or written by the processor, while the other is read or written by the USB device peripheral. This feature is mandatory for isochronous endpoints.

70.2. Embedded Characteristics

- 1 High-Speed Device
- 1 UTMI Transceiver Shared Between Host and Device
- USB v2.0 High Speed (480 Mbits/s) Compliant
- 7 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

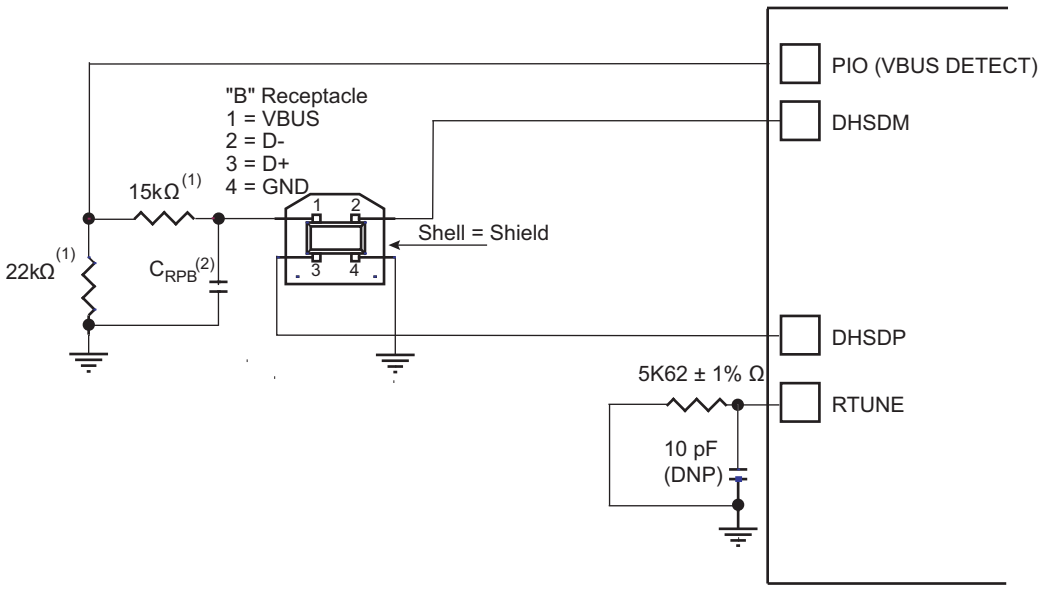
70.3. Block Diagram

Figure 70.1. UDPHS Block Diagram



70.4. Typical Connection

Figure 70.2. Board Schematic



Notes:

1. The values shown on the 22 kΩ and 15 kΩ resistors are only valid with 3.3V-supplied PIOs.
2. C_{RPB}: Upstream Facing Port Bypass Capacitance of 1 µF to 10 µF (refer to "DC Electrical Characteristics" in Universal Serial Bus Specification Rev. 2)
3. 10 pF capacitor on VBG is a provision and may not be populated.

70.5. Product Dependencies

70.5.1. 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). Then, UTMI (PLL and transceiver) must be enabled by first enabling the bandgap, then the voltage regulator in PMC.

However, if the application does not require UDPHS operations, the UDPHS clock can be stopped when not needed and restarted later.

70.5.2. Interrupt Sources

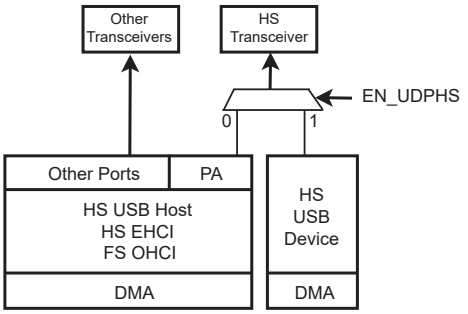
The UDPHS interrupt line is connected on one of the internal sources of the interrupt controller. Using the UDPHS interrupt requires the interrupt controller to be programmed first.

70.6. Functional Description

70.6.1. UTMI Transceivers Sharing

The High Speed USB Host Port A is shared with the High Speed USB Device port and connected to the second UTMI transceiver. The selection between Host Port A and USB Device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

Figure 70.3. USB Selection



70.6.2. 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.

70.6.3. 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 70.1. 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

70.6.4. USB Transfer Event Definitions

A transfer is composed of one or several transactions as shown in the table below.

Table 70.2. USB Transfer Events

Transfer		Transaction
Direction	Type	
CONTROL (bidirectional)	Control Transfer (1)	• Setup transaction → Data IN transactions → Status OUT transaction • Setup transaction → Data OUT transactions → Status IN transaction • Setup transaction → Status IN transaction

Table 70.2. USB Transfer Events (continued)

Transfer		Transaction
Direction	Type	
IN (device toward host)	Bulk IN Transfer	• Data IN transaction → Data IN transaction
	Interrupt IN Transfer	• Data IN transaction → Data IN transaction
	Isochronous IN Transfer (2)	• Data IN transaction → Data IN transaction
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 70.3. 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	2	Y	N	1024	Ctrl/Bulk/Iso(1)/Interrupt
2	EPT_2	2	Y	N	1024	Ctrl/Bulk/Iso(1)/Interrupt
3	EPT_3	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
4	EPT_4	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
5	EPT_5	3	Y	Y	1024	Ctrl/Bulk/Iso(1)/Interrupt
6	EPT_6	3	Y	Y	1024	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 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.

70.6.5. USB V2.0 High Speed BUS Transactions

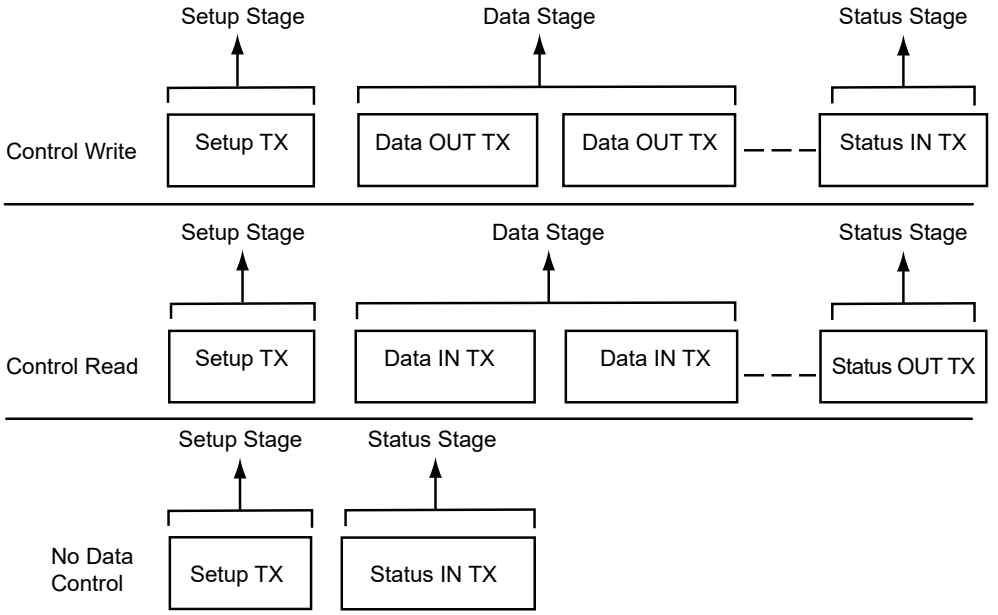
Each transfer results in one or more transactions over the USB bus.

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 70.4. Control Read and Write Sequences



70.6.6. 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.

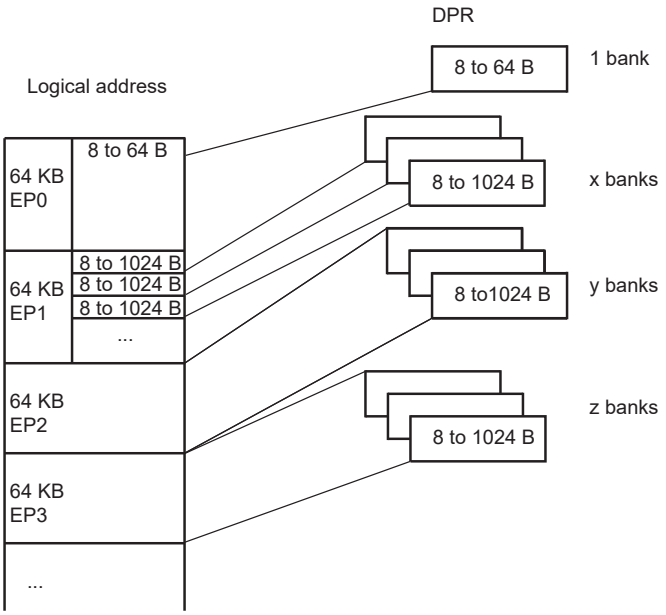
The maximum packet size they can accept corresponds to the maximum endpoint size.

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 70.5. 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.

70.6.7. 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.

70.6.8. 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.

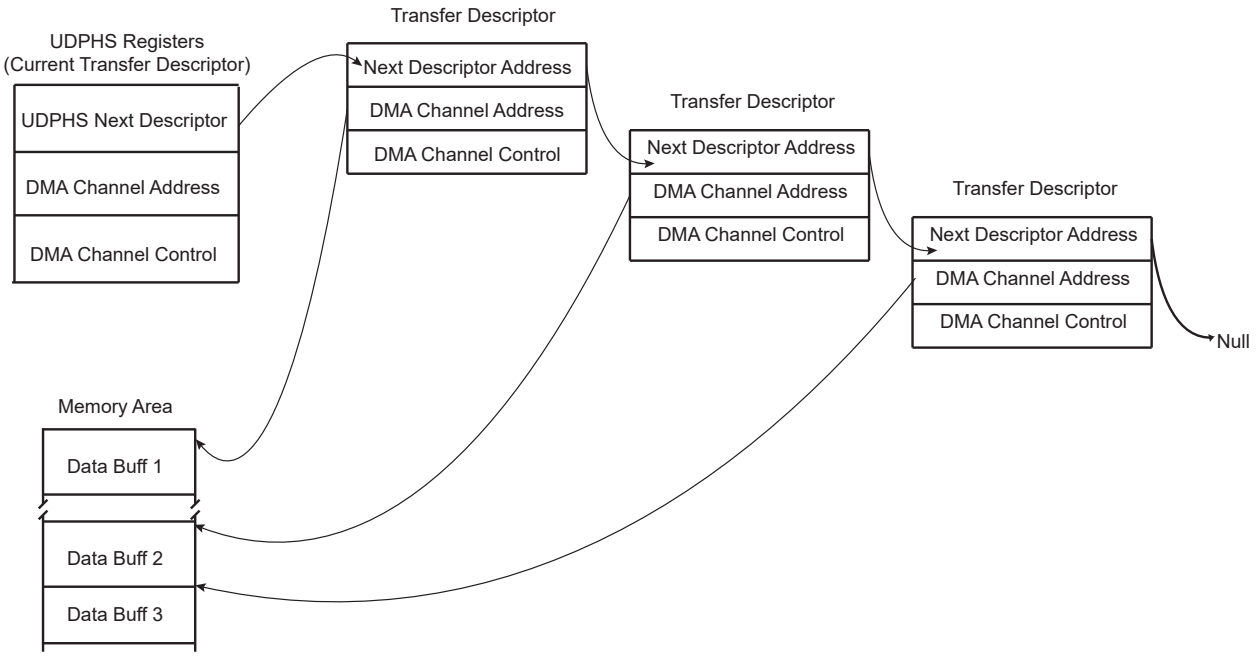
This maximum burst length is then controlled by the lowest programmed USB endpoint size (EPT_SIZE field in the UDPHS_EPTCFGx register) and DMA Size (BUFF_LENGTH field in the UDPHS_DMACONTROLx register).

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 70.6. Example of DMA Chained List



70.6.9. 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 occurred, the DMA could process data before an interrupt without the user's knowledge.

The recommended means to disable DMA are as follows:

```
// Reset IP UDPHS
  BASE_UDPHS->UDPHS_CTRL &= ~UDPHS_EN_UDPHS;
  BASE_UDPHS->UDPHS_CTRL |= UDPHS_EN_UDPHS;//
With OR without DMA !!!
  for( i=1; i<=DMA_CHANNEL_NBR); i++ )
// RESET endpoint canal DMA:
    // DMA stop channel command
   BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Disable endpoint
  BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLDIS |= 0XFFFFFFFF;
// Reset endpoint config
  BASE_UDPHS->UDPHS_EPT[i].UDPHS_EPTCTLCFG = 0;
// Reset DMA channel (Buff count and Control field)
  BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0x02; // NON
STOP command
// Reset DMA channel 0 (STOP)
  BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMACONTROL = 0; // STOP
command
// Clear DMA channel status (read the register for clear it)
  BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS =
  BASE_UDPHS->UDPHS_DMA[i].UDPHS_DMASTATUS;
}
```

70.6.10. Handling Transactions with USB V2.0 Device Peripheral

70.6.10.1.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).

70.6.10.2.NYET

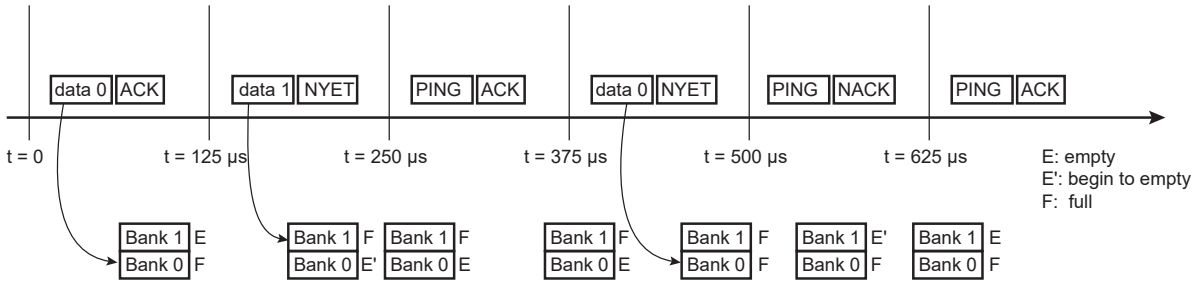
NYET is a High Speed only handshake. It is returned by a High Speed endpoint as part of the PING protocol.

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 70.7. NYET Example with Two Endpoint Banks



70.6.10.3.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 70.8. Data IN Transfer for Endpoint with One Bank

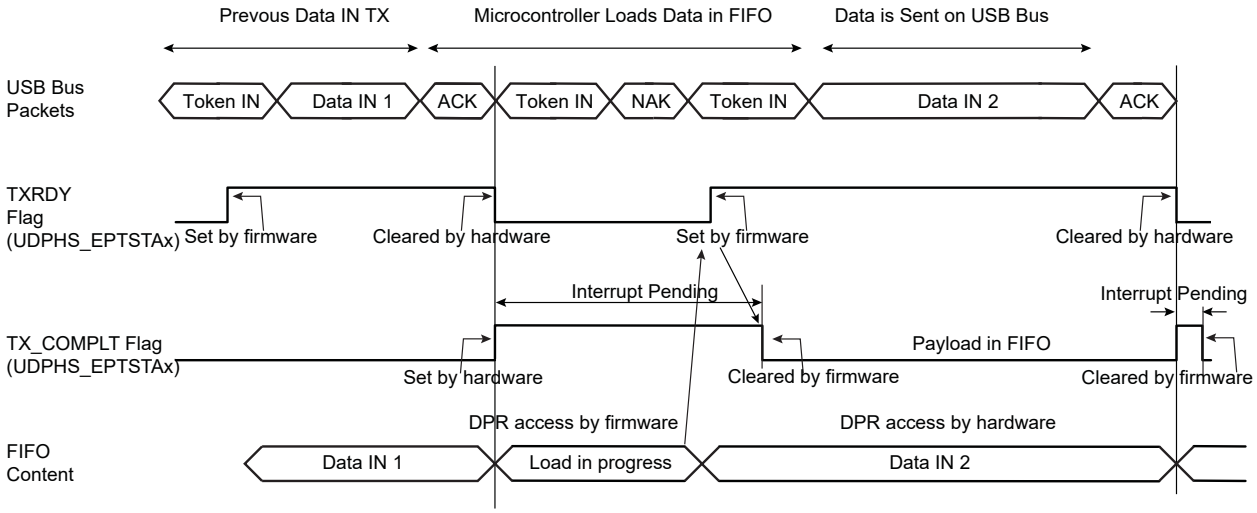


Figure 70.9. Data IN Transfer for Endpoint with Two Banks

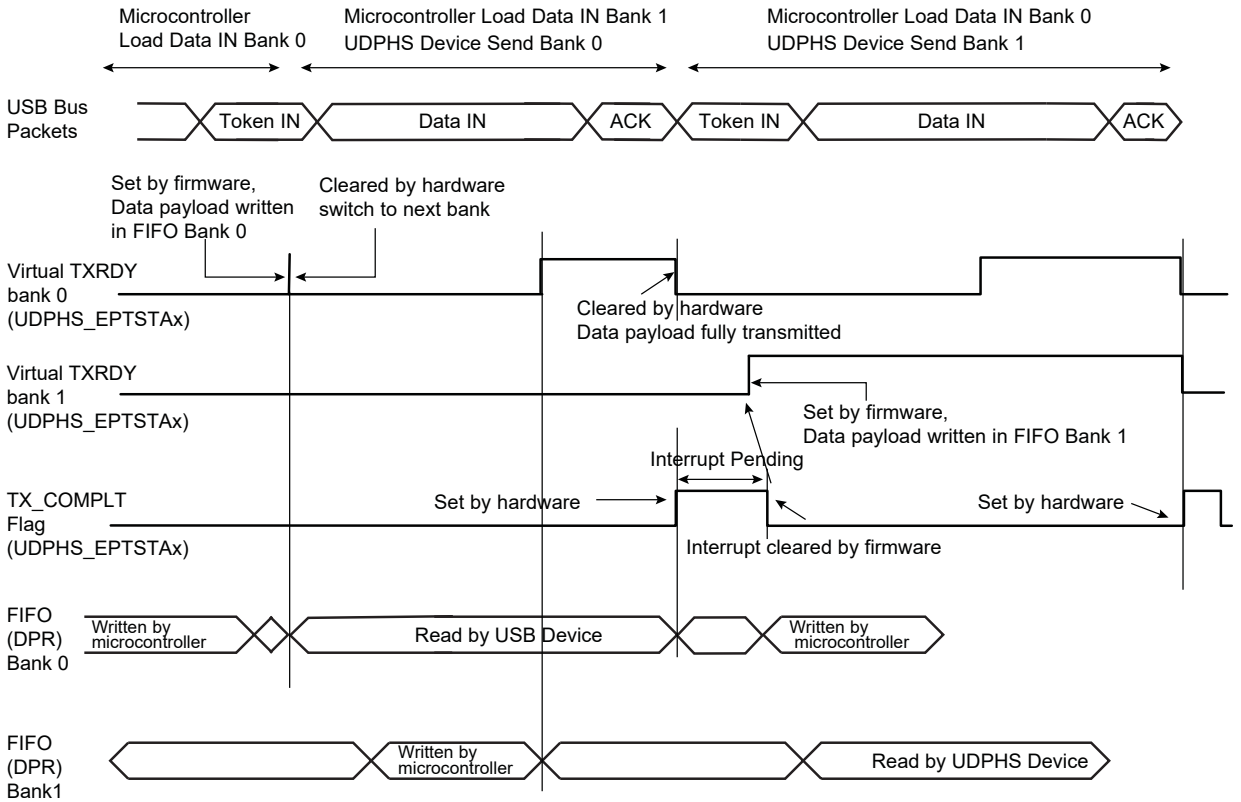
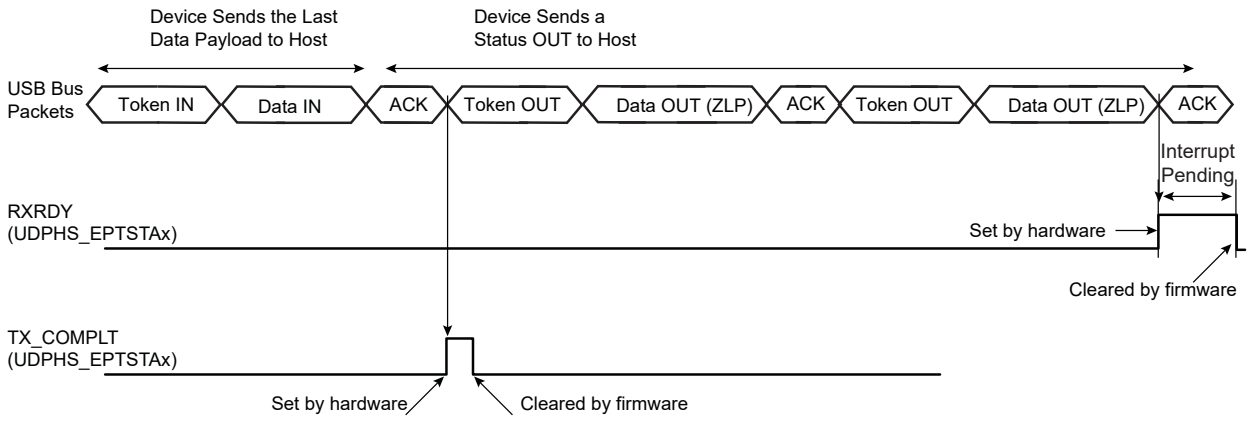
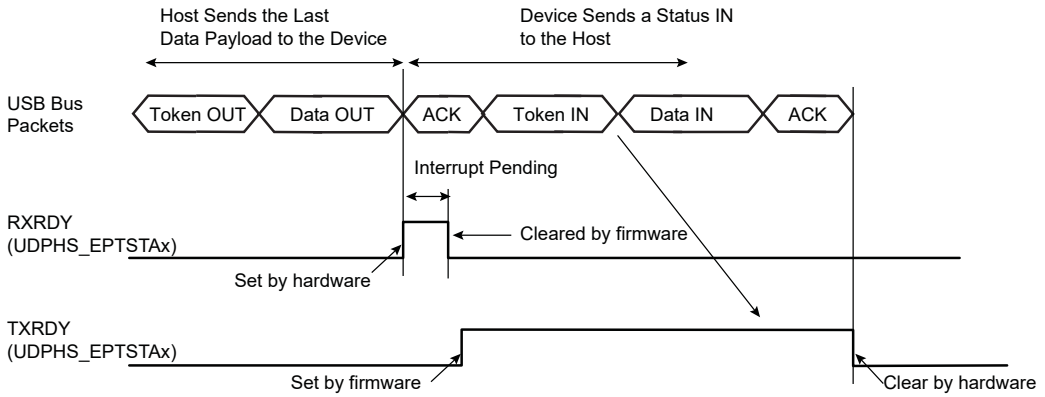


Figure 70.10. Data IN Followed By Status OUT Transfer at the End of a Control Transfer



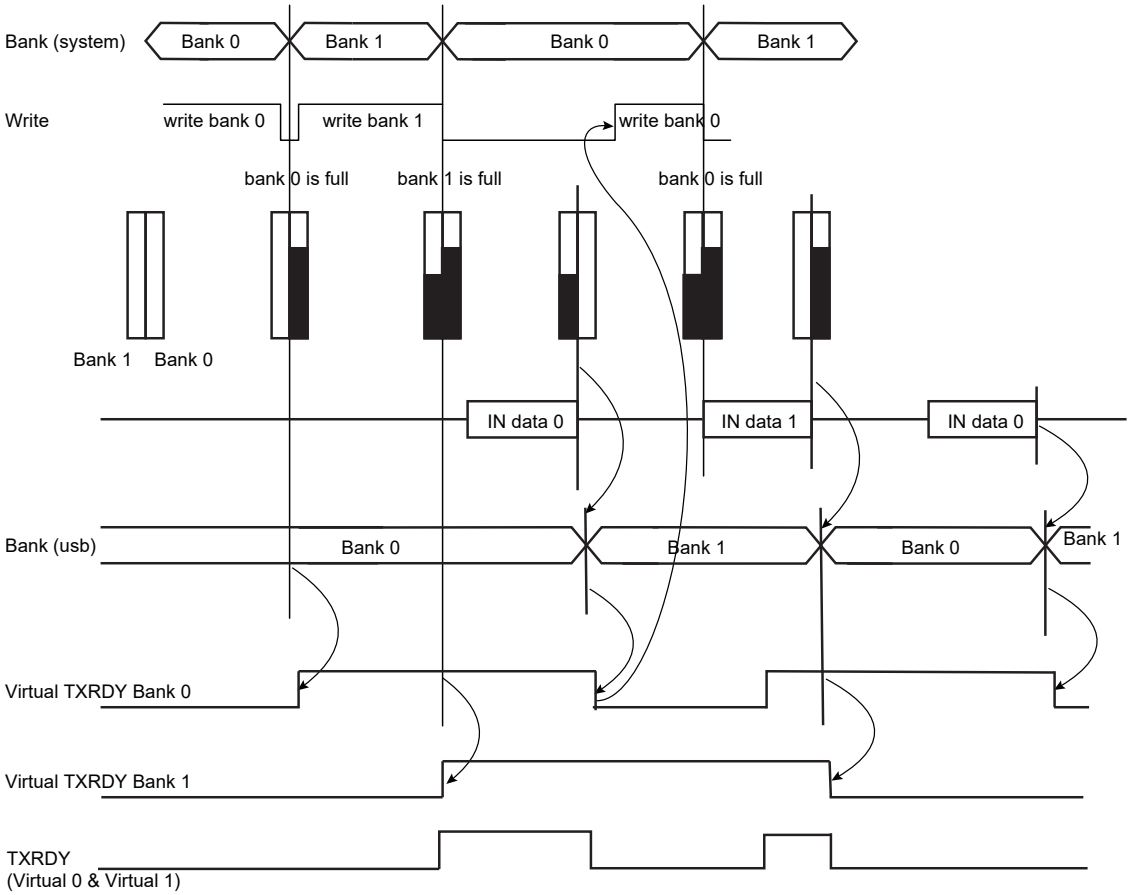
Note: A NAK handshake is always generated at the first status stage token.

Figure 70.11. 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 70.12. 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.

70.6.10.4.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 70.13. Data OUT Transfer for Endpoint with One Bank

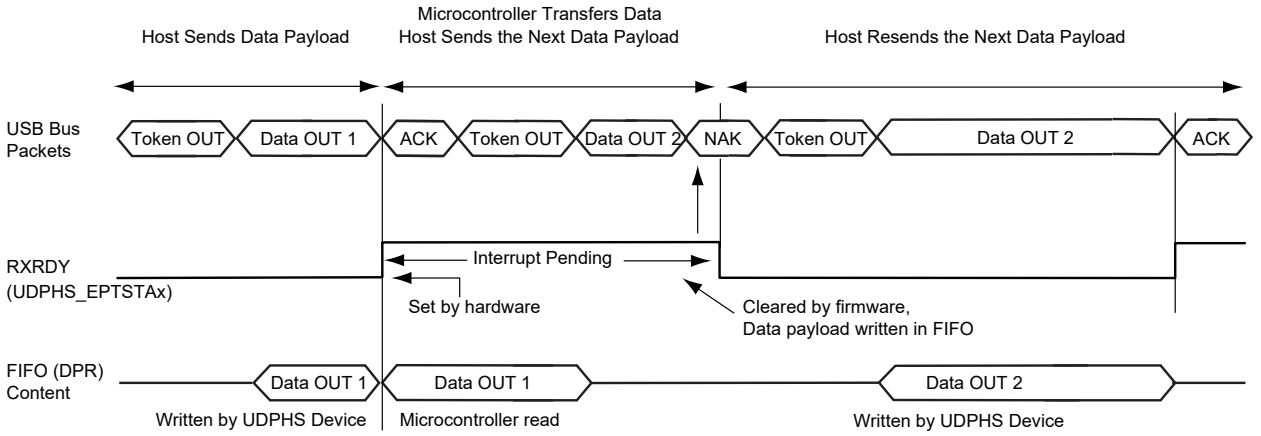
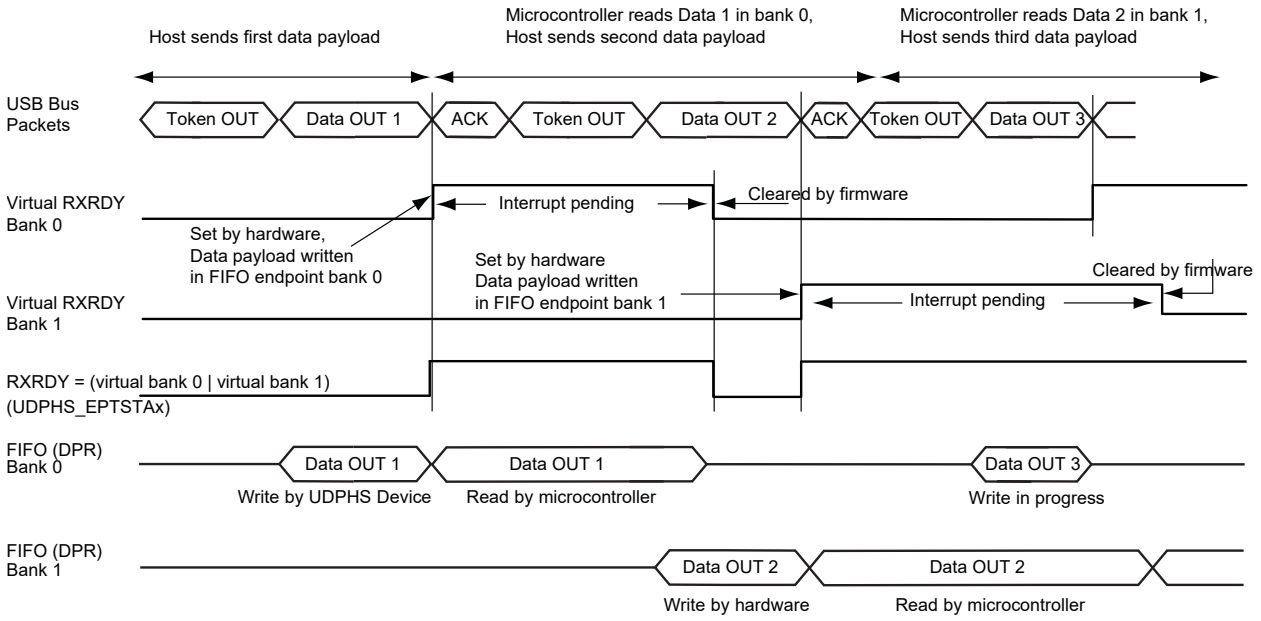


Figure 70.14. Data OUT Transfer for an Endpoint with Two Banks



- 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).

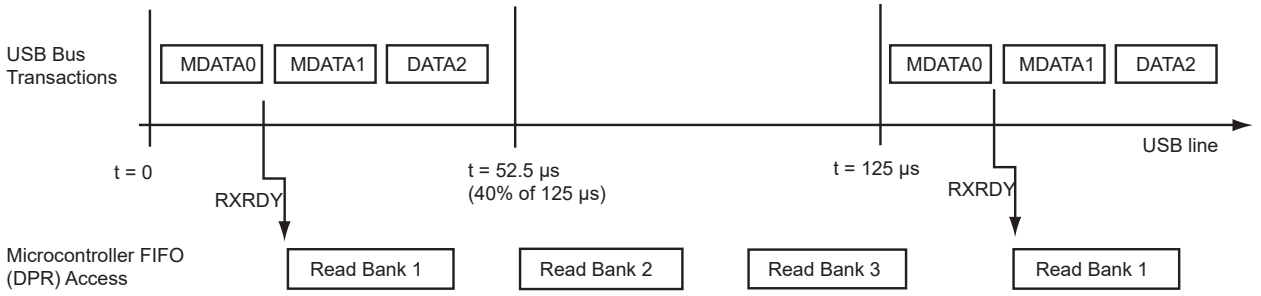
NB_TRANS field in UDPHS_EPTCFGx register = Number Of Transactions per Microframe.

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 70.15. 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.

70.6.10.5.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 70.16. Stall Handshake Data OUT Transfer

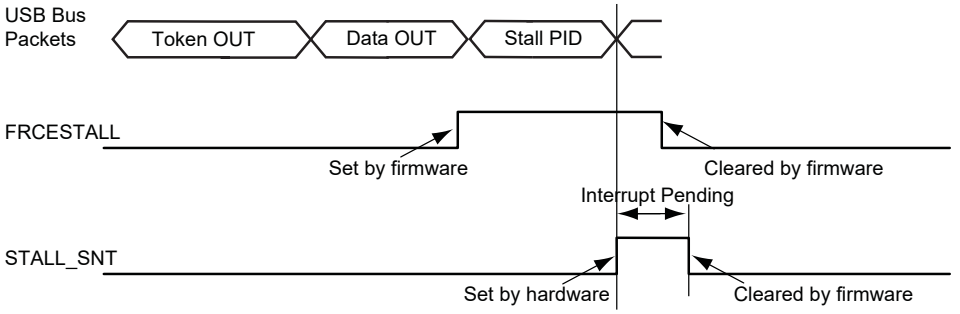
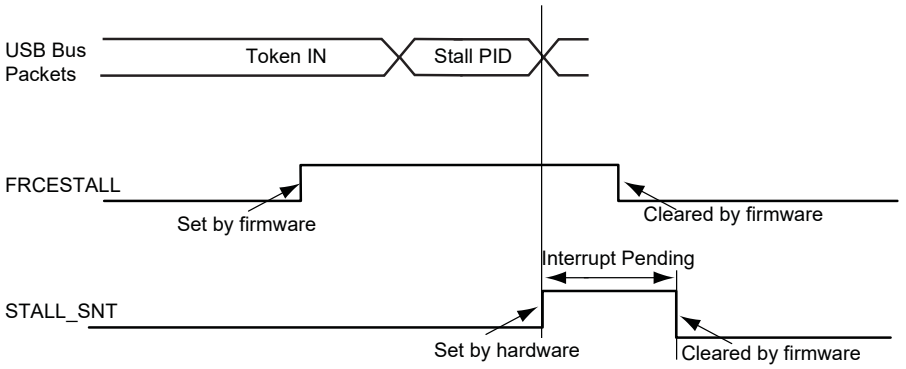


Figure 70.17. Stall Handshake Data IN Transfer



70.6.11. 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.

70.6.12. USB V2.0 High Speed Global Interrupt

Interrupts are defined in UDPHS_IEN and in UDPHS_INTSTA (Interrupt Status register).

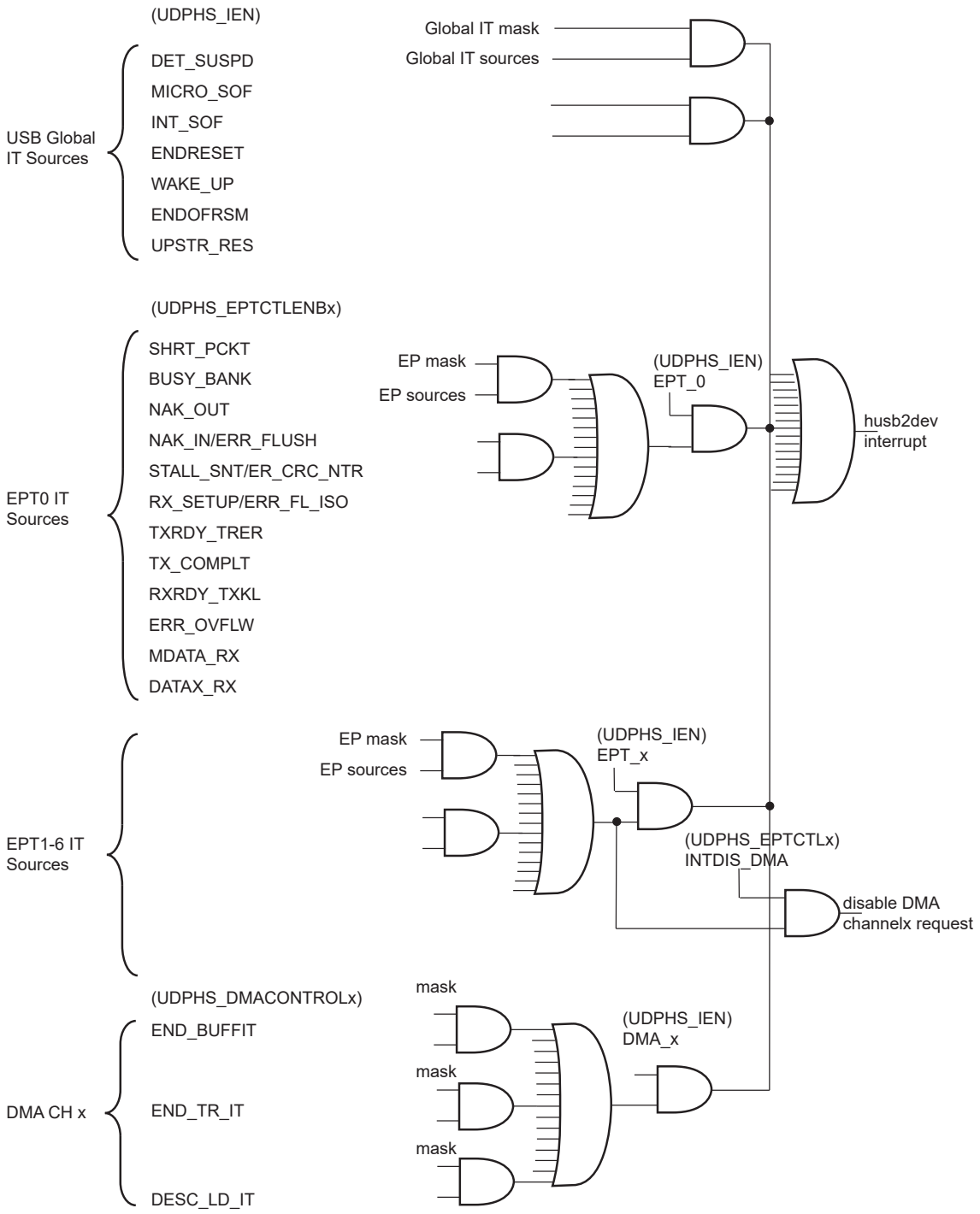
70.6.13. Endpoint Interrupts

Interrupts are enabled in UDPHS_IEN and individually masked in UDPHS_EPTCTLENBx.

Table 70.4. Endpoint Interrupt Source Masks

Acronym	Description
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 70.18. UDPHS Interrupt Control Interface

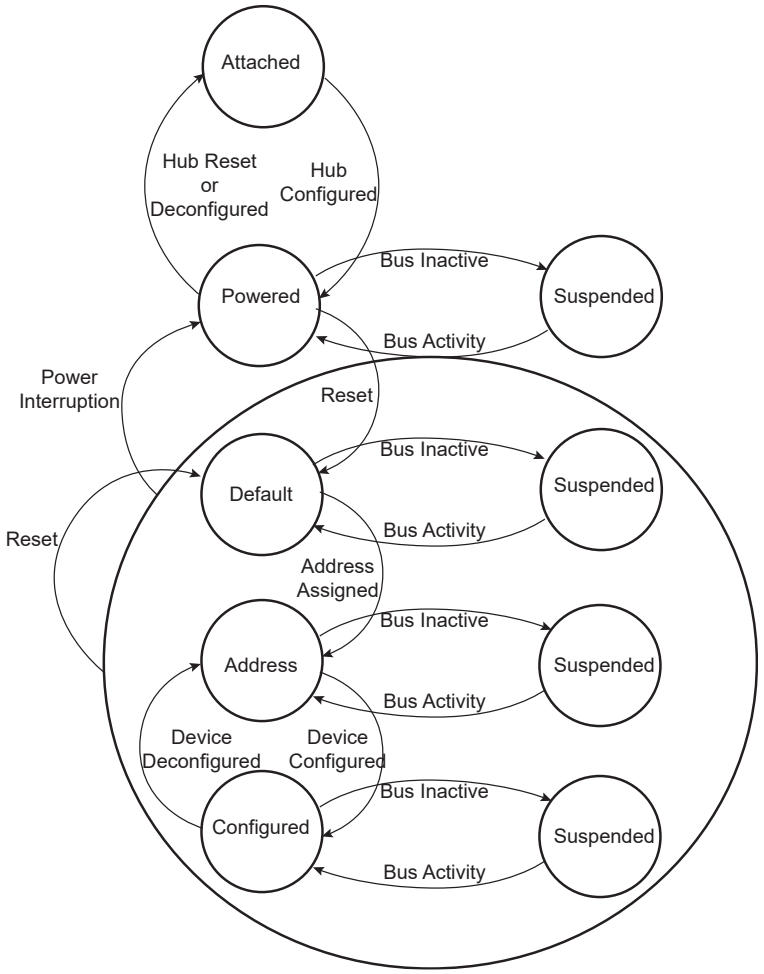


70.6.14. Power Modes

70.6.14.1.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 70.19. 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 wake-up request to the host (waking up a PC by moving a USB mouse, for example).

The wake-up feature is not mandatory for all devices and must be negotiated with the host.

70.6.14.2.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.

70.6.14.3.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.

70.6.14.4.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.

70.6.14.5.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.

70.6.14.6.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.

70.6.14.7.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.

70.6.14.8.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.

70.6.14.9.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.

70.6.15. 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
};
```

70.7. 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_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16								
		15:8		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_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
		23:16								
		15:8		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								
		7:0		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_EPTCTLENBx (DEFAULT_MODE0	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

Register Summary (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_EPTCTLDISx (DEFAULT_MODE0	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_EPTCTLx (DEFAULT_MODE0	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_EPTSETSTAx (DEFAULT_MODE0	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_EPTCLRSTAx (DEFAULT_MODE0	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_EPTSTAx (DEFAULT_MODE0	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					

Register Summary (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_EPTCTLENBx (DEFAULT_MODE1	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_EPTCTLDISx (DEFAULT_MODE1	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_EPTCTLx (DEFAULT_MODE1	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_EPTSETSTAx (DEFAULT_MODE1	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								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0138	UDPHS_EPTCLRSTAx (DEFAULT_MODE1)	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_EPTSTAx (DEFAULT_MODE1)	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_EPTCTLENBx (DEFAULT_MODE2)	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_EPTCTLDISx (DEFAULT_MODE2)	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_EPTCTLx (DEFAULT_MODE2)	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

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0150 ... 0x0153	Reserved									
0x0154	UDPHS_EPTSETSTAx (DEFAULT_MODE2)	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_EPTCLRSTAx (DEFAULT_MODE2)	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_EPTSTAx (DEFAULT_MODE2)	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_EPTCTLENBx (DEFAULT_MODE3)	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_EPTCTLDISx (DEFAULT_MODE3)	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

Register Summary (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_EPTCTLx (DEFAULT_MODE3	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_EPTSETSTAx (DEFAULT_MODE3	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_EPTCLRSTAx (DEFAULT_MODE3	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_EPTSTAx (DEFAULT_MODE3	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]		

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0184	UDPHS_EPTCTLENBx (DEFAULT_MODE4	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_EPTCTLDISx (DEFAULT_MODE4	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_EPTCTLx (DEFAULT_MODE4	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_EPTSETSTAx (DEFAULT_MODE4	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_EPTCLRSTAx (DEFAULT_MODE4	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						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x019C	UDPHS_EPTSTAx (DEFAULT_MODE4)	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_EPTCTLENBx (DEFAULT_MODE5)	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_EPTCTLDISx (DEFAULT_MODE5)	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_EPTCTLx (DEFAULT_MODE5)	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_EPTSETSTAx (DEFAULT_MODE5)	31:24								
		23:16								
		15:8					TXRDY		RXRDY_TXKL	
		7:0			FRCESTALL					

Register Summary (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_EPTCLRSTAx (DEFAULT_MODE5	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_EPTSTAx (DEFAULT_MODE5	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_EPTCTLENBx (DEFAULT_MODE6	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_EPTCTLDISx (DEFAULT_MODE6	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_EPTCTLx (DEFAULT_MODE6	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

Register Summary (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_EPTSETSTAx (DEFAULT_MODE6	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_EPTCLRSTAx (DEFAULT_MODE6	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_EPTSTAx (DEFAULT_MODE6	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 ... 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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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
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]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	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
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

70.7.1. 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 Wake-Up (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 Wake-Up is disabled (read), or this bit has no effect (write).
1	Remote Wake-Up is enabled (read), or this bit forces an external interrupt on the UDPHS controller for Remote Wake-Up 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).

70.7.2. 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).

70.7.3. UDPHS Interrupt Enable Register

Name: UDPHS_IEN
Offset: 0x10
Reset: 0x00000010
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
		DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
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
		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
Reset		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 – 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 – 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.

Value	Description
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.

70.7.4. UDPHS Interrupt Status Register

Name: UDPHS_INTSTA
Offset: 0x14
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	
Access		R	R	R	R	R	R	
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
		EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_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
	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 – 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 – 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.
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 UDPHS. 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 UDPHS. 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.

70.7.5. 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.

70.7.6. 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		EPT_6	EPT_5	EPT_4	EPT_3	EPT_2	EPT_1	EPT_0
Access		W	W	W	W	W	W	W
Reset		–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6 – 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.

70.7.7. 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.

70.7.8. UDPHS Endpoint Configuration Register

Name: UDPHS_EPTCFGx
Offset: 0x0100 + x*0x20 [x=0..6]
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

70.7.9. UDPHS Endpoint Control Enable Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTCTLENBx (DEFAULT_MODE)
Offset: 0x0104 + n*0x20 [n=0..6]
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.
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.

Value	Description
1	Enable endpoint according to the device configuration.

70.7.10. UDPHS Endpoint Control Enable Register (Isochronous Endpoints)

Name: UDPHS_EPTCTLENBx
Offset: 0x0104 + x*0x20 [x=0..6]
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.

Value	Description
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.

70.7.11. UDPHS Endpoint Control Disable Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTCTLDISx (DEFAULT_MODE)
Offset: 0x0108 + n*0x20 [n=0..6]
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, the configuration is:

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.

Value	Description
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.

70.7.12. UDPHS Endpoint Control Disable Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLDISx
Offset: 0x0108 + x*0x20 [x=0..6]
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.
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.

70.7.13. UDPHS Endpoint Control Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTCTLx (DEFAULT_MODE)
Offset: 0x010C + n*0x20 [n=0..6]
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.

70.7.14. UDPHS Endpoint Control Register (Isochronous Endpoint)

Name: UDPHS_EPTCTLx
Offset: 0x010C + x*0x20 [x=0..6]
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_NT	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.

Value	Description
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.

70.7.15. UDPHS Endpoint Set Status Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTSETSTAx (DEFAULT_MODE)
Offset: 0x0114 + n*0x20 [n=0..6]
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

70.7.16. UDPHS Endpoint Set Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSETSTAx
Offset: 0x0114 + x*0x20 [x=0..6]
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.

70.7.17. UDPHS Endpoint Clear Status Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTCLRSTAx (DEFAULT_MODE)
Offset: 0x0118 + n*0x20 [n=0..6]
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.

70.7.18. UDPHS Endpoint Clear Status Register (Isochronous Endpoint)

Name: UDPHS_EPTCLRSTAx
Offset: 0x0118 + x*0x20 [x=0..6]
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

70.7.19. UDPHS Endpoint Status Register (Control, Bulk, Interrupt Endpoints) (Default Mode)

Name: UDPHS_EPTSTAx (DEFAULT_MODE)
Offset: 0x011C + n*0x20 [n=0..6]
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

Value	Name	Description
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.

70.7.20. UDPHS Endpoint Status Register (Isochronous Endpoint)

Name: UDPHS_EPTSTAx
Offset: 0x011C + x*0x20 [x=0..6]
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_NT	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)

70.7.21. 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.

70.7.22. UDPHS DMA Next Descriptor Address Register

Name: UDPHS_DMANXTDSCx
Offset: 0x0310 + (x-1)*0x10 [x=1..6]
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.

70.7.23. UDPHS DMA Channel Address Register

Name: UDPHS_DMAADDRESSx
Offset: 0x0314 + (x-1)*0x10 [x=1..6]
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.

70.7.24. UDPHS DMA Channel Control Register

Name: UDPHS_DMACONTROLx
Offset: 0x0318 + (x-1)*0x10 [x=1..6]
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.
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.

70.7.25. UDPHS DMA Channel Status Register

Name: UDPHS_DMASTATUSx
Offset: 0x031C + (x-1)*0x10 [x=1..6]
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.

Value	Description
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.
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.

71. USB Host High Speed Port (UHPHS)

71.1. Description

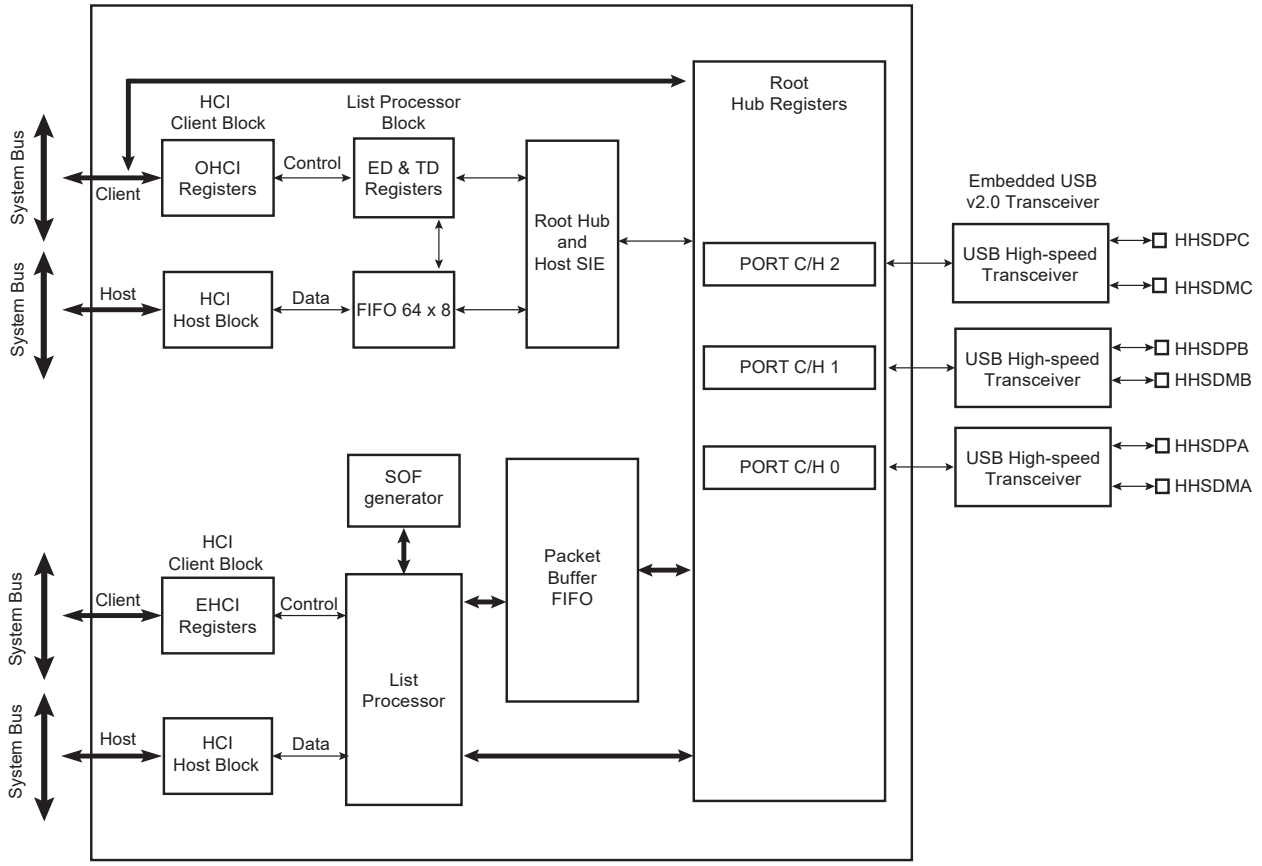
The USB Host High Speed Port (UHPHS) interfaces the USB with the host application. It handles Open HCI protocol (Open Host Controller Interface) as well as Enhanced HCI protocol (Enhanced Host Controller Interface).

71.2. 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 2 Downstream USB HS Ports and 1 FS Port
- Embedded USB Transceivers
- Supports Power Management
- 3 Hosts (A, B, and C) High Speed (EHCI), Port A shared with UDPHS

71.3. Block Diagram

Figure 71.1. 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

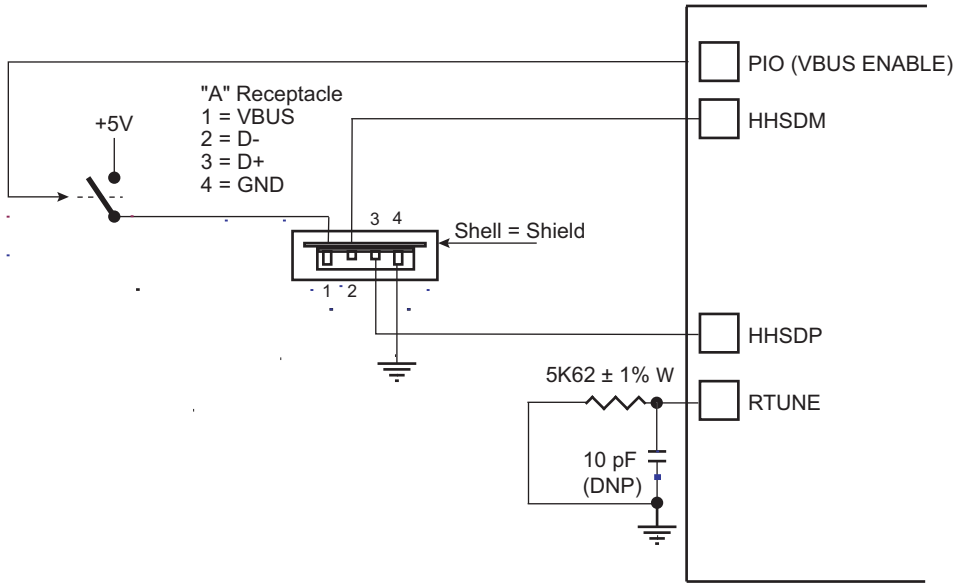
Memory access errors (abort, misalignment) lead to an “Unrecoverable Error” indicated by the corresponding flag in the host controller operational registers.

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.

71.4. Typical Connection

Figure 71.2. Board Schematic to Interface UHP High-speed Host Controller



71.5. Product Dependencies

71.5.1. I/O Lines

HHSDPs and HHSDMs are not controlled by any PIO controllers. The embedded USB High Speed physical transceivers are controlled by the USB host controller.

One transceiver is shared with the USB High Speed Device (port A). The selection between Host Port A and USB Device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

In the case the port A is driven by the USB High Speed Device, the output signals are DHSDP and DHSDM. The transceiver is automatically selected for Device operation once the USB High Speed Device is enabled.

In the case the port A is driven by the USB High Speed Host, the output signals are HHSDPA and HHSDMA.

71.5.2. Power Management

The system embeds three transceivers.

The USB Host High Speed requires a 480 MHz clock for the embedded High-speed transceivers. This clock (UPLLCK) is provided by the UTMI PLL.

In case power consumption is saved by stopping the UTMI PLL, high-speed operations are not possible. Nevertheless, OHCI Full-speed operations remain possible by selecting PLLACK as the input clock of OHCI.

The High-speed transceiver returns a 30 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.

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.

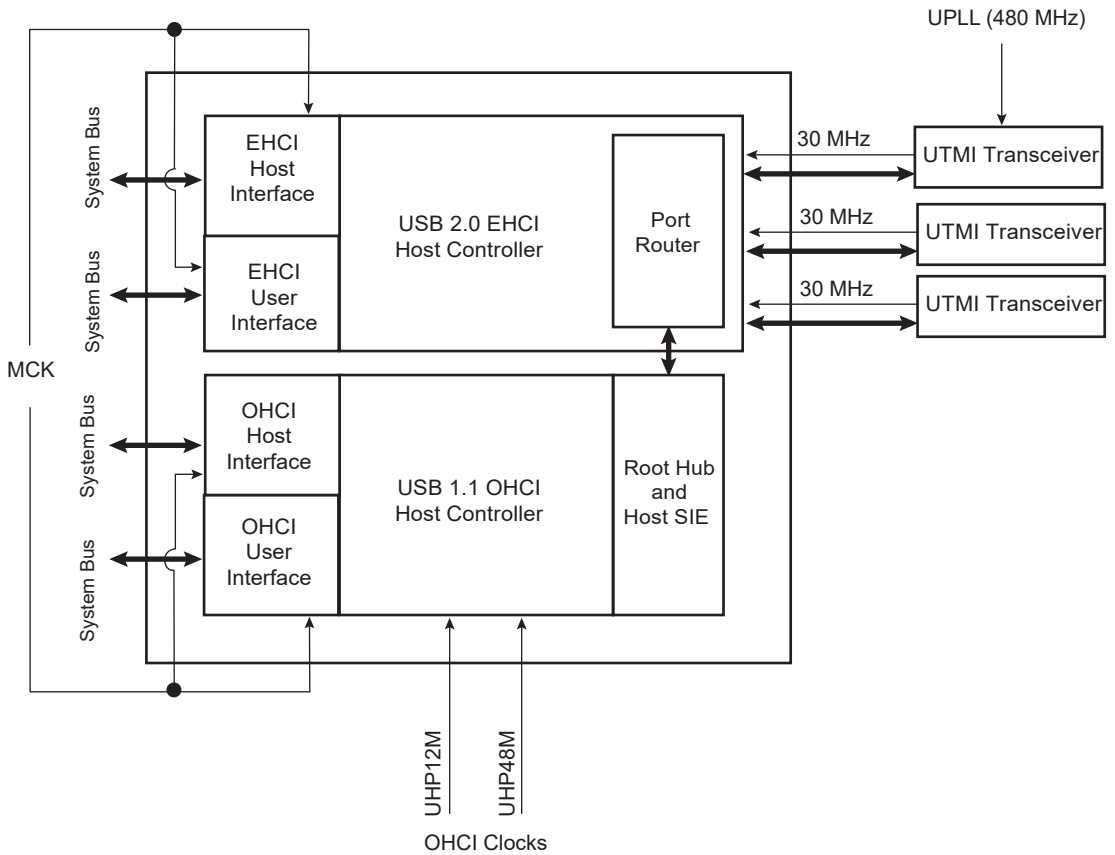
For High-speed operations, the user has to perform the following:

- Enable UHPHS peripheral clock in the PMC.
- Start the UPLL with a 480 MHz output frequency following the recommendation provided in section "USB Clock Controller" in chapter "Power Management Controller (PMC)".
- Select UPLLCK as Input clock of OHCI part (USBS bit in PMC_USB register).
- Program OHCI clocks (UHP48M and UHP12M) with USBDIV field in PMC_USB register. USBDIV must be 9 (division by 10) if UPLLCK is selected.
- Enable OHCI clocks with UHP bit in PMC_SCER.

For OHCI Full-speed operations only, the user can use the PLLA, and in this case has to perform the following:

- Enable UHPHS peripheral clock in the PMC.
- Select PLLACK as Input clock of OHCI part (USBS bit in PMC_USB register).
- Program OHCI clocks (UHP48M and UHP12M) with USBDIV field in PMC_USB register. USBDIV value is to be calculated according to the PLLACK value and USB Full-speed accuracy.
- Enable the OHCI clocks with UHP bit in PMC_SCER.

Figure 71.3. UHP Clock Trees



71.5.3. Interrupt Sources

The USB host interface has an interrupt line connected to the interrupt controller.

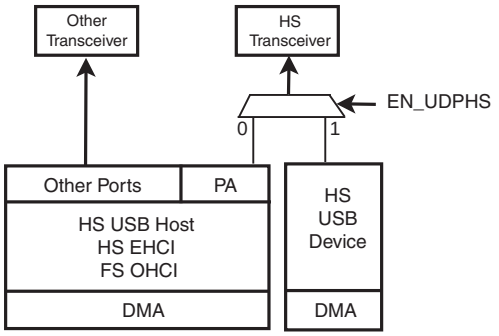
Handling USB host interrupts requires programming the interrupt controller before configuring the UHPHS.

71.6. Functional Description

71.6.1. UTMI Transceivers Sharing

The High Speed USB Host Port A is shared with the High Speed USB Device port and connected to the second UTMI transceiver. The selection between Host Port A and USB device is controlled by the UDPHS enable bit (EN_UDPHS) located in the UDPHS_CTRL register.

Figure 71.4. USB Selection



71.6.2. 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

71.6.3. 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.

71.7. 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									

Register Summary (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]							

71.7.1. 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.

71.7.2. 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.

71.7.3. 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

71.7.4. 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.

71.7.5. 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).

71.7.6. 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.

71.7.7. 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.

71.7.8. 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.

71.7.9. 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).

71.7.10. 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.

71.7.11. 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.
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.

71.7.12. 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.

71.7.13. 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.

72. Analog-to-Digital Controller (ADC)

72.1. 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. See the Block Diagram. It also integrates a 8-to-1 analog multiplexer, making possible the analog-to-digital conversions of 8 analog lines. The conversions extend from the voltage on pin ADVREFN 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 software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

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 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, pseudo-differential or fully differential input.

This ADC Controller includes a Resistive Touchscreen Controller. It supports 4-wire and 5-wire technologies.

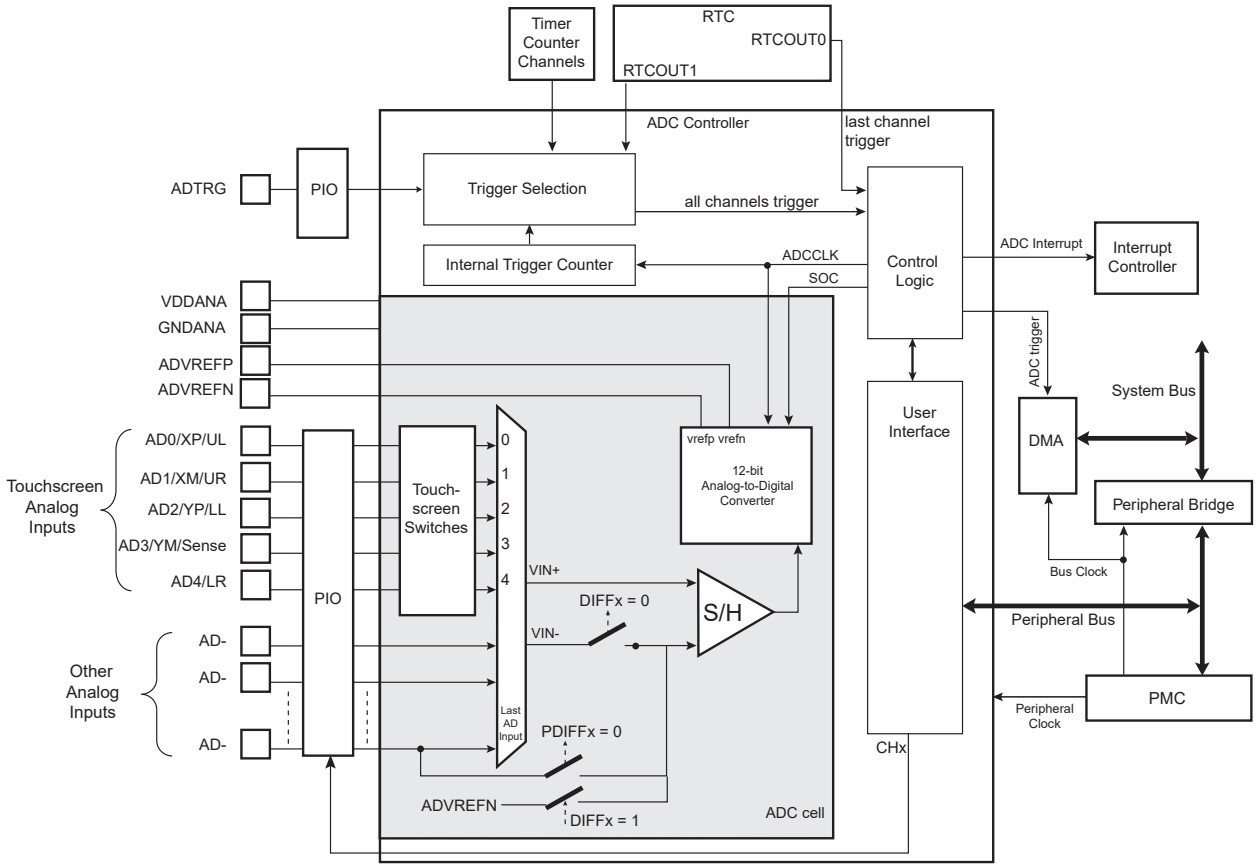
72.2. Embedded Characteristics

- 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
- Wide Range of Power Supply Operation
- Selectable Single-Ended, Pseudo-Differential or Differential Input Voltage
- Digital correction of offset and gain errors
- Resistive 4-wire and 5-wire Touchscreen Controller
  - Position and pressure measurement for 4-wire screens
  - Position measurement for 5-wire screens
  - Average of up to 8 measures for noise filtering
- Programmable Pen Detection Sensitivity
- Integrated Multiplexer Offering Up to 8 Independent Analog Inputs
- Individual Enable and Disable of Each Channel
- Hardware or Software Trigger from:

  - External trigger pin
  - Timer counter outputs (corresponding TIOA trigger)
  - ADC internal trigger counter
  - Trigger on pen contact detection
  - 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

72.3. Block Diagram

Figure 72.1. ADC Block Diagram



72.4. Signal Description

Table 72.1. ADC Pin Description

Pin Name	Description
VDDANA	Analog power supply

Table 72.1. ADC Pin Description (continued)

Pin Name	Description
GNDANA	Analog ground supply
ADVREFP	Reference voltage
ADVREFN	Reference voltage
AD[7:0]	Analog input channels
ADTRG	External trigger

72.5. Product Dependencies

72.5.1. 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.

72.5.2. 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.

72.5.3. I/O Lines

The digital inputs ADx and ADTRG are multiplexed with digital functions on the I/O lines.

ADx inputs are selected as inputs of the ADCC when writing a one in the corresponding ADC_CHER.CHx bit and the associated I/O is automatically turned in Analog mode.

72.5.4. Hardware Triggers

The ADC can use internal signals to start conversions. See the ADC_MR.TRGSEL field description in ADC_MR for exact wiring of internal triggers.

72.5.5. Fault Output

The ADC Controller has the FAULT output connected to the FAULT input of PWM. See Fault Event and section “Pulse Width Modulation Controller (PWM)”.

72.6. Functional Description

72.6.1. 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 the TRACKTIM field in the Mode register (ADC_MR) and the TRACKX4 bit 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 72.2. Sequence of Consecutive ADC Conversions with TRACKTIM = 0

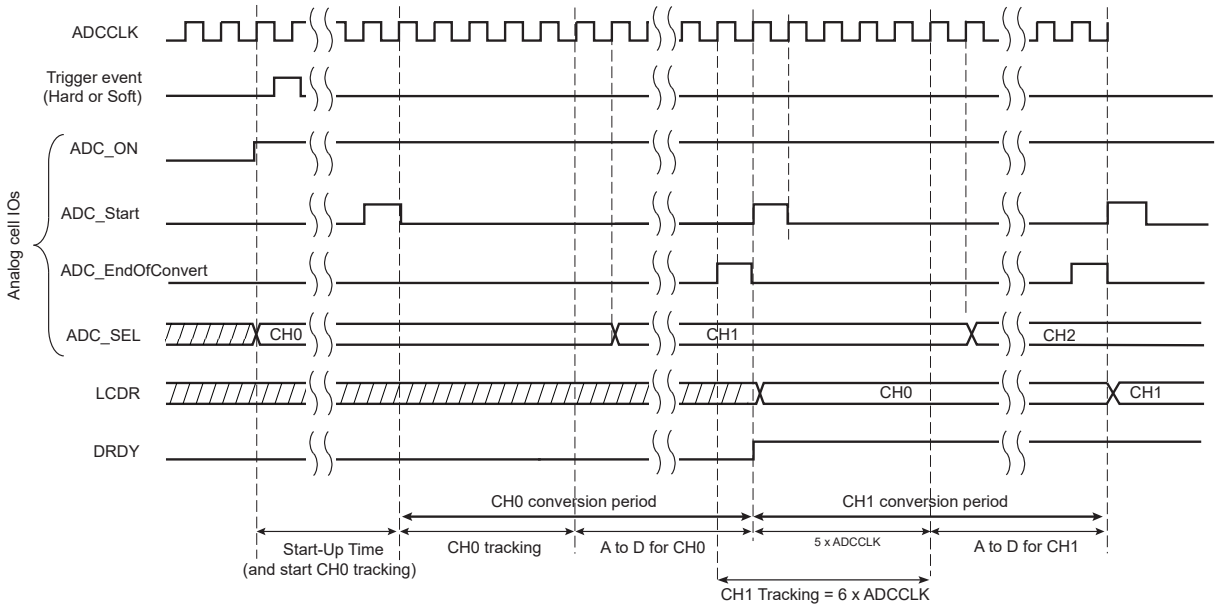
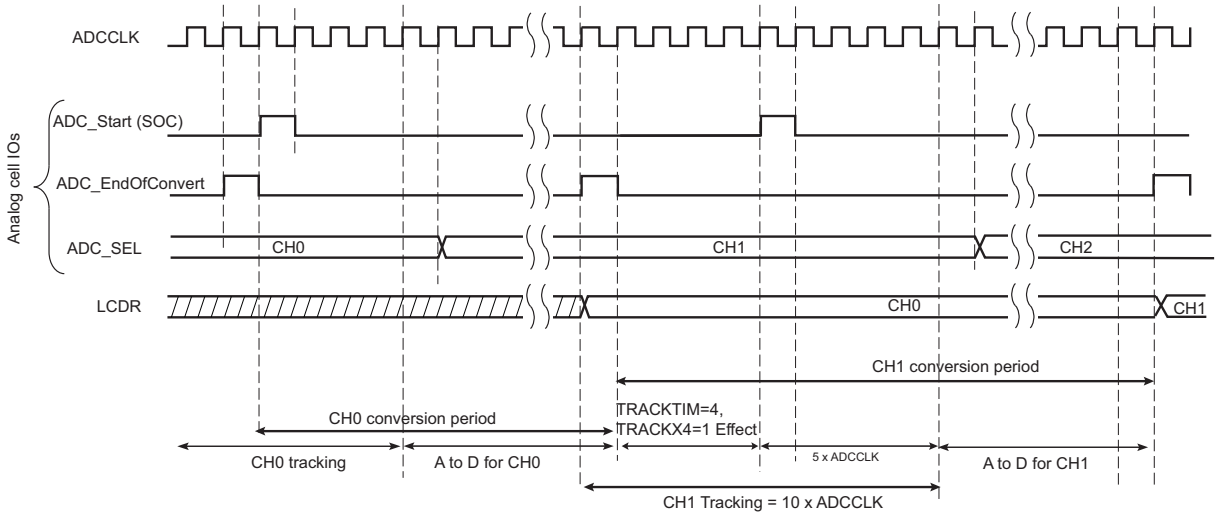


Figure 72.3. Sequence of Consecutive ADC Conversions with TRACKTIM = 4 and TRACK4X = 1



72.6.2. ADC Clock

The ADC uses the ADC clock (ADCCLK) to perform conversions. The ADC clock frequency is selected in the ADC_MR.PRESCAL.

To generate the ADC clock, the prescaler has two clock sources: the peripheral clock and the GCLK clock. This clock source is selected using the SRCCLK bit in the Extended Mode register (ADC_EMR).

If GCLK is selected as a source clock, the ADC clock frequency is independent of the processor/bus clock. At reset, the peripheral clock is selected.

If ADC_EMR.SRCCLK is cleared, the prescaler clock (presc_clk) is driven by peripheral_clock. If ADC_EMR.SRCCLK is set, the prescaler clock is driven by GCLK. The ADC clock frequency is between fpresc_clk/2, if PRESCAL is 0, and fpresc_clk/512, if PRESCAL is set to 255 (0xFF).

PRESCAL must be programmed to provide the ADC clock frequency parameter provided in the “Electrical Characteristics” section.

72.6.3. ADC Reference Voltage

The voltage reference input of the ADC is the ADVREFP pin and the negative reference voltage is ADVREFN. Refer to the section “Electrical Characteristics”.

72.6.4. 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 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.

The same applies when using the independent clock (ADC_MR.SRCCLK = 1), if the provided clock is asynchronous to ADC peripheral clock.

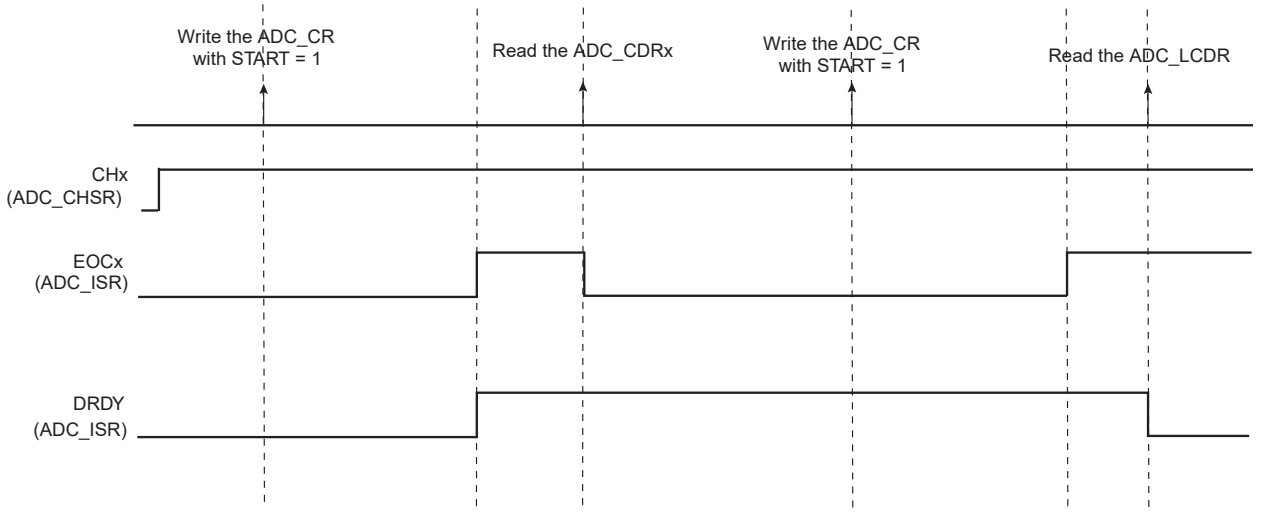
72.6.5. Conversion Results

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 the TAG option in ADC_EMR, ADC_LCDR presents the channel number associated with the last converted data in the CHNB field.

When a conversion is completed, the channel EOC bit and the DRDY bit 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 and DRDY can trigger an interrupt.

Reading one of the ADC_CDRx clears the corresponding EOC bit. Reading ADC_LCDR clears the DRDY bit.

Figure 72.4. EOCx and DRDY Flag Behavior

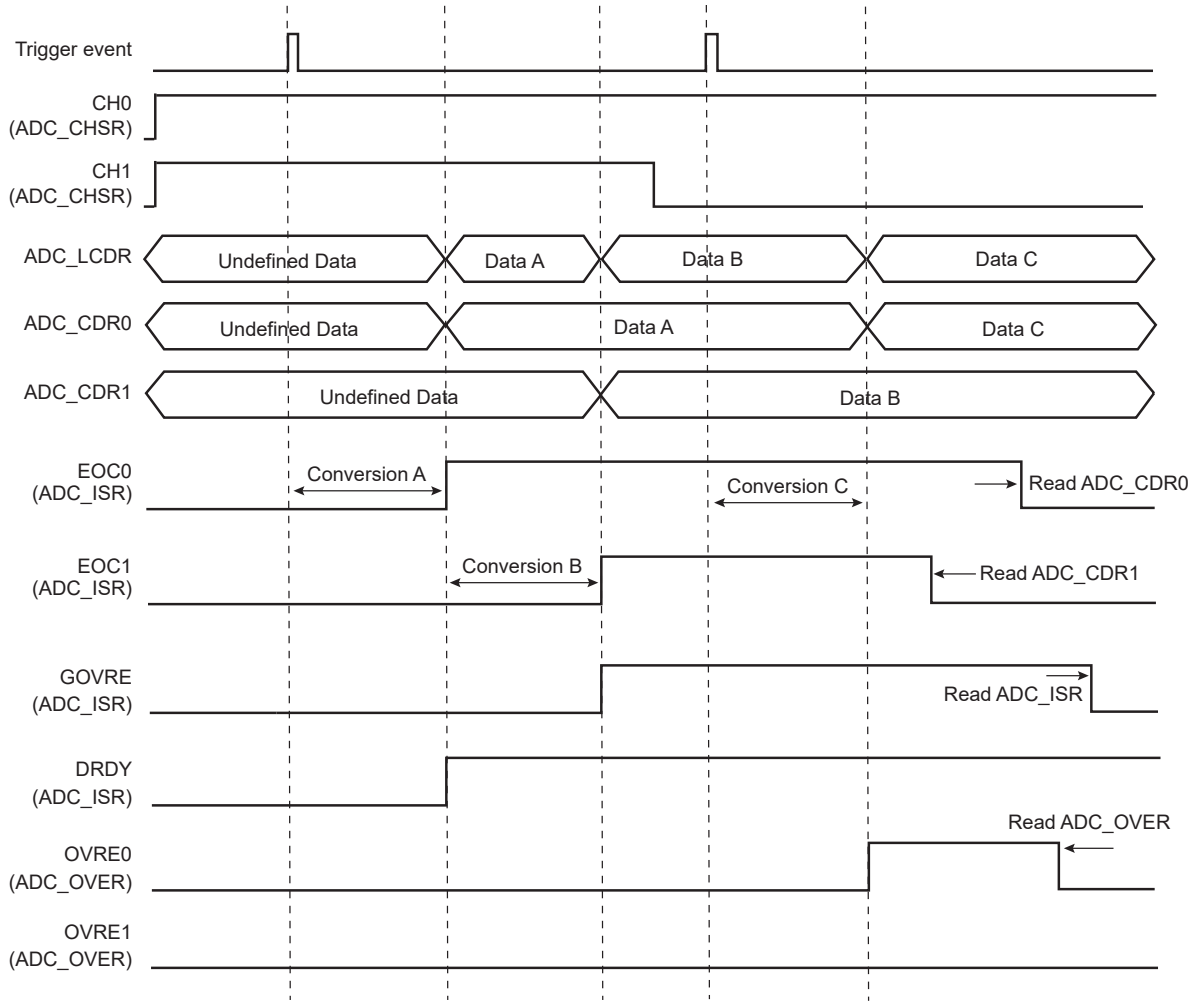


If ADC_CDR is not read before further incoming data is converted, the corresponding OVREx flag is set in the Overrun Status register (ADC_OVER).

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 72.5. EOCx, OVREx and GOVREx Flag Behavior



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 EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable.

72.6.6. Conversion Results Format

The conversion results can be signed (2's complement) or unsigned depending on the value of the ADC_EMR.SIGNMODE field.

If conversion results are signed and resolution is less than 16 bits, the sign is extended up to the bit 15 (for example, 0xF43 for 12-bit resolution is read as 0xFF43, and 0x467 is read as 0x0467).

72.6.7. 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 the START bit at 1 and ADC_TRGR.TRGMOD=0.

The list of external/internal events is provided in ADC_MR. The hardware trigger is selected using the ADC_MR.TRGSEL field. The selected hardware trigger is enabled if TRGMOD = 1, 2 or 3 in the Trigger

register (ADC_TRGR). In these modes, the software trigger is disabled (writing ADC_CR.START=1 has no effect).

The ADC also provides a Dual Trigger mode (ADC_LCTMR.DUALTRIG = 1) in which the higher 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 Last Channel Specific Measurement Trigger.

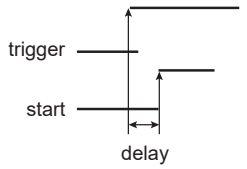
The ADC_TRGR.TRGMOD field selects the hardware trigger from the following:

- Any edge, either rising or falling or both, detected on the external trigger pin ADTRG or internal triggers
- The Pen Detect, depending on how the PENDET bit is set in the Touchscreen Mode register (ADC_TSMR)
- A continuous trigger, meaning the ADC Controller restarts the next sequence as soon as it finishes the current one
- A periodic trigger, which is defined by programming the ADC_TRGR.TRGPER field

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_CHSR, ADC_SEQR1, and ADC_TSMR.

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 72.6. Hardware Trigger Delay



If one of the TIOA outputs is selected, the corresponding Timer Counter channel must be programmed in Waveform mode.

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, only the transfers of converted data from enabled channels are performed and the resulting data buffers should be interpreted accordingly.

72.6.8. 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 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.

The sequence can be customized by programming the Sequence Channel registers ADC_SEQR1 and setting the USEQ bit of the Mode register (ADC_MR). The user can choose a specific order of channels and can program up to 8 conversions by sequence. The user is free to create a personal sequence by writing channel numbers in ADC_SEQR1. 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 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 all ADC channels (i.e., 8) 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). For example, if channel 4 is disabled (ADC_CHSR[4] = 0), ADC_SEQR1 fields USCH1 up to USCH8 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 over 8 (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.

72.6.9. 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 the ADC_EMR.CMPMODE field. The comparison can be done on all channels or only on the channel specified in the ADC_EMR.CMPSEL field. 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 the ADC_EMR.CMPFILTER field. The filtering option is dedicated to reinforcing the detection of an analog signal overpassing a predefined threshold. The filter is cleared as soon as ADC_ISR is read, so this filtering function must be used with peripheral 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 the ADC_EMR.SIGNMODE field, 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, the ADC_EMR.SIGNMODE field must be set to ALL_UNSIGNED or ALL_SIGNED and the thresholds must be set accordingly.

72.6.10. Pseudo-Differential, Differential and Single-Ended Input Modes

72.6.10.1.Input-Output Transfer Functions

The ADC can be configured to operate in the following input voltage modes:

- Single-Ended—ADC_CCR.DIFFx = 0 and ADC_PDR.PDIFFx = 0. This is the default mode after a reset.
- Differential—ADC_CCR.DIFFx = 1 and ADC_PDR.PDIFFx = 0 (see the figure below). In Differential mode, the ADC requires differential input signals having a VDD/2 common mode voltage (refer to the section “Electrical Characteristics”).
- • Pseudo-Differential—ADC_PDR.PDIFFx = 1. In Pseudo-Differential mode, one of the analog pins is used as the negative input of the ADC (see table Input Pins and Channel Numbers in Pseudo-Differential Mode). The ADC samples the other analog inputs with respect to this one. The common mode input range is 0 to VDD (refer to the section “Electrical Characteristics”).

The following equations provide the unsigned ADC input-output transfer function in each mode(1). With signed conversions (see field ADC_EMR.SIGNMODE), subtract 2047 from the ADC_LCDR.DATA value given below. Note that the Single-Ended mode introduces a x2 gain compared to the Pseudo-Differential mode.

In the formula, REFP = VREFP, REFN = VREFN.

Single-Ended mode:

$$\text{ADC\_LCDR.LDATA} = \frac{\text{ADx} - \text{REFN}}{\text{REFP} - \text{REFN}} \times 2^{12}$$

Differential mode:

$$\text{ADC\_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{ADx+1}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

Pseudo-Differential mode:

$$\text{ADC\_LCDR.LDATA} = \left(1 + \frac{\text{ADx} - \text{AD7}}{\text{REFP} - \text{REFN}}\right) \times 2^{11}$$

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 tables provide the internal positive and negative ADC inputs assignment with respect to the programmed mode (ADC_CCR.DIFFx and ADC_PDR.PDIFFx).

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.

Table 72.2. Input Pins and Channel Numbers in Single-Ended and Differential Modes

Internal ADC Inputs (VIN+, VIN-)		Channel Number	
Single-Ended Mode	Differential Mode	Single-Ended Mode	Differential Mode
AD0, ADVREFN	AD0, AD1	CH0	CH0
AD1, ADVREFN	–	CH1	
AD2, ADVREFN	AD2, AD3	CH2	CH2
AD3, ADVREFN	–	CH3	

Table 72.2. Input Pins and Channel Numbers in Single-Ended and Differential Modes (continued)

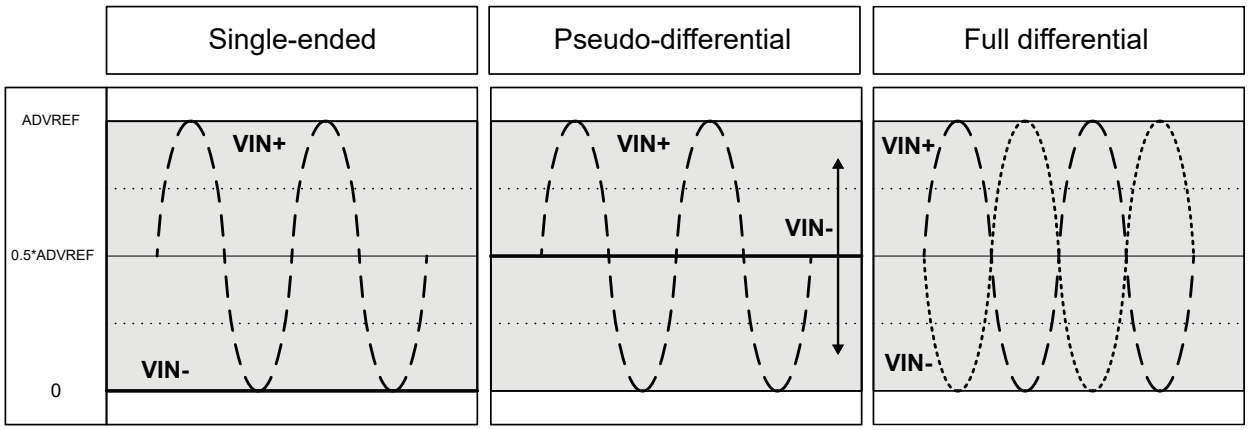
Internal ADC Inputs (VIN+, VIN-)		Channel Number	
Single-Ended Mode	Differential Mode	Single-Ended Mode	Differential Mode
AD4, ADVREFN	AD4, AD5	CH4	CH4
AD5, ADVREFN	–	CH5	
AD6, ADVREFN	AD6, AD7	CH6	CH6
AD7, ADVREFN	–	CH7	

In Pseudo-Differential mode, inputs are managed by a 8/2:1-channel analog multiplexer. See the table below.

Table 72.3. Input Pins and Channel Numbers in Pseudo-Differential Mode

Internal ADC Inputs (VIN+, VIN-)	Channel Number
AD0, AD7	CH0
AD1, AD7	CH1
AD2, AD7	CH2
AD3, AD7	CH3
AD4, AD7	CH4
AD5, AD7	CH5
AD6, AD7	CH6
GNDANA, AD7	CH7

Figure 72.7. Analog Full Scale Ranges in Single-Ended/Pseudo-Differential/Differential Applications



72.6.11. ADC Timings

The ADC start-up time is programmed through the ADC_MR.STARTUP field. 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 guarantee 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 using the ADC_MR.TRACKTIM field and ADC_EMR.TRACKX4.

WARNING No input buffer amplifier to isolate the source is included in the ADC. Refer to the section "Electrical Characteristics".

72.6.12. Last Channel Specific Measurement Trigger

The last channel (higher index available) embeds a specific mode allowing a measurement trigger period which differs from other active channels. This allows efficient management of the conversions especially if the channel is driven by a device with a variation of a different frequency from other converted channels (for example, but not limited to, temperature sensor).

The last channel can be sampled in different ways through the ADC Controller. The different methods of sampling depend on the ADC_TRGR.TRGMOD configuration field and on ADC_CHSR.CH7.

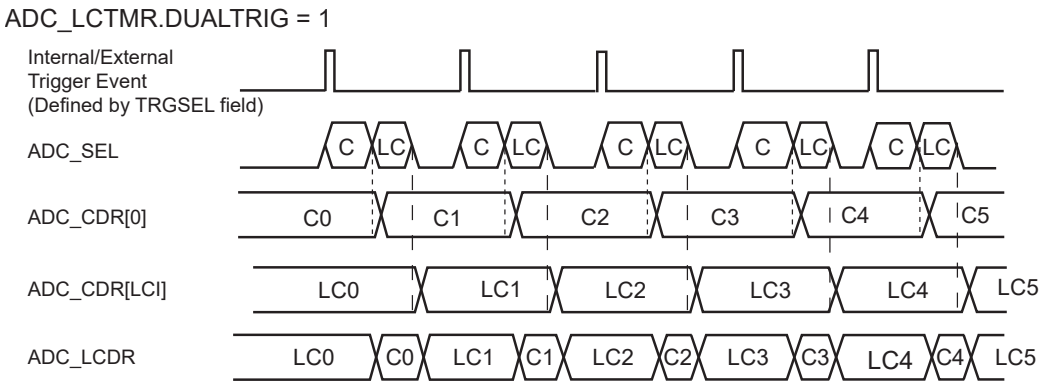
The last channel conversion can be triggered like the other channels by enabling ADC_CHER.CH7.

The manual start can only be performed if field TRGMOD = 0. When ADC_CR.START is set, the last channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in the ADC_CDR7 register, and the associated ADC_ISR.EOC7 flag is set.

If the last channel is enabled in the Channel Status register (ADC_CHSR), ADC_LCTMR.DUALTRIG is cleared and field TRGMOD = 1, 2, 3, 5, the last channel is periodically converted together with the other enabled channels and the result is placed in the ADC_LCDR and ADC_CDR7 registers. Thus the last channel conversion result is part of the DMA Controller buffer (see the following figure).

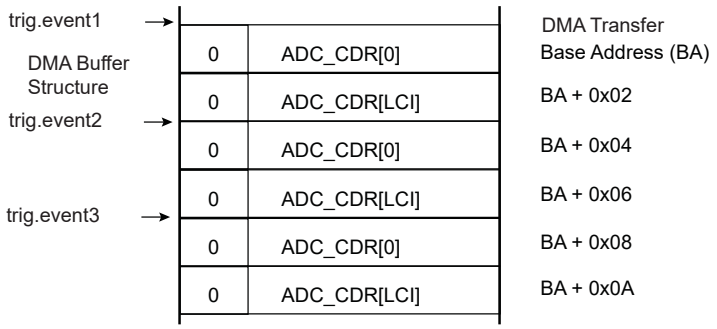
When the conversion result matches the conditions defined in ADC_LCTMR and ADC_LCCWR, the ADC_ISR.LCCHG flag is set.

Figure 72.8. Same Trigger for All Channels (ADC_CHSR[LCI] = 1 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)



Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index

Assuming ADC_CHSR[0] = 1 and ADC_CHSR[LCI] = 1



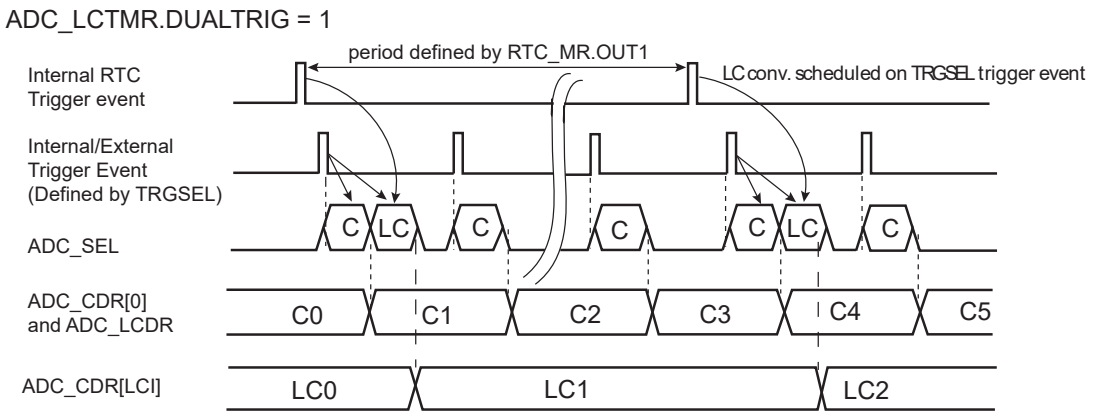
If the last channel is driven by a device with a slower variation compared to other channels (temperature sensor for example), the channel can be enabled/disabled at any time. However, this may not be optimal for downstream processing.

The ADC Controller allows a different way of triggering the measurement when DUALTRIG is set in the Last Channel Trigger Mode register (ADC_LCTMR) but CH7 is not set in ADC_CHSR.

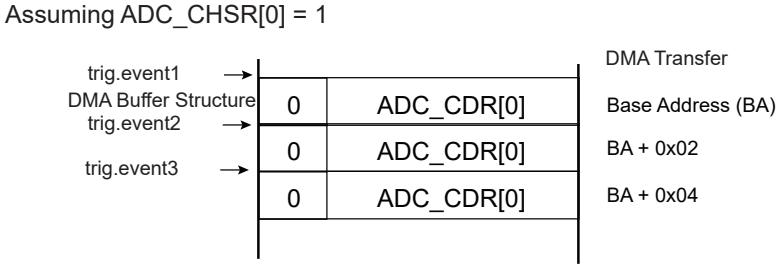
Under these conditions, the last channel conversion is triggered with a period defined by the field RTC_MR.OUTx (see Block Diagram for the value of 'x') while other channels are still active and triggered by internal/external triggers. The RTC event is processed on the next internal/external trigger event, as shown in the following figure. The internal/external trigger for other channels is selected through the ADC_MR.TRGSEL field.

When DUALTRIG = 1, the result of each conversion of channel 7 is only uploaded in the ADC_CDR7 register and not in ADC_LCDR (see the following figure). Therefore, there is no change in the structure of the peripheral DMA controller buffer due to the conversion of the last channel: only the enabled channels are kept in the buffer. The end of conversion of the last channel is reported by the ADC_ISR.EOC7 flag.

Figure 72.9. Independent Trigger Measurement for Last Channel (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 1, 2, 3, 5)



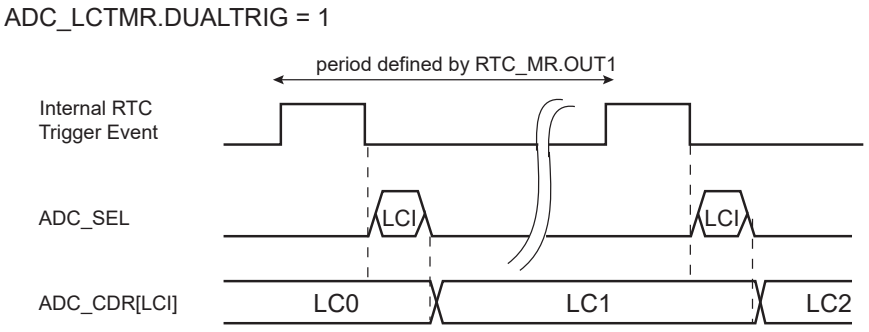
Notes: ADC_SEL: Command to the ADC analog cell
Cx: All ADC channel values except the last channel (highest index)
LCx: Last channel value
LCI: Last channel index



If DUALTRIG = 1 and field ADC_TRGR.TRGMOD = 0 and none of the channels are enabled in ADC_CHSR (ADC_CHSR = 0), then only channel 7 is converted at a rate defined by the trigger event signal that can be configured in RTC_MR.OUT1 (see the following figure).

This mode of operation, when combined with the Sleep mode operation of the ADC Controller, provides a low-power mode for last channel measure. This assumes there is no other ADC conversion to schedule at a high sampling rate or no other channel to convert.

Figure 72.10. Only Last Channel Measurement Triggered at Low Speed (ADC_CHSR[LCI] = 0 and ADC_TRGR.TRGMOD = 0)



Notes: ADC_SEL: Command to the ADC analog cell
LCx: Last channel value
LCI: Last channel index

72.6.13. Enhanced Resolution Mode and Digital Averaging Function

72.6.13.1.Enhanced Resolution Mode

The Enhanced Resolution mode is enabled if the OSR field is configured to 1, 2, 3 or 4 in ADC_EMR. 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.

$$\text{ADC\_LCDR.LDATA} = \frac{1}{M} \times \sum_{k=0}^{k=N-1} \text{ADC}(k)$$

where N and M are as shown in the table below.

Table 72.4. 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	16380
3	15 bits	64	8	32767	32760
4	16 bits	256	16	65535	65520

The average result is valid in ADC_CDRx (x corresponds to the index of the channel) only if the ADC_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.

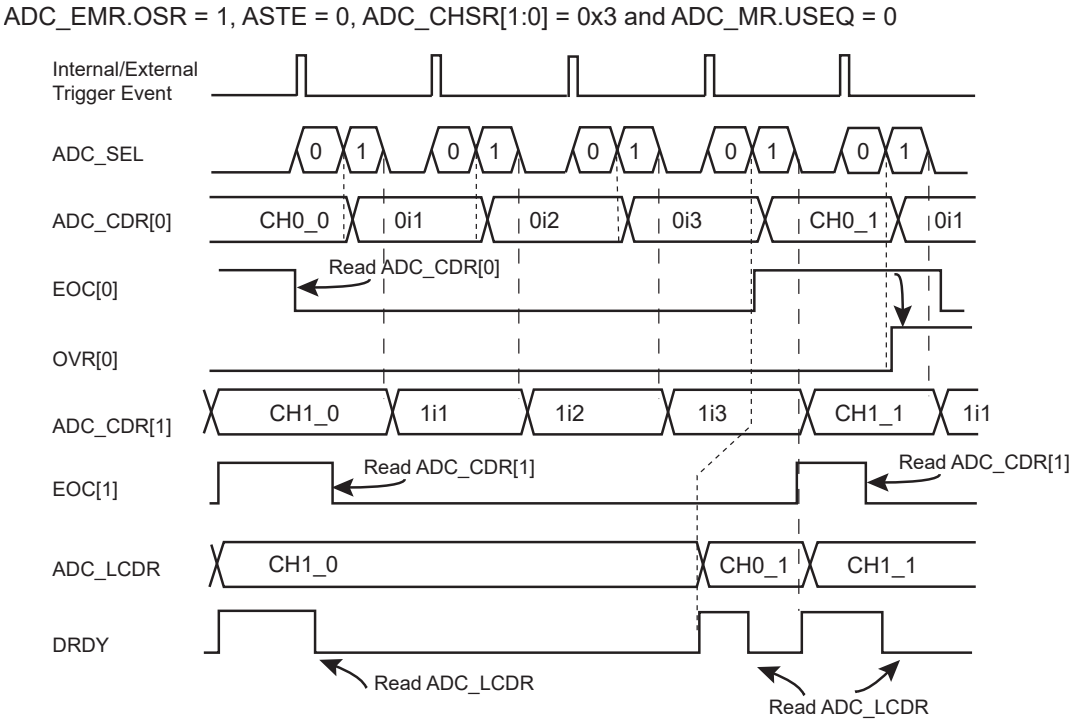
When an oversampling is performed, the maximum value that can be read on ADC_CDRx or ADC_LCDR is not the full-scale value, even if the maximum voltage is supplied on the analog input. See table above.

72.6.13.2.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 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 required to obtain the result of averaging if OSR = 1.

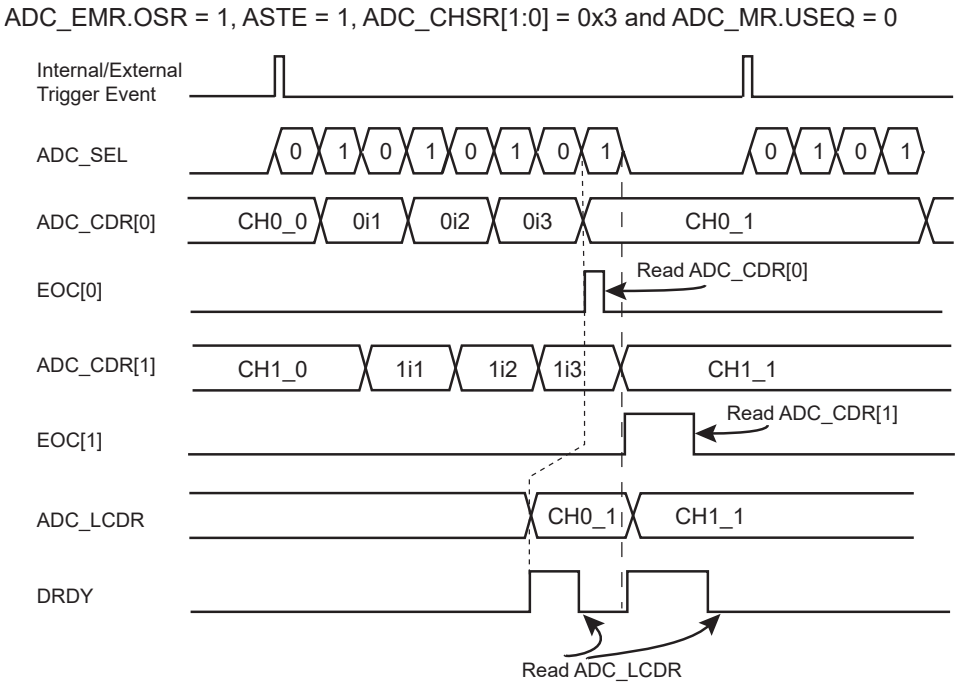
Figure 72.11. Digital Averaging Function Waveforms Over Multiple Trigger Events



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 the ADC_EMR.OSR field). As a result, only one trigger is required to obtain the result of the averaging function as described in the following figure.

Figure 72.12. 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_SEQR1 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 figure below.

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.

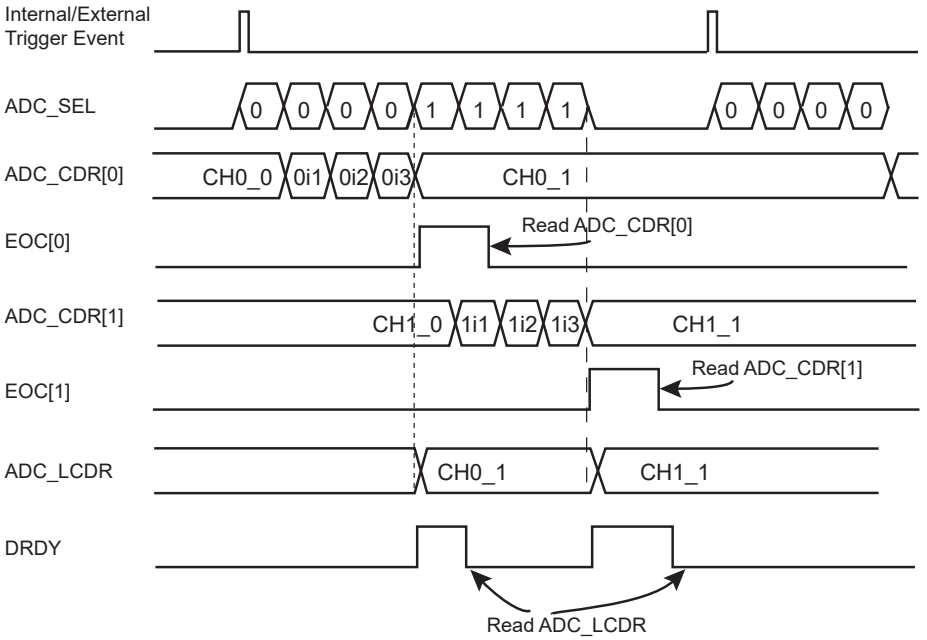
It is important that the user sequence follows a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a same channel is successively converted.

Table 72.5. 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	0x0000_000F	0x0000_00FF	0x0000_0FFF
ADC_SEQR1	0x0000_0000	0x1111_0000	0x1111_0000

Figure 72.13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

ADC_EMR.OSR = 1, ASTE = 1, ADC_CHSR[7:0] = 0xFF and ADC_MR.USEQ = 1
ADC_SEQR1 = 0x1111_0000



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.

72.6.14. 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 .

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 error correction for channels used with the touchscreen is available in the ADC Touchscreen Correction Values register (ADC_TSCVR).

The ADC_EMR.ADCMODE field is used to configure a running mode of the ADC Normal mode, Offset Error mode, or Gain Error mode (see ADC_EMR). ADCMODE uses 3 internal references 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 running mode of the ADC 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 table below 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 15
- 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 RESOLUTION+2, RESOLUTION+3, RESOLUTION+4).

Table 72.6. 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 – $2^{(Resolution}$ – 1)
		For signed conversions: OFFSETCORR = ConvValue
2	Gain Error	GCi = ConvValue
3		$GAINCORR = \frac{3584}{GCi - ConvValue} \times 2^{(Gs)}$

The final conversion result after error correction is obtained using the following formula:

$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(Gs)}}$$

72.6.15. Touchscreen

72.6.15.1.Touchscreen Mode

The ADC_TSMR.TSMODE parameter is used to enable/disable the touchscreen functionality, to select the type of screen (4-wire or 5-wire) and, in the case of a 4-wire screen, to activate (or not) the pressure measurement.

In 4-wire mode, channels 0, 1, 2 and 3 must not be used for classic ADC conversions. Likewise, in 5-wire mode, channels 0, 1, 2, 3, and 4 must not be used for classic ADC conversions.

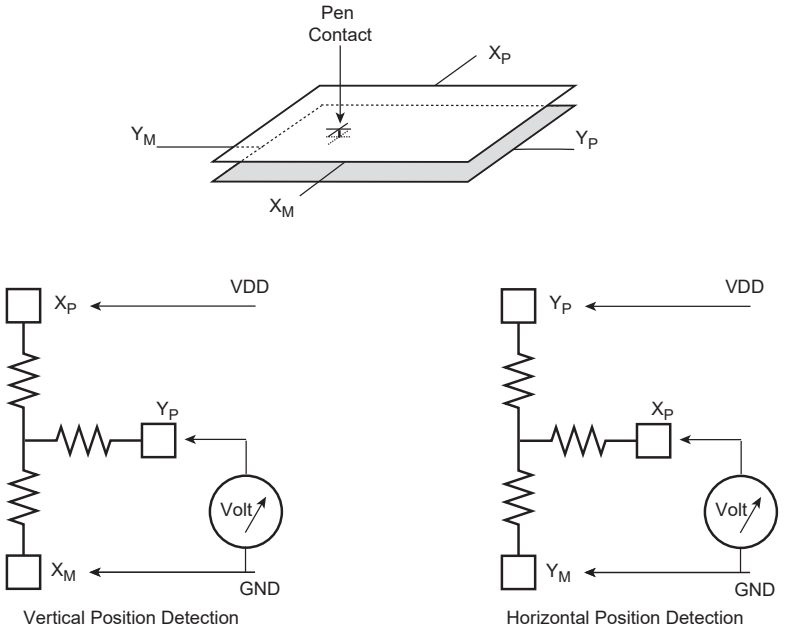
72.6.15.2.4-wire Resistive Touchscreen Principles

A resistive touchscreen is based on two resistive films, each one being fitted with a pair of electrodes, placed at the top and bottom on one film, and on the right and left on the other. In between, there is a layer acting as an insulator, but also enables contact when you press the screen. This is illustrated in the following figure.

The ADC Controller can perform the following tasks without external components:

- position measurement
- pressure measurement
- pen detection

Figure 72.14. Touchscreen Position Measurement



72.6.15.3.4-wire Position Measurement Method

As shown in the above figure, to detect the position of a contact, a supply is first applied from top to bottom. Due to the linear resistance of the film, there is a voltage gradient from top to bottom. When a contact is performed on the screen, the voltage propagates at the point the two surfaces come into contact. If the input impedance on the right and left electrodes is high enough, the film intrinsic resistor does not affect this voltage.

For the horizontal direction, the same method is used, but by applying supply from left to right. The range depends on the supply voltage and on the loss in the switches that connect to the top and bottom electrodes.

In an ideal world (linear, with no loss through switches), the horizontal position is equal to:

VY_M / VDD or VY_P / VDD.

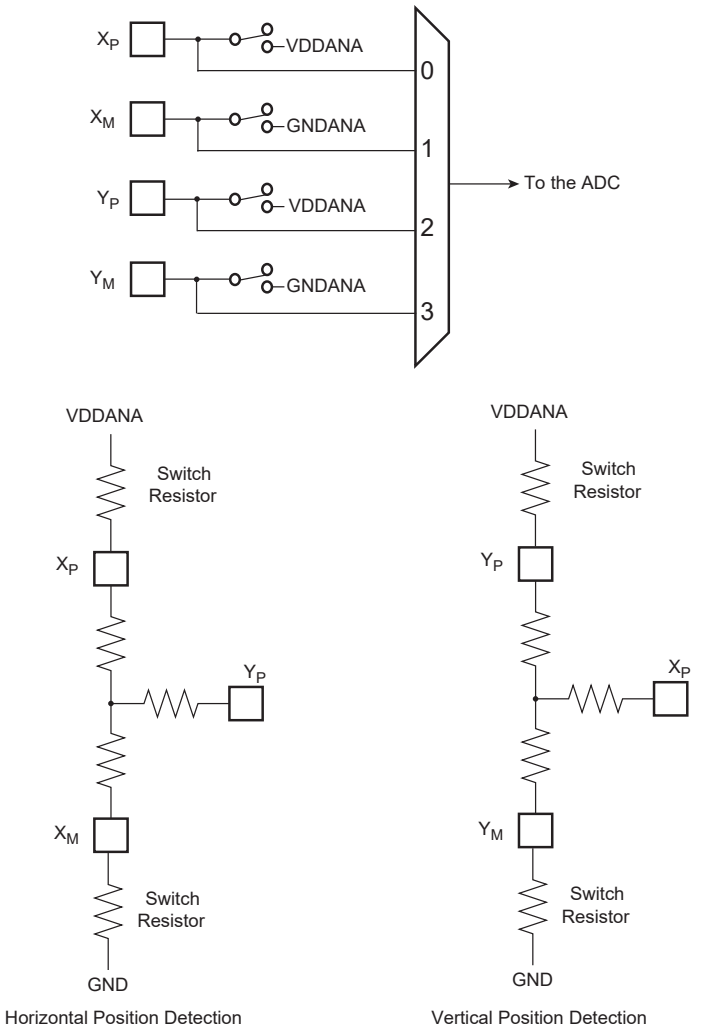
The implementation with on-chip power switches is shown in the figure below. The voltage measurement at the output of the switch compensates for the switches loss.

It is possible to correct for switch loss by performing the operation:

$[VY_P - VX_M] / [VX_P - VX_M]$.

This requires additional measurements, as shown in the figure below.

Figure 72.15. Touchscreen Switches Implementation



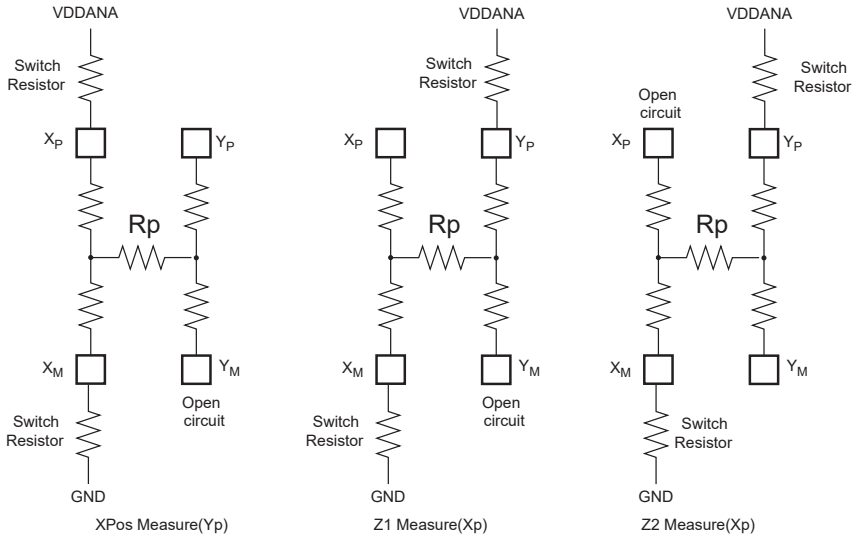
72.6.15.4.4-wire Pressure Measurement Method

The method to measure the pressure (Rp) applied to the touchscreen is based on the known resistance of the X-Panel resistance (Rxp).

Three conversions (Xpos,Z1,Z2) are necessary to determine the value of Rp (Zaxis resistance).

$Rp = Rxp \times (Xpos/1024) \times [(Z2/Z1)-1]$

Figure 72.16. Pressure Measurement



72.6.15.5.5-wire Resistive Touchscreen Principles

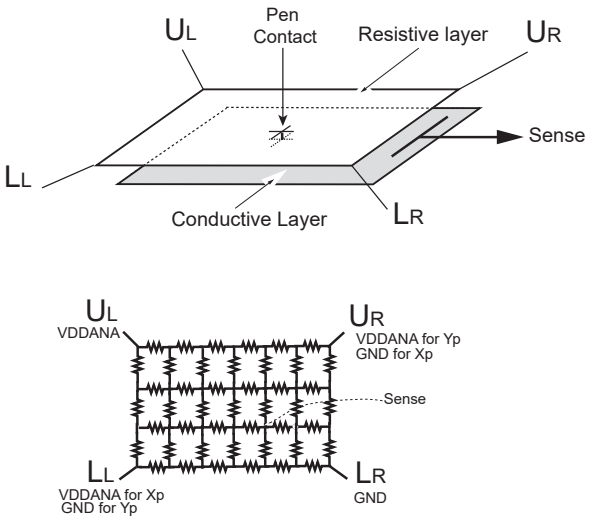
To make a 5-wire touchscreen, a resistive layer with a contact point at each corner and a conductive layer are used.

The 5-wire touchscreen differs from the 4-wire type mainly in that the voltage gradient is applied only to one layer, the resistive layer, while the other layer is the sense layer for both measurements.

The measurement of the X position is obtained by biasing the upper left corner and lower left corner to VDDANA and the upper right corner and lower right to ground.

To measure along the Y axis, bias the upper left corner and upper right corner to VDDANA and bias the lower left corner and lower right corner to ground.

Figure 72.17. 5-Wire Principle



72.6.15.6.5-wire Position Measurement Method

In an application only monitoring clicks, 100 points per second is typically needed. For handwriting or motion detection, the number of measurements to consider is approximately 200 points per second. This must take into account that multiple measurements are included (over sampling, filtering) to compute the correct point.

The 5-wire touchscreen panel works by applying a voltage at the corners of the resistive layer and measuring the vertical or horizontal resistive network with the sense input. The ADC converts the voltage measured at the point the panel is touched.

A measurement of the Y position of the pointing device is made by:

- Connecting Upper left (UL) and upper right (UR) corners to VDDANA
- Connecting Lower left (LL) and lower right (LR) corners to ground.

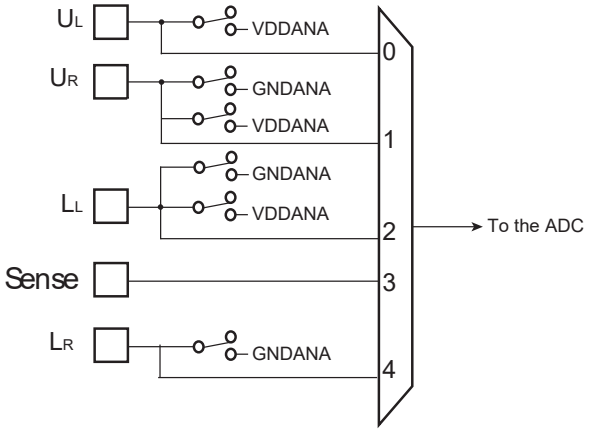
The voltage measured is determined by the voltage divider developed at the point of touch (Y position) and the SENSE input is converted by ADC.

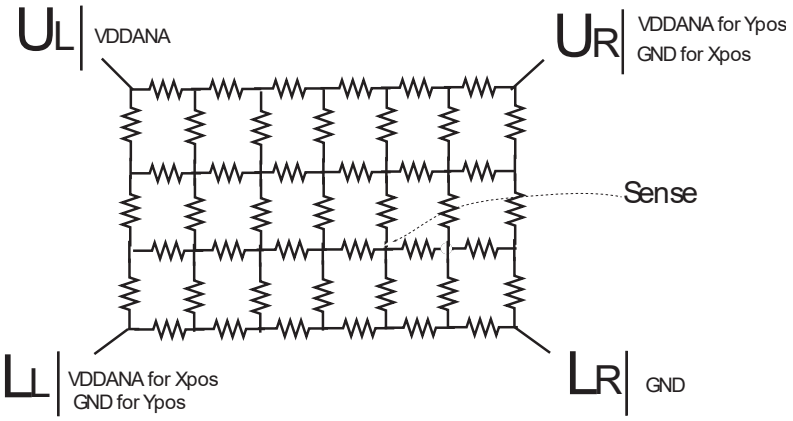
A measurement of the X position of the pointing device is made by:

- Connecting the upper left (UL) and lower left (LL) corners to ground
- Connecting the upper right and lower right corners to VDDANA.

The voltage measured is determined by the voltage divider developed at the point of touch (X position) and the SENSE input is converted by ADC.

Figure 72.18. Touchscreen Switches Implementation





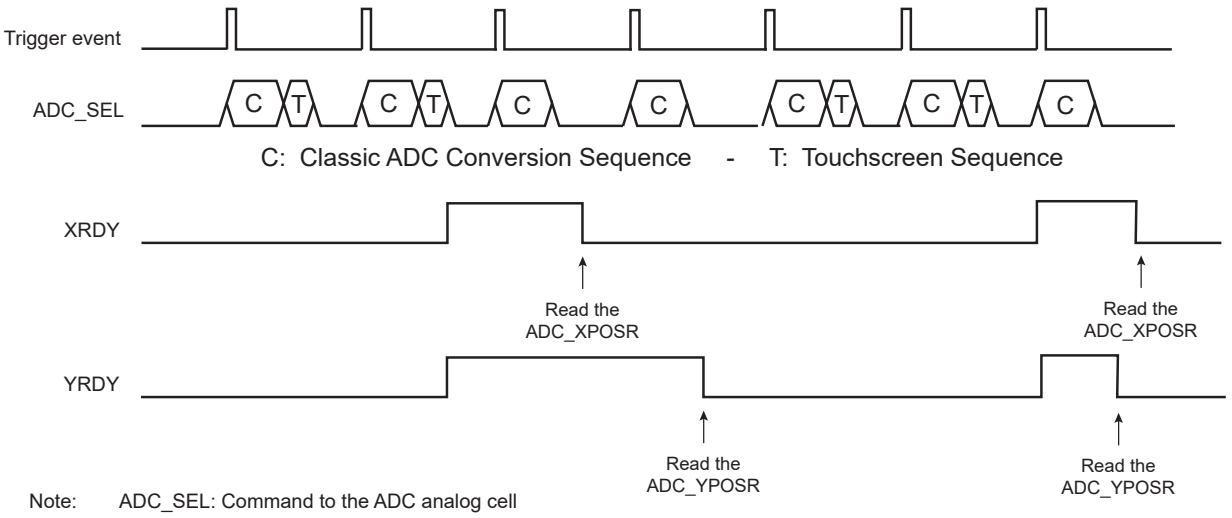
72.6.15.7.Sequence and Noise Filtering

The ADC Controller can manage ADC conversions and touchscreen measurement. On each trigger event the sequence of ADC conversions is performed as described in Sleep Mode and Conversion Sequencer. The touchscreen measure frequency can be specified in number of trigger events by writing the ADC_TSMR.TSFREQ parameter. An internal counter counts triggers up to TSFREQ, and

every time it rolls out, a touchscreen sequence is appended to the classic ADC conversion sequence (see figure below).

Additionally the user can average multiple touchscreen measures by writing the ADC_TSMR.TSAV parameter. This can be 1, 2, 4 or 8 measures performed on consecutive triggers as illustrated in the figure below. Consequently, the ADC_TSMR.TSFREQ parameter must be greater than or equal to the ADC_TSMR.TSAV parameter.

Figure 72.19. Insertion of Touchscreen Sequences (TSFREQ = 2; TSAV = 1)



72.6.15.8.Measured Values, Registers and Flags

As soon as the controller finishes the Touchscreen sequence, XRDY, YRDY and PRDY are set and can generate an interrupt. These flags can be read in the Interrupt Status register (ADC_ISR). They are reset independently by reading in the ADC Touchscreen X Position register (ADC_XPOSR), the ADC Touchscreen Y Position register (ADC_YPOSR) and the ADC Touchscreen Pressure register (ADC_PRESSR).

ADC_XPOSR presents XPOS (VX - VXmin) on its LSB and XSCALE (VXMAX - VXmin) aligned on the 16th bit.

ADC_YPOSR presents YPOS (VY - VYmin) on its LSB and YSCALE (VYMAX - VYmin) aligned on the 16th bit.

To improve the quality of the measure, the user must calculate XPOS/XSCALE and YPOS/YSCALE.

VXMAX, VXmin, VYMAX, and VYmin are measured at the first start-up of the controller. These values can change during use, so it can be necessary to refresh them. Refresh can be done by writing '1' in the ADC_CR.TSCALIB field.

ADC_PRESSR presents Z1 on its LSB and Z2 aligned on the 16th bit. See 4-wire Pressure Measurement Method.

72.6.15.9.Pen Detect Method

When there is no contact, it is not necessary to perform a conversion. However, it is important to detect a contact by keeping the power consumption as low as possible.

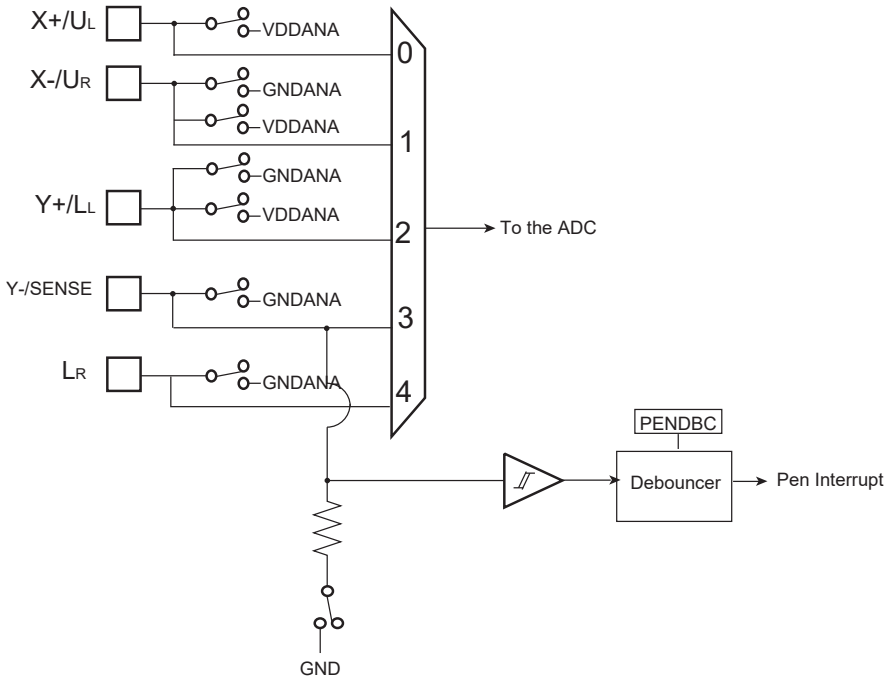
The implementation polarizes one panel by closing the switch on (X_P/U_L) and ties the horizontal panel by an embedded resistor connected to Y_M / Sense. This resistor is enabled by a fifth switch. Since there is no contact, no current is flowing and there is no related power consumption. As soon as a contact occurs, a current is flowing in the Touchscreen and a Schmitt trigger detects the voltage in the resistor.

The Touchscreen Interrupt configuration is entered by programming ADC_TSMR.PENDET. If this bit is written at 1, the controller samples the pen contact state when it is not converting and waiting for a trigger.

To complete the circuit, a programmable debouncer is placed at the output of the Schmitt trigger. This debouncer is programmable up to 2^{15} ADC clock periods. The debouncer length can be selected by programming the ADC_TSMR.PENDBC field.

Due to the analog switch's structure, the debouncer circuitry is only active when no conversion (touchscreen or classic ADC channels) is in progress. Thus, if the time between the end of a conversion sequence and the arrival of the next trigger event is lower than the debouncing time configured on ADC_TSMR.PENDBC, the debouncer will not detect any contact.

Figure 72.20. Touchscreen Pen Detect



The touchscreen pen detect can be used to generate an ADC interrupt to wake up the system. The pen detect generates two types of status, reported in ADC_ISR:

- ADC_ISR.PEN is set as soon as a contact exceeds the debouncing time as defined by ADC_TSMR.PENDBC and remains set until ADC_ISR is read.
- ADC_ISR.NOPEN is set as soon as no current flows for a time over the debouncing time as defined by PENDBC and remains set until ADC_ISR is read.

Both bits are automatically cleared as soon as ADC_ISR is read, and can generate an interrupt by writing ADC_IER.

Moreover, the rising of either one of them clears the other, they cannot be set at the same time.

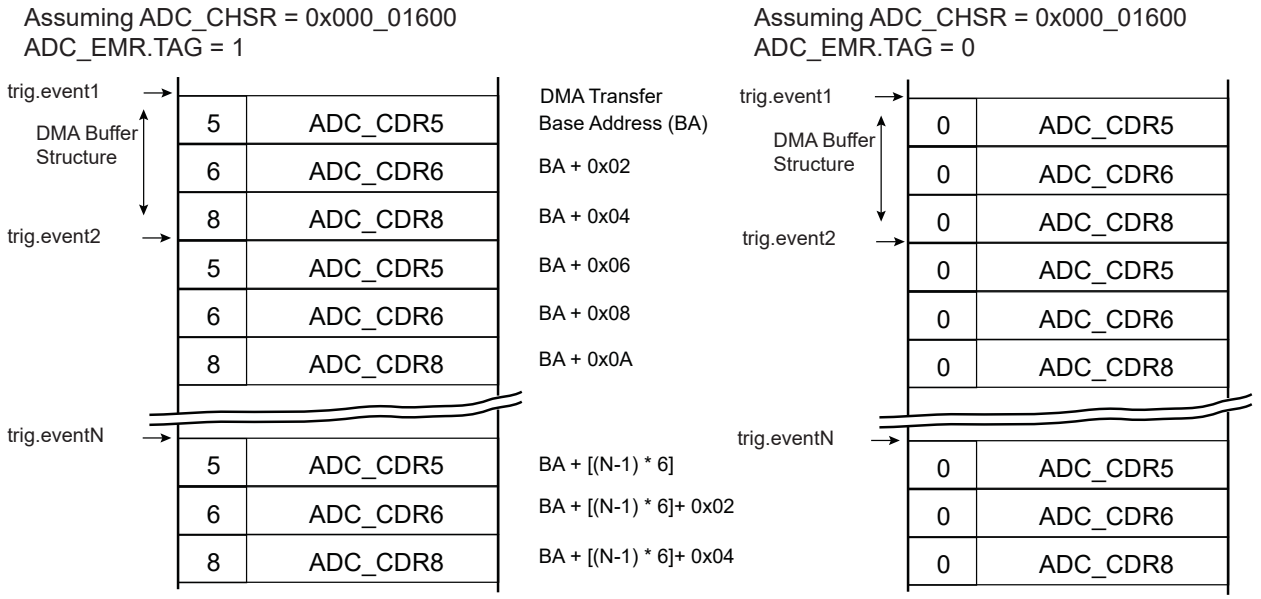
ADC_ISR.PENS shows the current status of the pen contact.

72.6.16. Buffer Structure

The DMA read channel is triggered each time a new data is stored in ADC_LCDR. The same data structure is repeatedly stored in ADC_LCDR each time a trigger event occurs. Depending on user mode of operation (ADC_MR, ADC_CHSR, ADC_SEQR1, ADC_TSMR) the structure differs. Each data read to DMA buffer, carried on a half-word (16-bit), consists of last converted data right-aligned and

when the ADC_EMR.TAG is set, the four most significant bits are carrying the channel number thus allowing an easier postprocessing in the DMA buffer or better checking the DMA buffer integrity.

Figure 72.21. Buffer Structure



As soon as touchscreen conversions are required, the pen detection function can help the postprocessing of the buffer. See Pen Detection Status.

72.6.16.1.Classic ADC Channels Only (Touchscreen Disabled)

When no touchscreen conversion is required (i.e., ADC_TSMR.TSMODE = 0), the data structure within the buffer is defined by ADC_MR, ADC_CHSR, ADC_SEQR1. See 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 will be 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_SEQR1 may modify the data structure of the buffer as well.

72.6.16.2.Touchscreen Channels Only

When only touchscreen conversions are required (i.e., TSMODE ≠ 0 in ADC_TSMR and ADC_CHSR equals 0), the structure of data within the buffer is defined by ADC_TSMR.

When TSMODE = 1 or 3, each trigger event adds two half-words in the buffer (assuming TSAV = 0), first half-word being ADC_XPOSR.XPOS, then ADC_YPOSR.YPOS. If TSAV/TSFREQ ≠ 0, the data structure remains unchanged. Not all trigger events add data to the buffer.

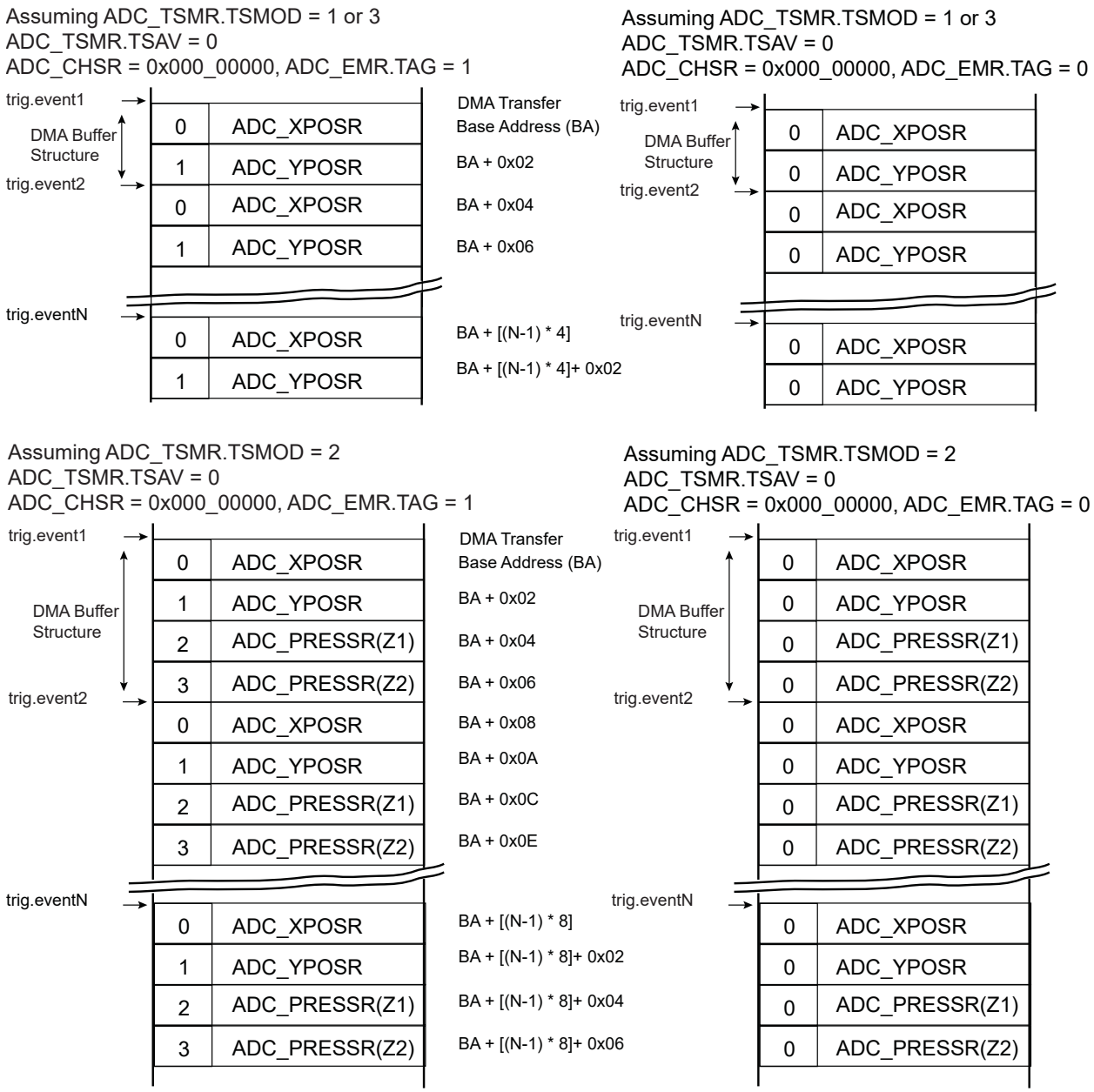
When TSMODE = 2, each trigger event adds four half-words to the buffer (assuming TSAV = 0), first half-word being ADC_XPOSR.XPOS, followed by ADC_YPOSR.YPOS and finally ADC_PRESSR.Z1, followed by ADC_PRESSR.Z2.

When ADC_EMR.TAG is set, the CHNB field (four most significant bits of ADC_LCDR) is cleared when ADC_XPOSR.XPOS is transmitted and set when ADC_YPOSR.YPOS is transmitted, allowing an easier post-processing of the buffer or a better checking of the buffer integrity. In case 4-wire with Pressure mode is selected, the Z1 value is transmitted to the buffer along with tag set to 2 and Z2 is tagged with value 3.

XSCALE and YSCALE (calibration values) are not transmitted to the buffer because they are supposed to be constant and moreover only measured at the very first start-up of the controller or upon user request.

There is no change in buffer structure whatever the value of PENDET.ADC_TSMR, but it is recommended to use the pen detection function for buffer postprocessing (see Pen Detection Status).

Figure 72.22. Buffer Structure When Only Touchscreen Channels are Enabled



72.6.16.3.Interleaved Channels

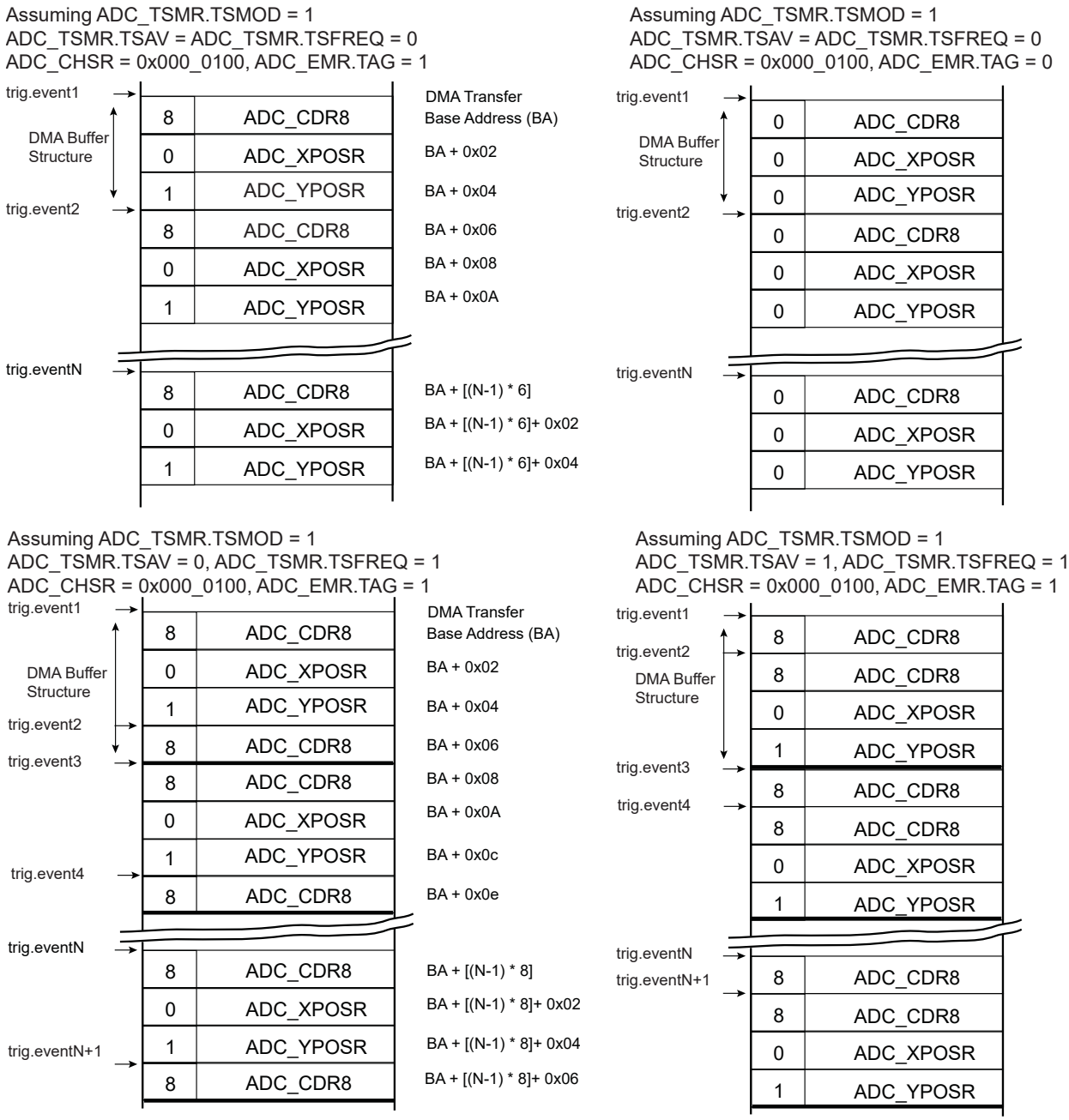
When both classic ADC channels (CH4/CH5 up to CH8 are set in ADC_CHSR) and touchscreen conversions are required (TSMODE ≠ 0 in ADC_TSMR), the structure of the buffer differs according to the ADC_TSMR.TSAV and ADC_TSMR.TSFREQ values.

If TSFREQ ≠ 0, not all events generate touchscreen conversions, therefore the buffer structure is based on 2^{TSFREQ} trigger events. Given a TSFREQ value, the location of touchscreen conversion results depends on TSAV value.

When TSFREQ = 0, TSAV must equal 0.

There is no change in buffer structure whatever the value of ADC_TSMR.PENDET, but it is recommended to use the pen detection function for buffer post-processing (see Pen Detection Status).

Figure 72.23. Buffer Structure When Classic ADC and Touchscreen Channels are Interleaved



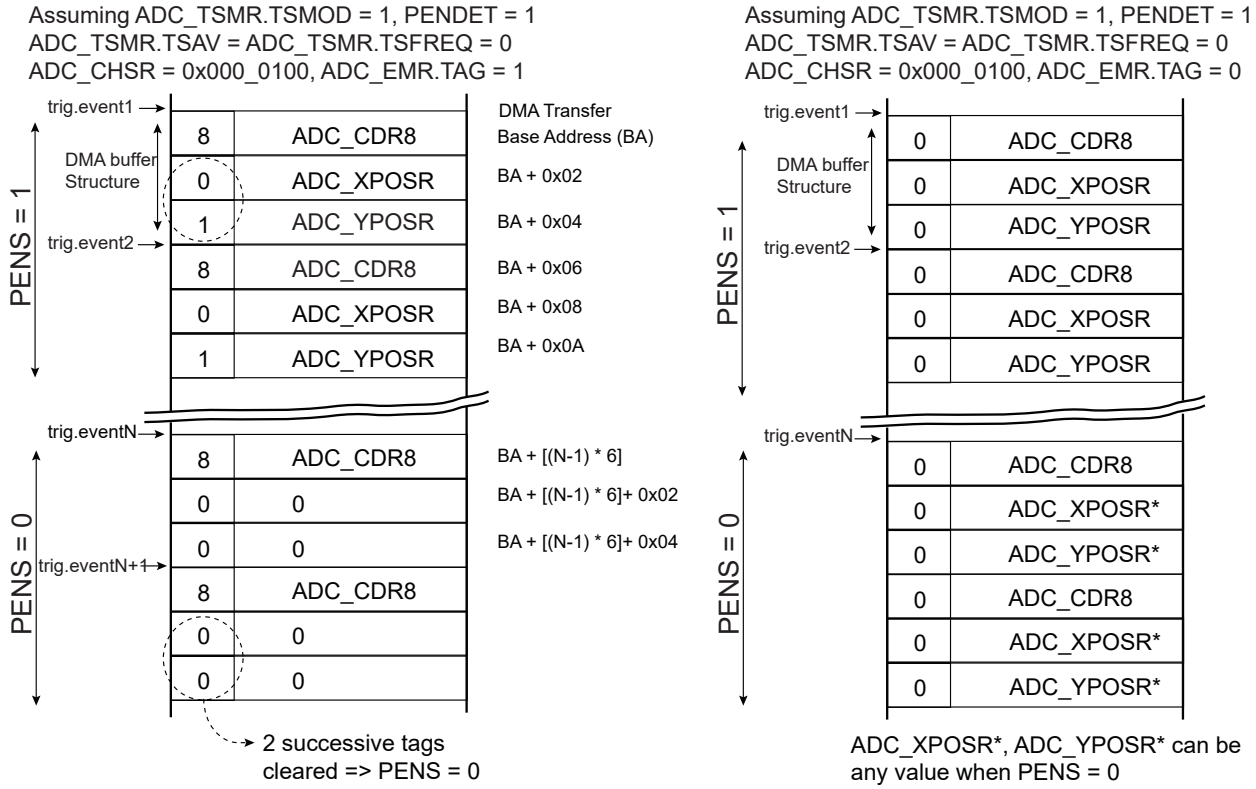
72.6.16.4.Pen Detection Status

If the pen detection measure is enabled (ADC_TSMR.PENDET is set), the XPOS, YPOS, Z1, Z2 values transmitted to the buffer through ADC_LCDR are cleared (including the CHNB field), if the ADC_ISR.PENS flag is 0. When the ADC_ISR.PENS flag is set, XPOS, YPOS, Z1, Z2 are normally transmitted.

Therefore, using pen detection together with tag function eases the post-processing of the buffer, especially to determine which touchscreen converted values correspond to a period of time when the pen was in contact with the screen.

When the pen detection is disabled or the tag function is disabled, XPOS, YPOS, Z1, Z2 are normally transmitted without tag and no relationship can be found with pen status, thus post-processing may not be easy.

Figure 72.24. Buffer Structure With and Without Pen Detection Enabled



72.6.17. 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 ADC_EMR, ADC_CWR, ADC_LCMR and ADC_LCCWR and converted values.

Two types of comparison can trigger a comparison event (fault output pulse):

- The first comparison type is based on ADC_LCCWR settings, i.e., on all converted channels except the last one;
- The second comparison type is linked to the last channel.

As an example, overcurrent and temperature exceeding limits can trigger a fault to 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.

72.6.18. 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_WPMR.

If a write access to the protected registers is detected, the WPVS flag in the 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 1 Register
- ADC Channel Enable Register
- ADC Channel Disable Register
- ADC Last Channel Trigger Mode Register
- ADC Last Channel Compare Window Register
- ADC Extended Mode Register
- ADC Compare Window Register
- ADC Channel Configuration Register
- ADC Analog Control Register
- ADC Pseudo-Differential Register
- ADC_Touchscreen Mode Register
- ADC Trigger Register
- ADC Correction Values Register
- ADC Channel Error Correction Register
- ADC Touchscreen Correction Values 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.WPCTEN is set:

- ADC Control Register

72.7. 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		TSCALIB	START	SWRST
0x04	ADC_MR	31:24	USEQ	MAXSPEED	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 ... 0x0F	Reserved									
0x10	ADC_CHER	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x14	ADC_CHDR	31:24								
		23:16								
		15:8								
		7:0	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
0x18	ADC_CHSR	31:24								
		23:16								
		15:8								
		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]							
0x24	ADC_IER	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x28	ADC_IDR	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x2C	ADC_IMR	31:24		NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x30	ADC_ISR	31:24	PENS	NOPEN	PEN			COMPE	GOVRE	DRDY
		23:16		PRDY	YRDY	XRDY	LCCHG			
		15:8								
		7:0	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
0x34	ADC_LCTMR	31:24								
		23:16								
		15:8								
		7:0			CMPMOD[1:0]					DUALTRIG
0x38	ADC_LCCWR	31:24					HIGHTHRES[11:8]			
		23:16	HIGHTHRES[7:0]							
		15:8					LOWTHRES[11:8]			
		7:0	LOWTHRES[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	ADC_OVER	31:24								
		23:16								
		15:8								
		7:0	OVRE7	OVRE6	OVRE5	OVRE4	OVRE3	OVRE2	OVRE1	OVRE0
0x40	ADC_EMR	31:24			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
		23:16		TRACKX4	SRCCLK	ASTE		OSR[2:0]		
		15:8			CMPFILTER[1:0]				CMPALL	
		7:0	CMPSEL[3:0]					CMPTYPE	CMPMODE[1:0]	
0x44	ADC_CWR	31:24	HIGHTHRES[15:8]							
		23:16	HIGHTHRES[7:0]							
		15:8	LOWTHRES[15:8]							
		7:0	LOWTHRES[7:0]							
0x48 ... 0x4B	Reserved									
0x4C	ADC_CCR	31:24								
		23:16	DIFF7	DIFF6	DIFF5	DIFF4	DIFF3	DIFF2	DIFF1	DIFF0
		15:8								
		7:0								
0x50	ADC_CDR0	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x54	ADC_CDR1	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x58	ADC_CDR2	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x5C	ADC_CDR3	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x60	ADC_CDR4	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x64	ADC_CDR5	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x68	ADC_CDR6	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x6C	ADC_CDR7	31:24								
		23:16								
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x70 ... 0x93	Reserved									
0x94	ADC_ACR	31:24								
		23:16								
		15:8							IBCTL[1:0]	
		7:0							PENDETSENS[1:0]	

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x98 ... 0x9F	Reserved									
0xA0	ADC_PDR	31:24								
		23:16								
		15:8								
		7:0	PDIFF7	PDIFF6	PDIFF5	PDIFF4	PDIFF3	PDIFF2	PDIFF1	PDIFF0
0xA4 ... 0xAF	Reserved									
0xB0	ADC_TSMR	31:24	PENDBC[3:0]							PENDET
		23:16		NOTSDMA			TSSCTIM[3:0]			
		15:8					TSFREQ[3:0]			
		7:0			TSAV[1:0]				TSMODE[1:0]	
0xB4	ADC_XPOSR	31:24					XSCALE[11:8]			
		23:16	XSCALE[7:0]							
		15:8					XPOS[11:8]			
		7:0	XPOS[7:0]							
0xB8	ADC_YPOSR	31:24					YSCALE[11:8]			
		23:16	YSCALE[7:0]							
		15:8					YPOS[11:8]			
		7:0	YPOS[7:0]							
0xBC	ADC_PRESSR	31:24					Z2[11:8]			
		23:16	Z2[7:0]							
		15:8					Z1[11:8]			
		7:0	Z1[7:0]							
0xC0	ADC_TRGR	31:24	TRGPER[15:8]							
		23:16	TRGPER[7:0]							
		15:8								
		7:0						TRGMOD[2:0]		
0xC4 ... 0xD3	Reserved									
0xD4	ADC_CVR	31:24	GAINCORR[15:8]							
		23:16	GAINCORR[7:0]							
		15:8	OFFSETCORR[15:8]							
		7:0	OFFSETCORR[7:0]							
0xD8	ADC_CECR	31:24								
		23:16								
		15:8								
		7:0	ECORR7	ECORR6	ECORR5	ECORR4	ECORR3	ECORR2	ECORR1	ECORR0
0xDC	ADC_TSCVR	31:24	TSGAINCORR[15:8]							
		23:16	TSGAINCORR[7:0]							
		15:8	TSOFFSETCORR[15:8]							
		7:0	TSOFFSETCORR[7:0]							
0xE0 ... 0xE3	Reserved									
0xE4	ADC_WPMR	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0						WPCTEN	WPITEN	WPEN
0xE8	ADC_WPSR	31:24								
		23:16	WPVSRC[15:8]							
		15:8	WPVSRC[7:0]							
		7:0								WPVS

72.7.1. ADC Control Register

Name: ADC_CR
Offset: 0x00
Reset: –
Property: Write-only

This register can only be written if the WPCTEN 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		TSCALIB	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 2 – TSCALIB Touchscreen Calibration

If conversion is in progress, the calibration sequence starts at the beginning of a new conversion sequence. If no conversion is in progress, the calibration sequence starts at the second conversion sequence located after the TSCALIB command (Sleep mode, waiting for a trigger event).
TSCALIB measurement sequence does not affect the Last Converted Data register (ADC_LCDR).

Value	Description
0	No effect.
1	Programs screen calibration (VDD/GND measurement)

Bit 1 – START Start Conversion

Value	Description
0	No effect.
1	Triggers a single sequence of analog-to-digital conversions if ADC_TRGR.TRGMOD=0.

Bit 0 – SWRST Software Reset

Value	Description
0	No effect.

Value	Description
1	Resets the ADC.

72.7.2. ADC Mode Register

Name: ADC_MR
Offset: 0x04
Reset: 0x20000000
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	MAXSPEED	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	1	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 can be used to convert the same channel several times.

Bit 30 – MAXSPEED Maximum Sampling Rate Enable in Freerun Mode
This bit should always be set to 1.

Bits 29:28 – TRANSFER[1:0] Transfer Time
The TRANSFER field must be set to 2 to guarantee the optimal transfer time.

Bits 27:24 – TRACKTIM[3:0] Tracking Time

ADC_EMR.TRACK4X	TRACKTIM		
	0 to 3	4 to 14	15
0	$6 \times t_{ADCCLK}$		$7 \times t_{ADCCLK}$
1	$6 \times t_{ADCCLK}$	$([4 \times (TRACKTIM + 1)] - 10) \times t_{ADCCLK}$	

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. .

Bits 19:16 – STARTUP[3:0] Start-Up Time

Value	Name	Description
0	SUT0	0 periods 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 Wake-Up

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
2	ADC_TRIG2	TIOA1
3	ADC_TRIG3	TIOA2
4	ADC_TRIG4	RTCOUT1
5	ADC_TRIG5	Reserved
6	ADC_TRIG6	Reserved
7	ADC_TRIG7	Reserved

72.7.3. ADC Channel Sequence 1 Register

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 – USCHx User Sequence Number x
This register can be used only if the ADC_MR.USEQ field is set to '1'.
Any USCHx field is processed only if the ADC_CHSR.CHx-1 bit 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

72.7.4. ADC Channel Enable Register

Name: ADC_CHER
Offset: 0x10
Reset: –
Property: Write-only

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
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Enable
If ADC_MR.USEQ = 1, CHx corresponds to the enable of sequence number x+1 described in ADC_SEQR1 (for example, CH0 enables sequence number USCH1).

Value	Description
0	No effect.
1	Enables the corresponding channel.

72.7.5. 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.

Bit	31	30	29	28	27	26	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
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Disable

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 EOCx and GOVRE flags in ADC_ISR and OVREx flags in ADC_OVER are unpredictable

Value	Description
0	No effect.
1	Disables the corresponding channel.

72.7.6. ADC Channel Status Register

Name: ADC_CHSR
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
	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7 – CHx Channel x Status

Value	Description
0	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx field) is disabled.
1	The corresponding channel (or part of sequence, see ADC_SEQyR.USCHx field) is enabled. 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.

72.7.7. 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 the ADC_EMR.TAG bit is set and the ADC_EMR0.OSR field is not equal to 0. If the ADC_EMR.TAG bit 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 halfword (16-bit) instead of 1 converted data in a 32-bit word, thus dividing by 2 the size of the memory buffer.

72.7.8. 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
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		W	W			W	W	W
Reset		–	–			–	–	–

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
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
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 30 – NOPEN No Pen Contact Interrupt Enable

Bit 29 – PEN Pen Contact Interrupt Enable

Bit 26 – COMPE Comparison Event Interrupt Enable

Bit 25 – GOVRE General Overrun Error Interrupt Enable

Bit 24 – DRDY Data Ready Interrupt Enable

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Enable

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Enable

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Enable

Bit 19 – LCCHG Last Channel Change Interrupt Enable

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Enable x

72.7.9. 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
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		W	W			W	W	W
Reset		–	–			–	–	–

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
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
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

Bit 30 – NOPEN No Pen Contact Interrupt Disable

Bit 29 – PEN Pen Contact Interrupt Disable

Bit 26 – COMPE Comparison Event Interrupt Disable

Bit 25 – GOVRE General Overrun Error Interrupt Disable

Bit 24 – DRDY Data Ready Interrupt Disable

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Disable

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Disable

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Disable

Bit 19 – LCCHG Last Channel Change Interrupt Disable

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Disable x

72.7.10. 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
		NOPEN	PEN			COMPE	GOVRE	DRDY
Access		R	R			R	R	R
Reset		0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
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
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 30 – NOPEN No Pen Contact Interrupt Mask

Bit 29 – PEN Pen Contact Interrupt Mask

Bit 26 – COMPE Comparison Event Interrupt Mask

Bit 25 – GOVRE General Overrun Error Interrupt Mask

Bit 24 – DRDY Data Ready Interrupt Mask

Bit 22 – PRDY Touchscreen Measure Pressure Ready Interrupt Mask

Bit 21 – YRDY Touchscreen Measure YPOS Ready Interrupt Mask

Bit 20 – XRDY Touchscreen Measure XPOS Ready Interrupt Mask

Bit 19 – LCCHG Last Channel Change Interrupt Disable

Bits 0, 1, 2, 3, 4, 5, 6, 7 – EOCx End of Conversion Interrupt Mask x

72.7.11. ADC Interrupt Status Register

Name: ADC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	PENS	NOPEN	PEN			COMPE	GOVRE	DRDY
Access	R	R	R			R	R	R
Reset	0	0	0			0	0	0

Bit	23	22	21	20	19	18	17	16
		PRDY	YRDY	XRDY	LCCHG			
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
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – PENS Pen Detect Status
PENS is not a source of interruption.

Value	Description
0	The pen does not press the screen.
1	The pen presses the screen.

Bit 30 – NOPEN No Pen Contact (cleared on read)

Value	Description
0	No loss of pen contact since the last read of ADC_ISR.
1	At least one loss of pen contact since the last read of ADC_ISR.

Bit 29 – PEN Pen contact (cleared on read)

Value	Description
0	No pen contact since the last read of ADC_ISR.
1	At least one pen contact since the last read of ADC_ISR.

Bit 26 – COMPE Comparison Event (cleared on read)

Value	Description
0	No comparison event 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 22 – PRDY Touchscreen Pressure Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_PRESSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 21 – YRDY Touchscreen YPOS Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_YPOSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 20 – XRDY Touchscreen XPOS Measure Ready (cleared on read)

Value	Description
0	No measure has been performed since the last read of ADC_XPOSR.
1	At least one measure has been performed since the last read of ADC_ISR.

Bit 19 – LCCHG Last Channel Change (cleared on read)

Value	Description
0	There is no comparison match (defined in the Last Channel Compare Window register (ADC_LCCWR) since the last read of ADC_ISR.
1	The converted value reported on ADC_CDR7 has changed since the last read of ADC_ISR, according to what is defined in ADC_LCTMR and ADC_LCCWR.

Bits 0, 1, 2, 3, 4, 5, 6, 7 – 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.

72.7.12. ADC Last Channel Trigger Mode Register

Name: ADC_LCTMR
Offset: 0x34
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
			CMPMOD[1:0]					DUALTRIG
Access			R/W	R/W				R/W
Reset			0	0				0

Bits 5:4 – CMPMOD[1:0] Last Channel Comparison Mode

Value	Name	Description
0	LOW	Generates the ADC_ISR.LCCHG flag when the converted data is lower than the low threshold of the window.
1	HIGH	Generates the ADC_ISR.LCCHG flag when the converted data is higher than the high threshold of the window.
2	IN	Generates the ADC_ISR.LCCHG flag when the converted data is in the comparison window.
3	OUT	Generates the ADC_ISR.LCCHG flag when the converted data is out of the comparison window.

Bit 0 – DUALTRIG Dual Trigger On

Value	Description
0	All channels are triggered by event defined by ADC_MR.TRGSEL.
1	Last channel (higher index) trigger period is defined by RTC_MR.OUT1.

72.7.13. ADC Last Channel Compare Window Register

Name: ADC_LCCWR
Offset: 0x38
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[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
	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[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
	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 27:16 – HIGHTHRES[11:0] High Threshold
High threshold associated to compare settings of ADC_LCTMR.

Bits 11:0 – LOWTHRES[11:0] Low Threshold
Low threshold associated to compare settings of ADC_LCTMR.

72.7.14. ADC Overrun Status Register

Name: ADC_OVER
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
Access								
Reset								

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 0, 1, 2, 3, 4, 5, 6, 7 – OVREx Overrun Error x

Note: An overrun error does not always mean that the unread data has been replaced by a new valid data. See Enhanced Resolution Mode and Digital Averaging Function for details.

Value	Description
0	No overrun error 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.

72.7.15. ADC Extended Mode Register

Name: ADC_EMR
Offset: 0x40
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
			ADCMODE[1:0]			SIGNMODE[1:0]		TAG
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
		TRACKX4	SRCCLK	ASTE		OSR[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
			CMPFILTER[1:0]				CMPALL	
Access			R/W	R/W			R/W	
Reset			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

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 Table 72.6.
2	GAIN_ERROR_HIGH	Gain Error mode to measure the gain error. See Table 72.6.
3	GAIN_ERROR_LOW	Gain Error mode to measure the gain error. See Table 72.6.

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 the bit 15 (for example, 0xF43 for 12-bit resolution will be read as 0xFF43 and 0x467 will be read as 0x0467). See Conversion Results Format.

Value	Name	Description
0	SE_UNSG_DF_SIGN	Single-Ended channels: Unsigned conversions.Pseudo-differential channels and Differential channels: Signed conversions.
1	SE_SIGN_DF_UNSG	Single-Ended channels: Signed conversions.Pseudo-differential channels and Differential channels: Unsigned conversions.
2	ALL_UNSIGNED	All channels: Unsigned conversions.
3	ALL_SIGNED	All channels: Signed conversions.

Bit 24 – TAG Tag of ADC_LCDR

Value	Description
0	Sets ADC_LCDR.CHNB field to zero.

Value	Description
1	Appends the channel number to the conversion result in ADC_LCDR.

Bit 22 – TRACKX4 Tracking Time x4

Value	Description
0	The ADC_MR.TRACKTIM field effect is multiplied by 1.
1	The ADC_MR.TRACKTIM field effect is multiplied by 4.

Bit 21 – SRCCLK External Clock Selection

Value	Name	Description
0	PERIPH_CLK	The peripheral clock is the source for the ADC prescaler.
1	GCLK	GCLK is the source clock for the ADC prescaler, thus the ADC clock can be independent of the core/peripheral clock.

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	3-bit enhanced resolution by averaging. ADC sample rate divided by 64.
4	OSR256	4-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 Windows when using the filtering option (CMPFILTER > 0).

Bit 9 – CMPALL Compare All Channels

Value	Description
0	Only channel indicated in CMPSEL field is compared.
1	All channels are compared.

Bits 7:4 – CMPSEL[3:0] 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 ADC_ISR.COMPE flag.
1	START_CONDITION	Comparison conditions must be met to start the storage of all conversions until the ADC_CR.CMPRST bit 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 ADC_ISR.COMPE flag.

Value	Name	Description
1	HIGH	When the converted data is higher than the high threshold of the window, generates the ADC_ISR.COMPE flag.
2	IN	When the converted data is in the comparison window, generates the ADC_ISR.COMPE flag.
3	OUT	When the converted data is out of the comparison window, generates the ADC_ISR.COMPE flag.

72.7.16. ADC Compare Window Register

Name: ADC_CWR
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
	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 compare settings of ADC_EMR.

Bits 15:0 – LOWTHRES[15:0] Low Threshold
Low threshold associated to compare settings of ADC_EMR.

72.7.17. Channel Configuration Register

Name: ADC_CCR
Offset: 0x4C
Reset: 0
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
	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

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 16, 17, 18, 19, 20, 21, 22, 23 – DIFFx Differential Inputs for Channel x

Value	Description
0	The corresponding channel is set in Single-Ended mode.
1	The corresponding channel is set in Differential mode.

72.7.18. ADC Channel Data Register

Name: ADC_CDRx
Offset: 0x50 + x*0x04 [x=0..7]
Reset: 0
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
	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 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.

72.7.19. ADC Analog Control Register

Name: ADC_ACR
Offset: 0x94
Reset: 0x00000101
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
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
							IBCTL[1:0]	
Access							R/W	R/W
Reset							0	1

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

Bits 9:8 – IBCTL[1:0] ADC Bias Current Control
Adapts performance versus power consumption. Refer to the section “Electrical Characteristics” for further details.

Bits 1:0 – PENDETSENS[1:0] Pen Detection Sensitivity
Modifies the pen detection input pull-up resistor value. Refer to the section “Electrical Characteristics” for further details.

72.7.20. ADC Pseudo-Differential Register

Name: ADC_PDR
Offset: 0xA0
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
	PDIFF7	PDIFF6	PDIFF5	PDIFF4	PDIFF3	PDIFF2	PDIFF1	PDIFF0
Access	R/W	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 – PDIFFx Pseudo-Differential Inputs for Channel x

Value	Description
0	The channel is configured as defined by the ADC_CCR.DIFFx bit.
1	The channel is configured in Pseudo-Differential mode.

72.7.21. ADC Touchscreen Mode Register

Name: ADC_TSMR
Offset: 0xB0
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
	PENDBC[3:0]							PENDET
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
		NOTSDMA			TSSCTIM[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
					TSFREQ[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
			TSAV[1:0]				TSMODE[1:0]	
Access			R/W	R/W			R/W	R/W
Reset			0	0			0	0

Bits 31:28 – PENDBC[3:0] Pen Detect Debouncing Period
Debouncing period = 2^{PENDBC} ADCCLK periods.

Bit 24 – PENDET Pen Contact Detection Enable
When PENDET = 1, XPOS, YPOS, Z1, Z2 values of ADC_XPOSR, ADC_YPOSR, ADC_PRESSR are automatically cleared when ADC_ISR.PENS = 0.

Value	Description
0	Pen contact detection disabled.
1	Pen contact detection enabled.

Bit 22 – NOTSDMA No TouchScreen DMA

Value	Description
0	XPOS, YPOS, Z1, Z2 are transmitted in ADC_LCDR.
1	XPOS, YPOS, Z1, Z2 are never transmitted in ADC_LCDR, therefore the buffer does not contains touchscreen values.

Bits 19:16 – TSSCTIM[3:0] Touchscreen Switches Closure Time
Defines closure time of analog switches necessary to establish the measurement conditions.
The closure time is:
Switch Closure Time = (TSSCTIM × 4) ADCCLK periods.

Bits 11:8 – TSFREQ[3:0] Touchscreen Frequency
Defines the touchscreen frequency compared to the trigger frequency.

TSFREQ must be greater or equal to TSAV.
The touchscreen frequency is:
Touchscreen Frequency = Trigger Frequency / 2^{TSFREQ}

Bits 5:4 – TSAV[1:0] Touchscreen Average

Value	Name	Description
0	NO_FILTER	No filtering. Only one ADC conversion per measure.
1	AVG2CONV	Averages 2 ADC conversions.
2	AVG4CONV	Averages 4 ADC conversions.
3	AVG8CONV	Averages 8 ADC conversions.

Bits 1:0 – TSMODE[1:0] Touchscreen Mode
When TSMOD equals 01 or 10 (i.e., 4-wire mode), channels 0, 1, 2 and 3 must not be used for classic ADC conversions. When TSMOD equals 11 (i.e., 5-wire mode), channels 0, 1, 2, 3, and 4 must not be used.

Value	Name	Description
0	NONE	No touchscreen.
1	4_WIRE_NO_PM	4-wire touchscreen without pressure measurement.
2	4_WIRE	4-wire touchscreen with pressure measurement.
3	5_WIRE	5-wire touchscreen.

72.7.22. ADC Touchscreen X Position Register

Name: ADC_XPOSR
Offset: 0xB4
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					XSCALE[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	XSCALE[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
					XPOS[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

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

Bits 27:16 – XSCALE[11:0] Scale of XPOS
Indicates the max value that XPOS can reach. This value should be close to 2^{12}.

Bits 11:0 – XPOS[11:0] X Position
The position measured is stored here. If XPOS = 0 or XPOS = XSIZE, the pen is on the border. When pen detection is enabled (ADC_TSMR.PENDET set to '1'), XPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

72.7.23. ADC Touchscreen Y Position Register

Name: ADC_YPOSR
Offset: 0xB8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
					YSCALE[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	YSCALE[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
					YPOS[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

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

Bits 27:16 – YSCALE[11:0] Scale of YPOS
Indicates the max value that YPOS can reach. This value should be close to 2^{12}.

Bits 11:0 – YPOS[11:0] Y Position
The position measured is stored here. If YPOS = 0 or YPOS = YSIZE, the pen is on the border. When pen detection is enabled (ADC_TSMR.PENDET set to '1'), YPOS is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

72.7.24. ADC Touchscreen Pressure Register

Name: ADC_PRESSR
Offset: 0xBC
Reset: 0x00000000
Property: Read-only

Note: These values are unavailable if ADC_TSMR.TSMODE is not set to 2.

Bit	31	30	29	28	27	26	25	24
					Z2[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	23	22	21	20	19	18	17	16
	Z2[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
					Z1[11:8]			
Access					R	R	R	R
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	Z1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – Z2[11:0] Data of Z2 Measurement
Data Z2 necessary to calculate pen pressure.
When pen detection is enabled (ADC_TSMR.PENDET set to ‘1’), Z2 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

Bits 11:0 – Z1[11:0] Data of Z1 Measurement
Data Z1 necessary to calculate pen pressure.
When pen detection is enabled (ADC_TSMR.PENDET set to ‘1’), Z1 is tied to 0 while there is no detection of contact on the touchscreen (i.e., when the ADC_ISR.PENS bit is cleared).

72.7.25. ADC Trigger Register

Name: ADC_TRGR
Offset: 0xC0
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[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
	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	15	14	13	12	11	10	9	8
Access								
Reset								

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

Bits 31:16 – TRGPER[15:0] 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_SEQR1, ADC_TSMR.
When TRGMOD is set to pen detect trigger (i.e., 100) and averaging is used (i.e., field TSAV ≠ 0 in ADC_TSMR) only one measure is performed. Thus, XRDY, YRDY, PRDY, DRDY will not rise on pen contact trigger. To achieve measurement, several triggers must be provided either by software or by setting the TRGMOD on continuous trigger (i.e., 110) until flags rise.

Bits 2:0 – TRGMOD[2:0] Trigger Mode

Value	Name	Description
0	NO_TRIGGER	No hardware trigger enabled, only software trigger can start conversions
1	EXT_TRIG_RISE	Rising edge of the selected hardware trigger event, defined in ADC_MR.TRGSEL
2	EXT_TRIG_FALL	Falling edge of the selected hardware trigger event
3	EXT_TRIG_ANY	Any edge of the selected hardware trigger event
4	PEN_TRIG	Pen Detect Trigger (shall be selected only if PENDET is set and TSMODE > 0)
5	PERIOD_TRIG	ADC internal hardware periodic trigger (see field TRGPER)
6	CONTINUOUS	Continuous mode

72.7.26. ADC Correction Values Register

Name: ADC_CVR
Offset: 0xD4
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 15 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).

72.7.27. ADC Channel Error Correction Register

Name: ADC_CECR
Offset: 0xD8
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
	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 0, 1, 2, 3, 4, 5, 6, 7 – 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.

72.7.28. ADC Touchscreen Correction Values Register

Name: ADC_TSCVR
Offset: 0xDC
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
	TSGAINCORR[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
	TSGAINCORR[7:0]							
Access	R/W	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
	TSOFFSETCORR[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
	TSOFFSETCORR[7:0]							
Access	R/W	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 – TSGAINCORR[15:0] Touchscreen Gain Correction
Gain correction to apply on converted data for the touchscreen channels. Only bits 0 to 15 are relevant (other bits are ignored and read as 0).

Bits 15:0 – TSOFFSETCORR[15:0] Touchscreen Offset Correction
Offset correction to apply on converted data for the touchscreen channels. The offset is signed (2's complement), only bits 0 to 11 are relevant (other bits are ignored and read as 0).

72.7.29. ADC Write Protection Mode Register

Name: ADC_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	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 bit. Always reads as 0

Bit 2 – WPCTEN Write Protection Control Enable
See Register Write Protection for the list of registers that can be protected.

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 registers that can be protected.

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 the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).
1	Enables the write protection if WPKEY value corresponds to 0x414443 (“ADC” in ASCII).

72.7.30. ADC Write Protection Status Register

Name: ADC_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 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.

Electrical and Mechanical Characteristics

73. Electrical Characteristics

73.1. Electrical Parameters Usage

Tables in the following sections define the limiting values for several electrical parameters:

- I/O Characteristics
- Digital Peripheral Characteristics
- Analog Peripheral Characteristics
- Power Consumption in Active Mode
- Operation and Power Consumption in Low-Power Modes

- Unless otherwise noted, these values are valid over the junction temperature range T_J= [-40°C; +125°C].
- 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. Additionally, specifications in the section Digital Peripheral Characteristics are simulation data, even though they are not annotated as such.
- These limits may be affected by the board on which the device is mounted. 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 ranges, it is mandatory to keep the power supply variations within the limits shown in the following table during the device operation.

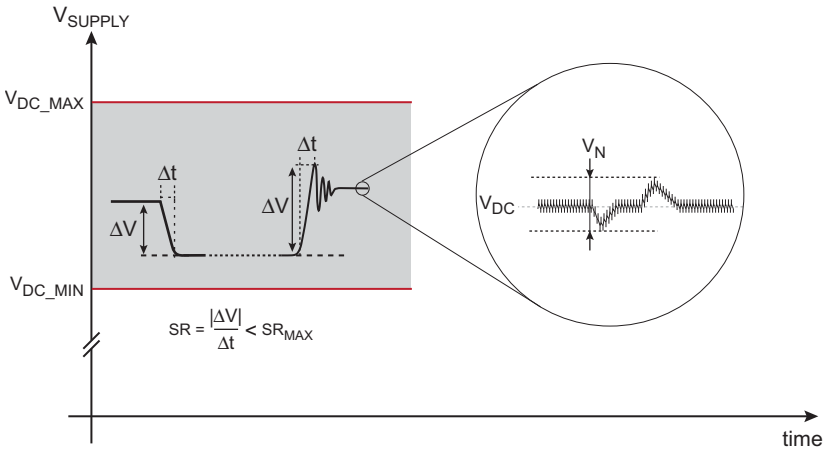
Table 73.1. Maximum Power Supply Variations[1]

Power Input	Parameters	Conditions	Min	Max	Unit
V_N	Peak-to-peak ripple and noise voltage[2]	VDDCORE, VDDBU, VDDIOP0, VDDIOP1, VDDIOP2, VDDQSPI, VDDNF, VDDIOM	–	3	% V_{DC}
		VDDIN33, VDDANA, VDDLVDS[3]	–	1	% V_{DC}
SR	Slewrate of power supply variations[4][5]	$\Delta V < 5\% V_{DC\_MIN}$	–	±50	V/ms
		$\Delta V < 10\% V_{DC\_MIN}$	–	±10	V/ms
		$\Delta V \geq 10\% V_{DC\_MIN}$	–	±1	V/ms

1. VDDMIPI is not mentioned in this table, as it must be connected to the VDDOUT25 regulator output that fulfills the electrical requirements of this power input.

2. V_{DC} is the power supply DC value.
3. The peak-to-peak ripple and noise voltage for VDDLVDS can be relaxed to 3% V_{DC} when this segment is not used with the LVDS PHY.
4. V_{DC_MIN} is the minimum operating voltage of the supply input as described in Table 73.4.
5. ΔV is the variation amplitude. The slewrate specification applies when $\Delta V \geq V_{N.}$ The following examples and figure illustrate the above 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 maintained.
   - 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 maintained.

Figure 73.1. Maximum Power Supply Variation



73.2. Absolute Maximum Ratings

Table 73.2. 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 and GNDANA)	±50 mV	
Voltage on Power Supply Inputs with respect to ground pins:		
VDDCORE	-0.3V to 1.4V	
VDDIOP0, VDDIOP1, VDDIOP2	-0.3V to 4.0V	
VDDIOM	-0.3V to 2.0V	
VDDANA, VDDIN33	-0.3V to 4.0V	
VDDNF, VDDQSPI, VDDBU	-0.3V to 4.0V	
VDDMIPI, VDDLVDS	-0.3V to 4.0V	
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 Pin Description table. Ex: for PA2, this corresponds to VDDIOP0 and GND.

73.3. ESD Ratings

Table 73.3. Electrostatic Discharge (ESD) Ratings

Symbol	Description	Conditions	Value	Unit
Electrostatic Discharge, Human Body Model (HBM)				
ESD_HBM	AEC-Q100-002 Rev-E stress voltage level	All pins	±2	kV
Electrostatic Discharge, Charged-Device Model (CDM)				
ESD_CDM	AEC-Q100-011 Rev-D classification level	All pins	C2a	–
	Corresponding stress voltage level	All pins	±500	V
		Corner pins	±750	V

73.4. Recommended Operating Conditions

Table 73.4. Recommended Operating Conditions on Power Supply Inputs

Power Input	Parameters	Conditions	Min	Max	Unit
VDDIN33	VDDOUT25 regulator input, USB interface I/O lines and Main crystal oscillator power supply[1]	–	3.00	3.60	V
VDDANA	VDDANA I/O lines, A/D Converter and OTP memory power supply[1]	–	3.00	3.60	V

Table 73.4. Recommended Operating Conditions on Power Supply Inputs (continued)

Power Input	Parameters	Conditions	Min	Max	Unit
VDDCORE	Core logic (processor, peripherals, memories, UTMI logic, etc.) power supply (2)	$f_{CPU} \leq 600$ MHz, $f_{MCK} \leq 200$ MHz	1.03	1.21	V
		$f_{CPU} \leq 800$ MHz, $f_{MCK} \leq 266$ MHz	1.12	1.21	V
		Device in ULP0 (f < 25 MHz)(3) or ULP1 mode	1.00	1.21	V
VDDIOM	SDRAM I/O lines power supply	DDR2-SDRAM	1.70	1.90	V
		DDR3-SDRAM	1.425	1.575	V
		DDR3L-SDRAM	1.283	1.450	V
VDDNF	NAND Flash I/O lines power supply(4)	–	1.70	3.60	V
VDDIOP0	VDDIOP0 I/O lines power supply(4)	–	1.70	3.60	V
VDDIOP1	VDDIOP1 I/O lines power supply(4)	–	1.70	3.60	V
VDDIOP2	VDDIOP2 I/O lines power supply(4)	–	1.70	3.60	V
VDDQSPI	VDDQSPI I/O lines power supply(4)	–	1.70	3.60	V
VDDLVDS	LVDS PHY and VDDLVDS I/O lines power supply(4)(5)	–	1.70	3.60	V
VDDMIPI	MIPI PHY and I/O lines power supply(6)	–	2.25	2.75	V
VDDBU	Backup domain power supply	–	1.6	3.60	V
t_{R_VDD}	Power supply slope at power-up	–	0.2	20	mV/µs
t_{F_VDD}	Power supply slope at power-down	–	–20	–1(7)	mV/µs

Notes:

1. VDDANA and VDDIN33 are powered from one single source: V(VDDANA,VDDIN33) ≤ ±50mV.
2. For device lifetime estimation as a function of VDDCORE and temperature, refer to the application note “SAM9X7 Series Product Lifetime Estimation” (AN5531), available on www.microchip.com.
3. In ULP0 mode, all PLLs are off. The maximum clock frequency in the system is 25 MHz: $f_{MAINCK} < 25$ MHz, $f_{CPU} < 25$ MHz, $f_{MCK} < 25$ MHz.
4. Supply range restrictions apply when using the digital peripheral timing characteristics. See I/O Characteristics.
5. When the LVDS PHY is used, VDDLVDS must be connected to VDDOUT25.
6. VDDMIPI must be connected to VDDOUT25.
7. For VDDBU, this value is 0 mV/µs.

Table 73.5. Recommended Operating Conditions on Input Pins(1)

Symbol	Parameters	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 input (4)(5)	–	–	±0.2	mA
I_{TOT_INJ}	Total current injection per power rail or 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 Pin Description table. Ex: for PA2, 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 analog inputs (PA[31:24]), 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 (PA[31:24]) can 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 Pin Description table.

Table 73.6. Recommended Operating Conditions on Internal Clocks

Symbol	Parameters	Conditions	Min	Max	Unit
f_{CPU_CLK}	Processor clock (CPU_CLK) frequency	VDDCORE ≥ 1.12V	–	800	MHz
		VDDCORE ≥ 1.03V	–	600	MHz
f_{MCK}	Main system bus clock (MCK) frequency	VDDCORE ≥ 1.12V	–	266	MHz
		VDDCORE ≥ 1.03V	–	200	MHz

Table 73.7. Recommended Operating Conditions on SDRAM Interface

Symbol	Parameters	Conditions		Min	Max	Unit
f_{SDRAM_CLK}	SDRAM clock frequency	VDDCORE ≥ 1.12V	DDR2-SDRAM	125	266	MHz
			DDR3(L)-SDRAM (DLL ON)[1]	266	–	MHz
			DDR3(L)-SDRAM (DLL OFF)[2]	–	200	MHz
		VDDCORE ≥ 1.03V	DDR2-SDRAM	125	200	MHz
			DDR3(L)-SDRAM (DLL OFF)[2]	–	200	MHz

Notes:

1. According to the JEDEC specification, DDR3(L) “DLL On mode” is supported for clock frequencies of 300 MHz and above. Most memory suppliers accept operations down to 266 MHz. Contact the memory supplier for further details.
2. According to the JEDEC specification, DDR3(L) “DLL Off mode” is supported for clock frequencies up to 125 MHz. Most memory suppliers accept operations up to 200 MHz. Contact the memory supplier for further details.

Table 73.8. Recommended Thermal Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
T_A	Ambient temperature range	SAM9X7x-I devices	-40	85	°C
		SAM9X7x-V devices	-40	105	°C
T_J	Junction temperature range	–	-40	125	°C

Note: For lifetime estimation as a function of VDDCORE and temperature, refer to the application note "SAM9X7 Series Product Lifetime Estimation" (AN5531), available on www.microchip.com.

Table 73.9. Package Characteristics – SAM9X7x-I Devices

Symbol	Parameter	Conditions	Min	Max	Unit
R_{JA}	Junction-to-ambient thermal resistance	–		38	°C/W
R_{JC}	Junction-to-case thermal resistance	–		16	°C/W
R_{JB}	Junction-to-board thermal resistance	–		31	°C/W
$\Psi_{J\text{-}top}$	Junction-to-package top characterization parameter	–		0.57	°C/W

Note: The package characteristics in the above table are provided according to the JEDEC JESD51-2 standard with the 2s2p board and 0 m/s air flow. These values are not directly applicable to the final application. As per JEDEC standards, these parameters represent the device mounted on a specific PCB under controlled conditions. In real-world applications, the PCB design and construction, airflow, and other factors can significantly impact thermal characteristics.

Table 73.10. Package Characteristics – SAM9X7x-V Devices – BGA240

Symbol	Parameter	Conditions	Min	Max	Unit
R_{JA}	Junction-to-ambient thermal resistance	–		33	°C/W
R_{JC}	Junction-to-case thermal resistance	–		12	°C/W
R_{JB}	Junction-to-board thermal resistance	–		26	°C/W
$\Psi_{J\text{-}top}$	Junction-to-package top characterization parameter	–		0.45	°C/W

Note: The package characteristics in the above table are provided according to the JEDEC JESD51-2 standard with the 2s2p board and 0 m/s air flow. These values are not directly applicable to the final application. As per JEDEC standards, these parameters represent the device mounted on a specific PCB under controlled conditions. In real-world applications, the PCB design and construction, airflow, and other factors can significantly impact thermal characteristics.

Table 73.11. Package Characteristics – SAM9X7x-V Devices – BGA256

Symbol	Parameter	Conditions	Min	Max	Unit
R_{JA}	Junction-to-ambient thermal resistance	–		34	°C/W
R_{JC}	Junction-to-case thermal resistance	–		17	°C/W
R_{JB}	Junction-to-board thermal resistance	–		25	°C/W
$\Psi_{J\text{-}top}$	Junction-to-package top characterization parameter	–		0.4	°C/W

Note: The package characteristics in the above table are provided according to the JEDEC JESD51-2 standard with the 2s2p board and 0 m/s air flow. These values are not directly applicable to the final application. As per JEDEC standards, these parameters represent the device mounted on a specific PCB under controlled conditions. In real-world applications, the PCB design and construction, airflow, and other factors can significantly impact thermal characteristics.

73.5. Recommended Power Supply Sequencing

In the following sections, various recommended power sequences are described. Operating the device outside this scope may lead to unpredictable behavior.

73.5.1. Power-Up and Power-Down

At power-up, from a power supply sequencing perspective, the SAM9X7 Series power supply inputs are categorized into three independent groups:

- VDDCORE
- VDDIN33
- Periphery group containing all other power supply inputs except VDDBU

The figure below shows the recommended power-up sequence.

Notes:

- VDDBU
  - When supplied from a pre-charged storage element (battery or supercapacitor), VDDBU is an always-on supply input and is therefore not part of the power supply sequencing.
  - When no storage element is used on VDDBU in the application, VDDBU must be tied to VDDIN33.
  - When a supercapacitor is used in the application to power VDDBU during Backup mode, this element must be isolated from VDDBU during its (slow) charge, so that VDDBU closely follows VDDIN33. In table Power-up Timing Requirements, the parameter t_0 limits the delay to establish VDDBU after VDDIN33.
- VDDOUT25 is the output of the internal 2.5V VDDOUT25 regulator, therefore, there is no power supply requirement on this pin. VDDOUT25 is automatically started at VDDIN33 rise when VDDIN33 is above its Power-On-Reset threshold.
- No board-level requirement on VDDMIPI as it is connected to VDDOUT25.
- VDDLVDS requirements do not apply when VDDLVDS is connected to VDDOUT25.

NRST must be asserted low during the whole power-up sequence.

Figure 73.2. Recommended Power Sequence at Power-Up

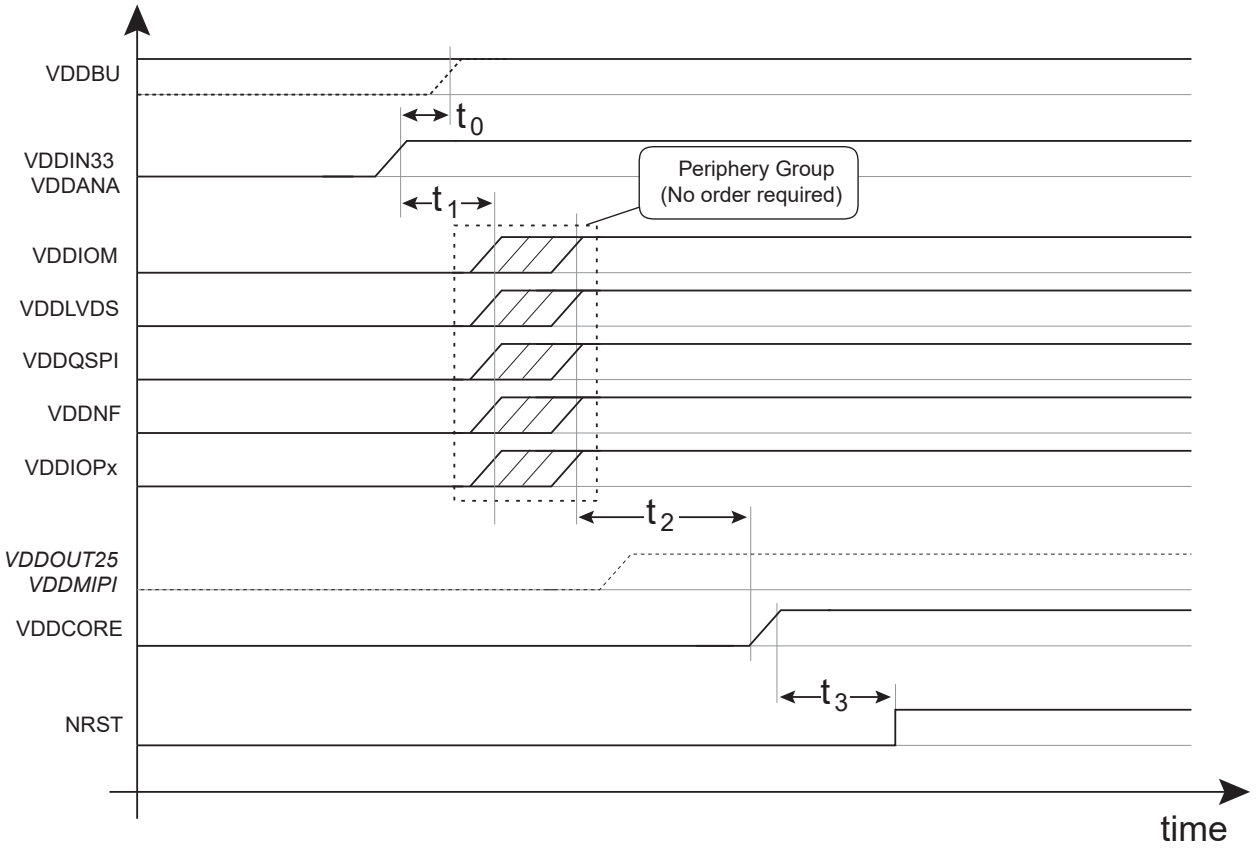


Table 73.12. Power-Up Timing Requirements[1]

Symbol	Parameters	Conditions	Min	Max	Unit
t_0	VDDBU delay	Delay from established VDDIN33 to established VDDBU	–	0.2	ms

Table 73.12. Power-Up Timing Requirements[1] (continued)

Symbol	Parameters	Conditions	Min	Max	Unit
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 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 to 90% of its final value.

The following figure shows the SAM9X7 Series 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. VDDBU 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 73.3. Recommended Power Sequence at Power-Down

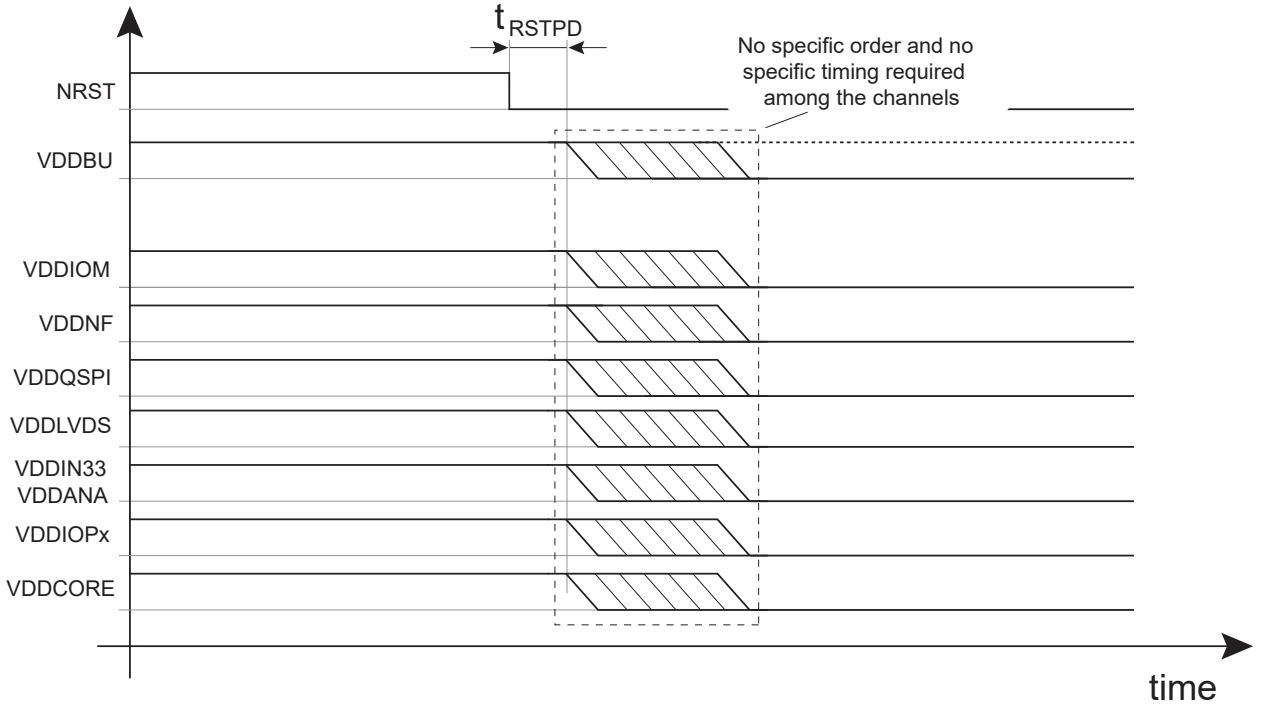


Table 73.13. Power-Down Timing Requirements

Symbol	Parameters	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

73.5.2. Backup Mode Entry and Wake-Up

The following figure shows the recommended power-down sequence to place the SAM9X7 Series device in Backup mode. The SHDN signal, output of Shutdown Controller (SHDWC), signals the shutdown request to the external power supply. This output is supplied by VDDBU that is present in

Backup mode. In a similar way to the power-down sequence, the NRST signal must be asserted prior to turning the SAM9X7 Series power supplies off.

Figure 73.4. Recommended Power Sequence at Backup Mode Entry

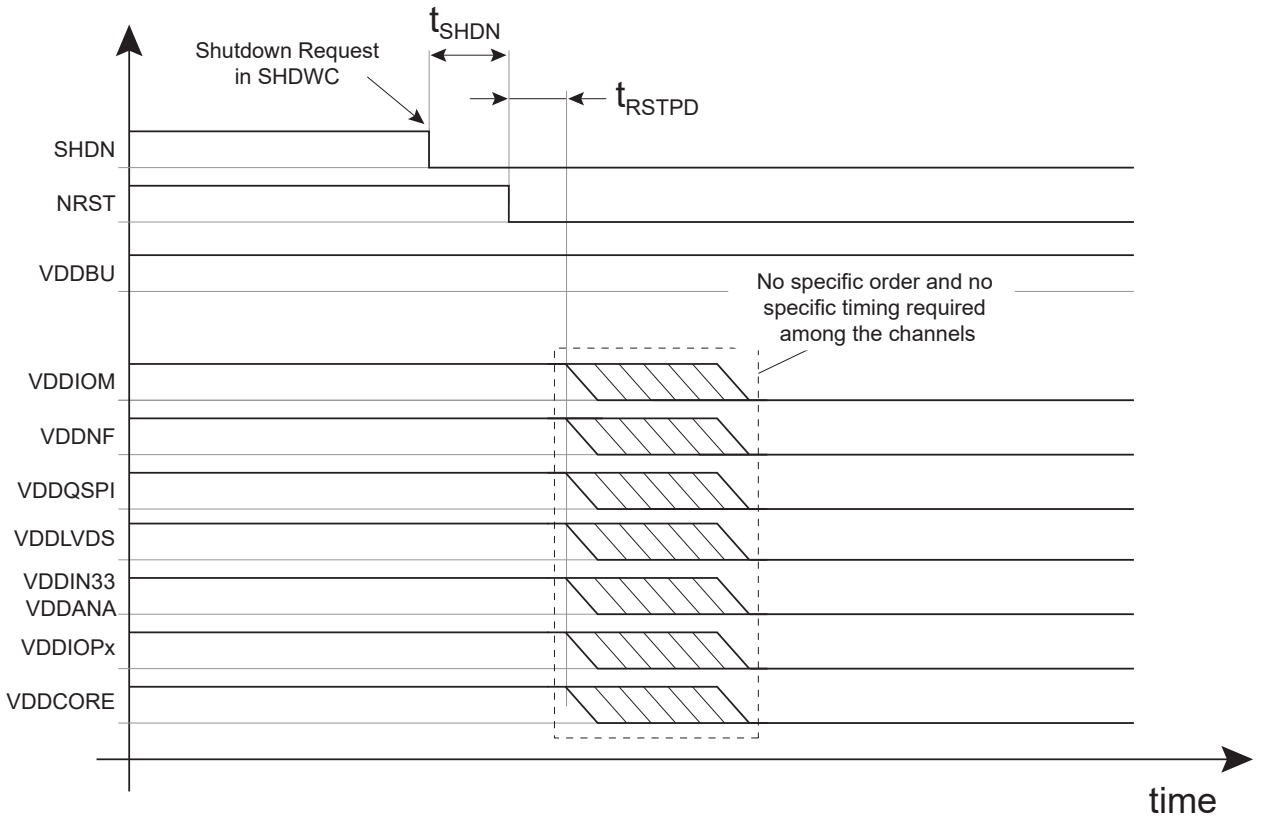


Table 73.14. Backup Mode Entry Timing Requirements

Symbol	Parameters	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 SAM9X7 series device from Backup mode. Upon a wake-up event (WKUP pin, RTC alarm, etc.), the Shutdown Controller automatically toggles its SHDN output back to VDDBU to request the external power supply to restart. This power-up sequence is the same as the one shown in Figure 73.2. In particular, the supply groups definition is the same.

Figure 73.5. Recommended Power Sequence at Wake-Up from Backup Mode

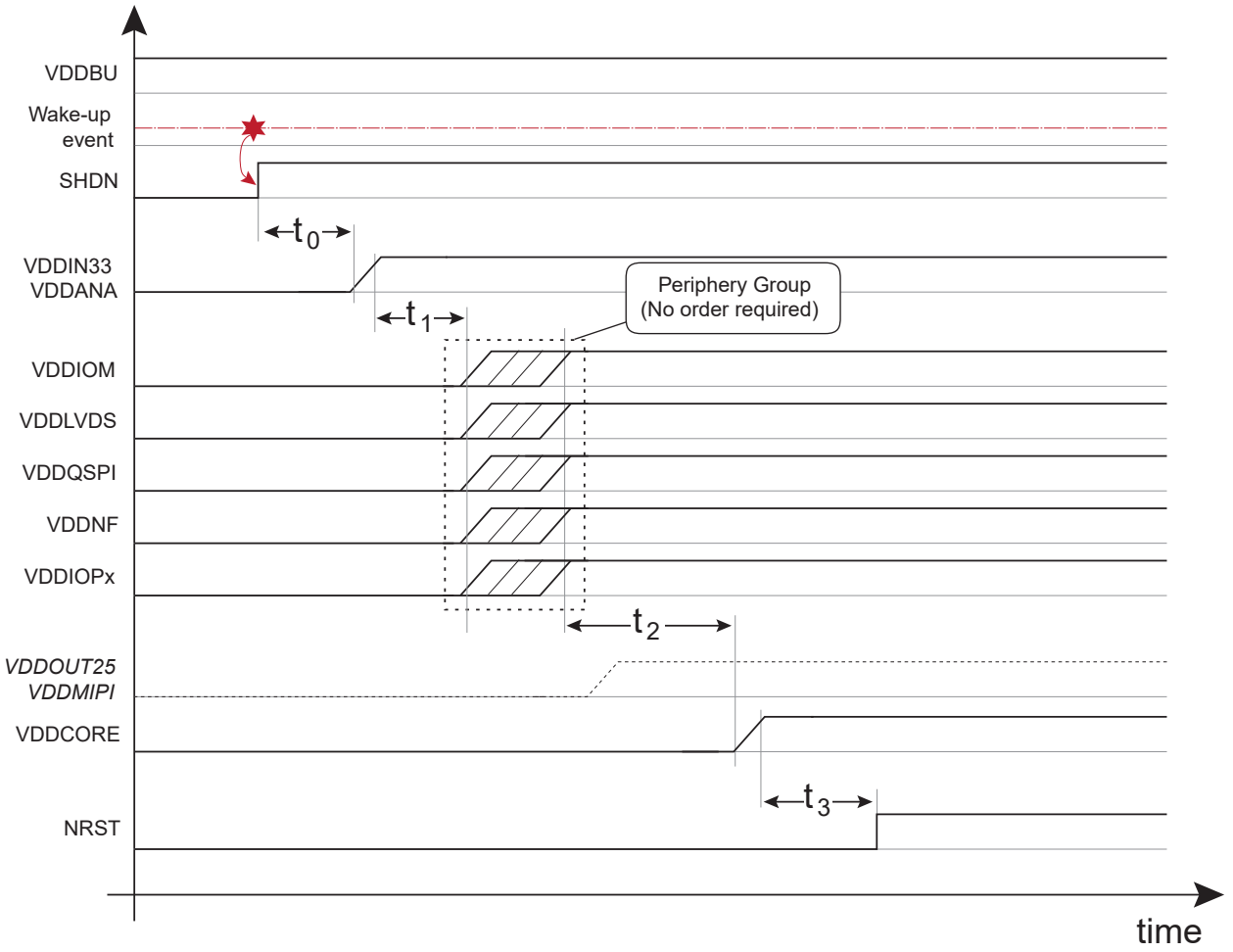


Table 73.15. Wake-Up from Backup Mode Timing Requirements[1]

Symbol	Parameters	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 established periphery group 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 to 90% of its final value.

73.6. I/O Characteristics

The device features two types of Input/Output (I/O) circuits:

- GPIO
- DDRIO

Unless otherwise specified:

- The following specifications apply to the GPIO type.
- V_{DD} refers to the voltage of the associated power rail of the I/O line, as defined in the Pin Description table. For example, for PA2, V_{DD} refers to the voltage applied on VDDIOP0.
- The GPIO type drive and slewrate controls are set in the PIO_DRIVE and PIO_SLEWR registers, respectively.

73.6.1. I/O DC Characteristics

Table 73.16. Input DC Characteristics

Symbol	Parameters	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
V_{OL}	Low-level output voltage	I_O max	–	0.2 x V_{DD}	V
V_{OH}	High-level output voltage	I_O max	0.8 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} = 0; 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)	Digital Input mode	–	4	pF
V_{hys}	Input hysteresis(1)	–	0.15	–	V

Note:

1. Simulation data

Table 73.17. Output DC Characteristics (1.7V < V_{DD} < 1.9V)

Symbol	Parameters	Conditions		Min	Max	Unit
I_{OH} or I_{OL}	High-level or low-level output current	Any	I_{OL}: V_{OL} = 0.2 x V_{DD} I_{OH}: V_{OH} = 0.8 x V_{DD}	–	–	–
		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

Table 73.18. Output DC Characteristics (3.0V < V_{DD} < 3.6V)

Symbol	Parameters	Conditions		Min	Max	Unit
I_{OH} or I_{OL}	High-level or low-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	
			Drive = 0, Slewrate = 1	–	3	
			Drive = 1, Slewrate = 1	–	6	

Note:

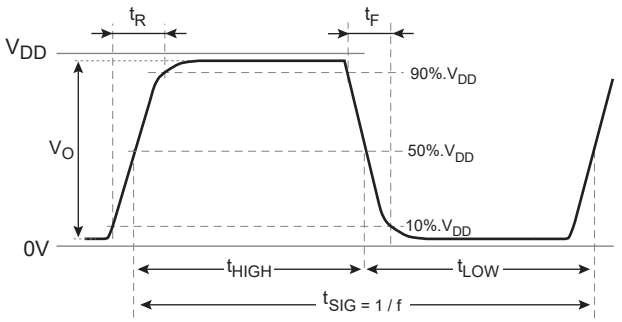
1. Simulation data

73.6.2. I/O AC Characteristics

73.6.2.1. Output Driver AC Characteristics

The following figure provides the timing definitions required to specify the maximum operating frequency of an output driver.

Figure 73.6. 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 Table 73.20 and Table 73.21, the maximum operating frequency f_{MAX} guarantees 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 and slewrate programming. The effective maximum output frequency of a specific output line may be limited by the peripheral that drives this line.

Table 73.20 and Table 73.21 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 < 3.6V
  - 3.0V < VDD < 3.6V
- Two Drive settings: 0 and 1
- Two Slewrate settings for the GPIO type: 0 and 1. The DDRIO drivers do not have slewrate settings and rather use an autocalibration setting.

For the GPIO drivers, the table below shows 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 73.19. Recommended GPIO Drive and Slewrate Settings versus GPIO Use Case

VDD Range	Low-Speed (f_{GPIO} < 50 MHz)(1)	High-Speed (f_{GPIO} ≥ 50 MHz)
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. Indicative value. See Table 73.20 and Table 73.21 for accurate maximum frequency specifications.

Table 73.20. Output Driver AC Characteristics (1.7V < V_{DD} < 1.9V, C_L = 10 pF)

Symbol	Parameters	Conditions		Min	Max	Unit
t_R / 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	10.3	ns
			Drive = 1, Slewrate = 1	2.1	7.0	ns
f_{MAX}	Maximum frequency(2)(3)(4)	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

Notes:

1. Measured between V_O = 10% V_{DD} and V_O = 90% V_{DD}
2. Simulation data
3. f_{MAX} = 0.75 / (t_R+t_F). Frequency numbers are rounded for legibility.
4. f_{MAX} may be limited by the peripheral that drives the I/O line.

Table 73.21. Output Driver AC Characteristics (3.0V < V_{DD} < 3.6V, C_L = 10 pF)

Symbol	Parameter	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
f_{MAX}	Maximum frequency(2)(3)(4)		Drive = 0, Slewrate = 0	90	–	MHz
			Drive = 1, Slewrate = 0	130	–	MHz
			Drive = 0, Slewrate = 1	25	–	MHz
			Drive = 1, Slewrate = 1	50	–	MHz

Notes:

1. Measured between V_O = 10% V_{DD} and V_O = 90% V_{DD}
2. Simulation data
3. f_{MAX} = 0.75 / (t_R+t_F). Frequency numbers are rounded for legibility.
4. f_{MAX} may be limited by the peripheral that drives the I/O line.

73.6.2.2. Input AC Characteristics

The following table provides the input requirements of the signals connected to I/O lines when they are configured as digital inputs. 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 Table 73.16 apply.

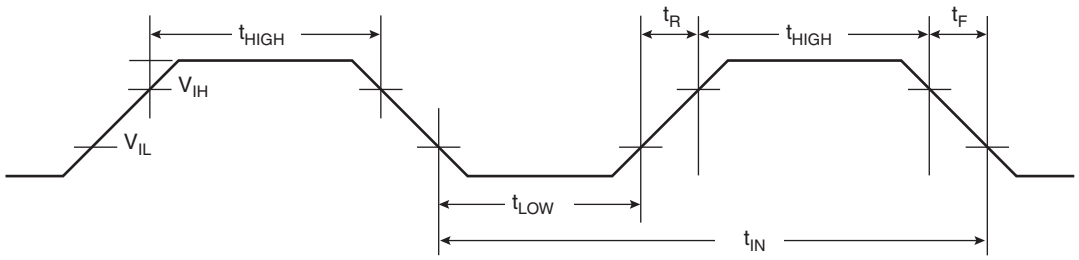
Table 73.22. Input AC Characteristics

Symbol	Parameters	Conditions	Min	Max	Unit
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	ms

Note:

1. The maximum input frequency may be limited by the peripheral receiving this signal.

Figure 73.7. Digital Input AC Characteristics



73.6.3. DDR I/O Calibration and DDR Voltage Reference

73.6.3.1. DDR I/O Calibration

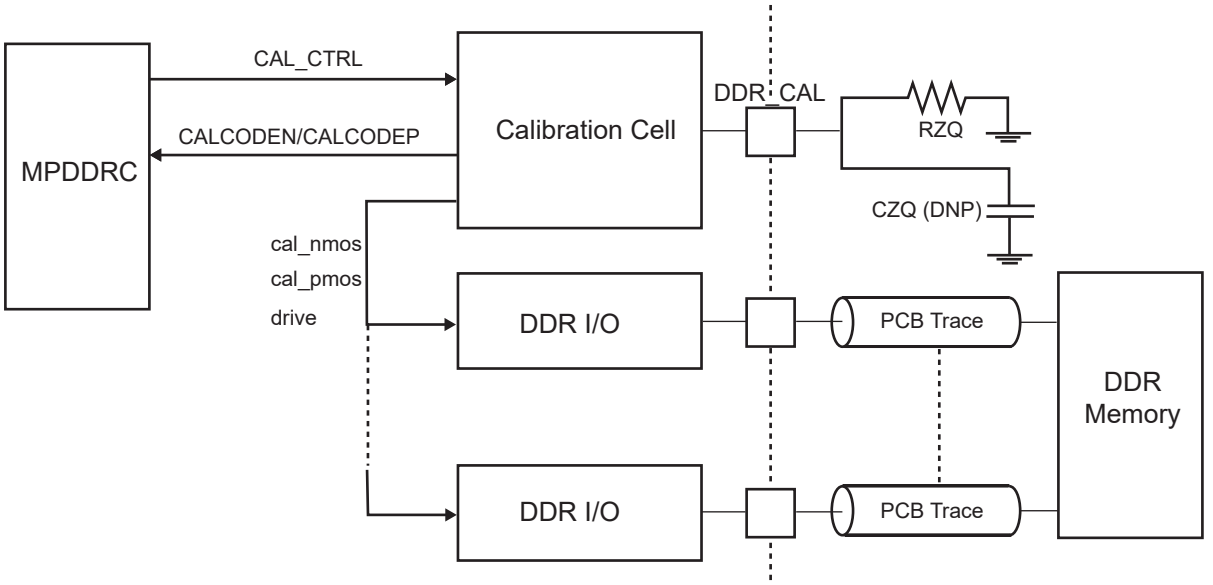
The DDR I/Os embed an automatic impedance matching control to avoid overshoots and reach the best performance levels depending on the bus load and external memories. A serial termination connection scheme, where the driver has an output impedance matched to the characteristic impedance of the line, is used to improve signal quality and reduce EMI.

One specific analog input, DDR_CAL, is used to calibrate all DDR I/Os.

The MPDDRC supports the ZQ calibration procedure used to calibrate the SAM9X7 Series DDR I/O drive strength and the commands to set up the external memory device drive strength (refer to DDR-SDRAM Controller (MPDDRC)).

Note: All DDRC pads are calibrated to 50 ohms.

Figure 73.8. DDR Calibration Cell



The calibration cell provides an input pin, DDR_CAL, loaded with the following resistor RZQ value:

- 20 KΩ ±1% for DDR2 and DDR3L

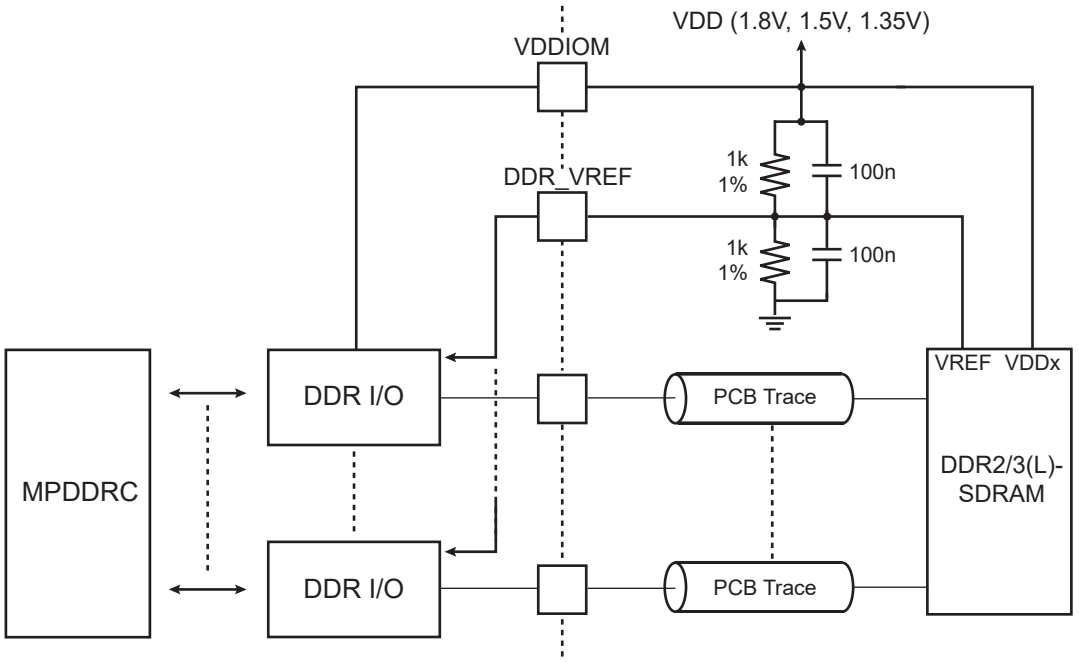
The CZQ capacitor must not be mounted.

DDR I/Os calibration must be done before initializing the DDR-SDRAM. The calibration is performed by software, via the SFR_CAL1 register. Refer to DDR-SDRAM Controller (MPDDRC) for further details.

73.6.3.2. DDR Voltage Reference

The DDR_VREF pin serves as a voltage reference input for the DDR I/Os when DDR2, DDR3 or DDR3L external SDRAM memories are used.

Figure 73.9. DDR_VREF Recommended Schematic with DDR2/3(L)-SDRAM



73.7. Digital Peripheral Characteristics

73.7.1. DDR-SDRAM Controller (MPDDRC)

The MPDDRC complies with the following JEDEC standards:

- DDR3-SDRAM: JESD79-3C with operating frequencies up to 266 MHz
- DDR3L-SDRAM: JESD79-3-1A with operating frequencies up to 266 MHz
- DDR2-SDRAM: JESD79-2F with operating frequencies up to 266 MHz

The physical interface (PCB layout) between the processor and its memory has a major impact on the signal integrity. Microchip provides IBIS models of the SAM9X7 Series device and strongly recommends verifying this processor-memory interface on a PCB simulation tool. For design guidance, refer to the Microchip application note “SAM9X75 Hardware Design Considerations” (AN4962), available on www.microchip.com.

The following table provides the recommended settings on the MPDDRC_RD_DATA_PATH.SHIFT_SAMPLING field depending on the memory type and on its operating frequency.

Table 73.23. SHIFT_SAMPLING Settings

SDRAM Type	SDRAM Clock Frequency (MHz)	SHIFT_SAMPLING
DDR2-SDRAM	$125 \le f_{SDRAM\_CLK} \le 266$	1
DDR3(L)-SDRAM (DLL on)	$f_{SDRAM\_CLK} = 266$	3
DDR3(L)-SDRAM (DLL off)	$f_{SDRAM\_CLK} \le 200$	3

73.7.2. Secure Digital MultiMedia Card Controller (SDMMC)

The SDMMC complies with the following specifications:

- SD Host Controller Standard Specification Version 3.00

- Embedded MultiMedia Card Specification (e.MMC) Version 4.51
  - e.MMC default speed (maximum SDCLK frequency = 26 MHz)
  - e.MMC high speed (maximum SDCLK frequency = 52 MHz)
  - e.MMC high speed DDR (maximum SDCLK frequency = 52 MHz)
- SD Memory Card Specification Version 3.00
- SDIO Specification Version 3.00
  - SD/SDIO default speed (maximum SDCLK frequency = 25 MHz)
  - SD/SDIO high speed (maximum SDCLK frequency = 50 MHz)

73.7.3. Quad Serial Peripheral Interface (QSPI)

The following timings are given for the Serial Memory mode (QSPI_MR.SMM=1). In this mode, only SPI mode 0 is supported. Refer to Quad Serial Peripheral Interface (QSPI) for more details.

Figure 73.10. QSPI Host Mode 0 in Single Data Rate

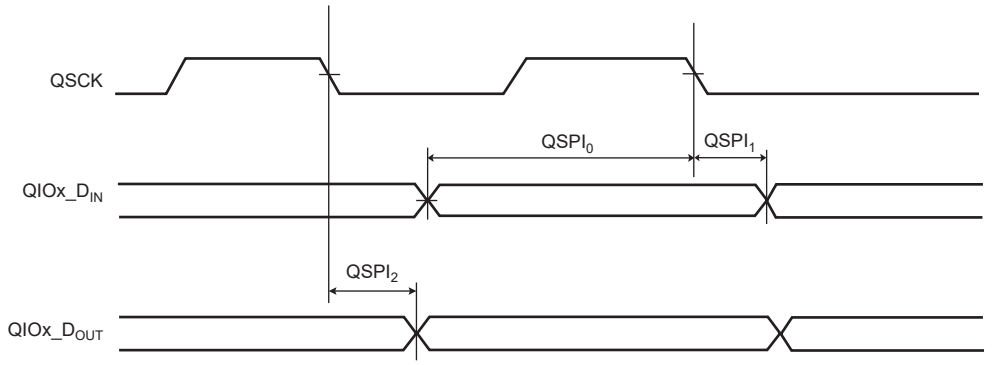


Figure 73.11. QSPI Host Mode in Double Data Rate Mode

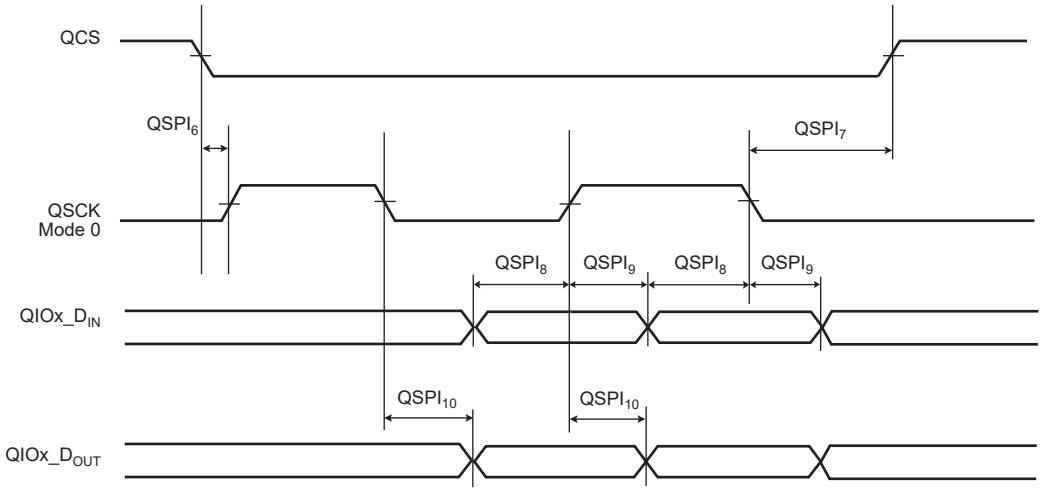
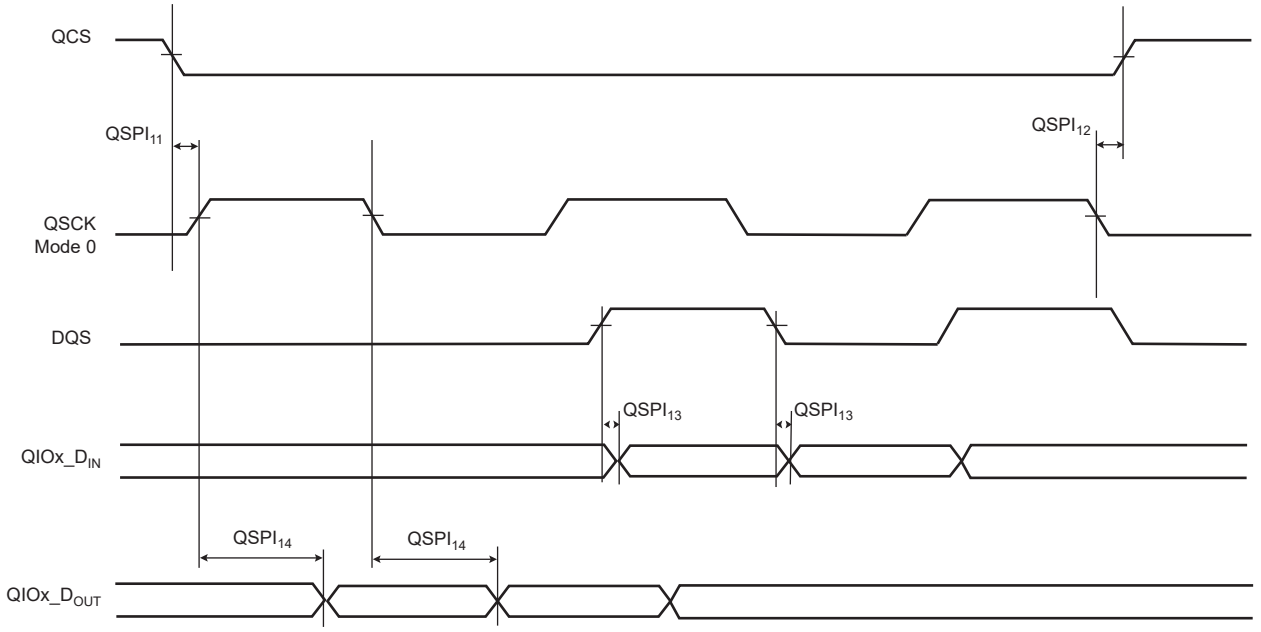


Figure 73.12. QSPI Host Mode in Double Data Rate Mode with DQS



73.7.3.1. 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**

  $$f_{QSCK} = \frac{1}{(QSPI_2 + t_{SU\_CLIENT})}$$

  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} = 100 MHz applies in this case.

- **Host Write in Dual Data Rate Mode**

  $$f_{QSCK} = \frac{0.5}{(QSPI_8 + t_{SU\_CLIENT})}$$

  Where t_{SU_CLIENT} is the input setup time of the client device.

- **Host Read in Dual Data Rate Mode**

  $$f_{QSCK} = \frac{1}{2x(QSPI_{10} + t_{VALID})}$$

  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 $QSPI_{10}$ = 0.7 ns, the formula returns a value of 74 MHz, therefore f_{QSCK} = 57.7 MHz applies in this case.

73.7.3.2.QSPI Timings

For the QSPI0 instance, the timings shown in the tables below are provided in the following domains:

- 1.8V domain: VDDQSPI from 1.7V to 1.9V, maximum external capacitor: 10 pF, Drive = 1, Slewrate = 0
- 3.3V domain: VDDQSPI from 3.0V to 3.6V, maximum external capacitor: 10 pF, Drive = 0, Slewrate = 0

Table 73.24. QSPI Timings in Single Data Rate Mode (STR)

Symbol	Parameter	Conditions	Min	Max	Unit
f_{QSCK}	QSCK maximum frequency	–	–	100	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 delay	–	0	1	ns

Table 73.25. QSPI Timings in Double Data Rate Mode (DTR)

Symbol	Parameter	Conditions	Min	Max	Unit
f_{QSPI}	QSCK operating frequency	–	–	57.7	MHz
t_{QSCK_MIN}	Minimum SPCK period	–	17.3	–	ns
$QSPI_6$	CS low before QSCK edge (rising or falling) (1)	–	1.1	–	ns
$QSPI_7$	QSCK edge (rising or falling) to CS high(2)	–	0.2	1	ns
$QSPI_8$	QIOx input data setup to QSCK edge (rising or falling)	–	1.4	–	ns
$QSPI_9$	QIOx input data hold after QSCK edge (rising or falling)	–	0.9	–	ns
$QSPI_{10}$	QSCK edge (rising or falling) to QIOx delay	–	0	0.7	ns

Notes:

1. Refer to DLYCS and DLYBS descriptions in Quad Serial Peripheral Interface (QSPI) for more configuration details.
2. Refer to DLYBCT description in Quad Serial Peripheral Interface (QSPI) for more configuration details.

Table 73.26. QSPI Timings in DTR Mode with Data Strobe (DQS)

Symbol	Parameter	Conditions	Min	Max	Unit
f_{QSPI}	QSCK operating frequency	–	–	100	MHz
t_{QSCK_MIN}	Minimum SPCK period	–	10	–	ns
$QSPI_{11}$	CS low before QSCK edge (rising or falling) (1)	–	1.1	–	ns
$QSPI_{12}$	QSCK edge (rising or falling) to CS high(2)	–	0.2	1	ns
$QSPI_{13}$	QIOx input skew to DQS edge (rising or falling)	–	0	2.2	ns
$QSPI_{14}$	QSCK edge (rising or falling) to QIOx delay	–	0	$t_{QSCK}/4 + 0.7$	ns

Notes:

1. Refer to DLYCS and DLYBS descriptions in Quad Serial Peripheral Interface (QSPI) for more configuration details.
2. Refer to DLYBCT description in Quad Serial Peripheral Interface (QSPI) for more configuration details.

73.7.4. Flexible Serial Communication Controller (FLEXCOM) - SPI

In Figure 73.14 and Figure 73.15, the MOSI line shifting edge is represented with a 0 hold time. 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 Figure 73.13, the device sampling point extends the propagation delay (t_p) for device 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 device working in mode 0 can be safely driven if the SPI host is configured in mode 0.

Figure 73.13. FLEXCOM in SPI Mode: MISO Capture in Host Mode

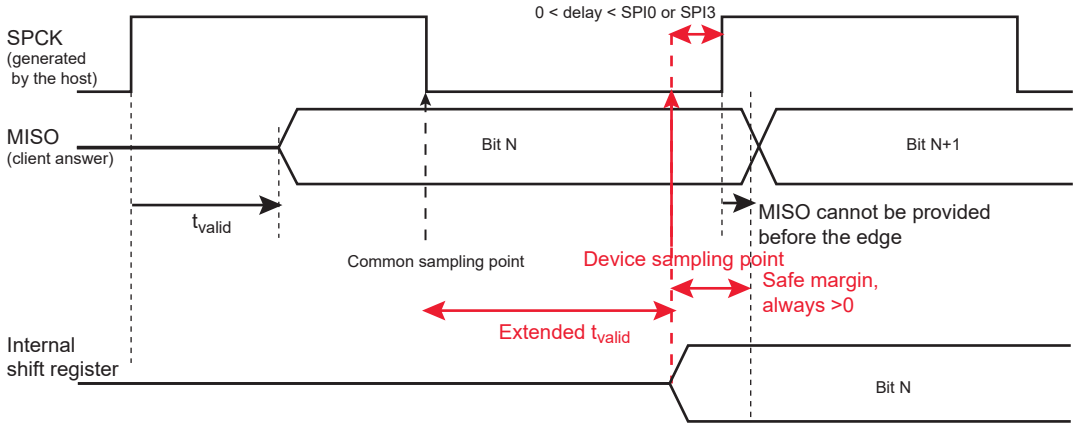


Figure 73.14. FLEXCOM in SPI Host Modes 1 and 2

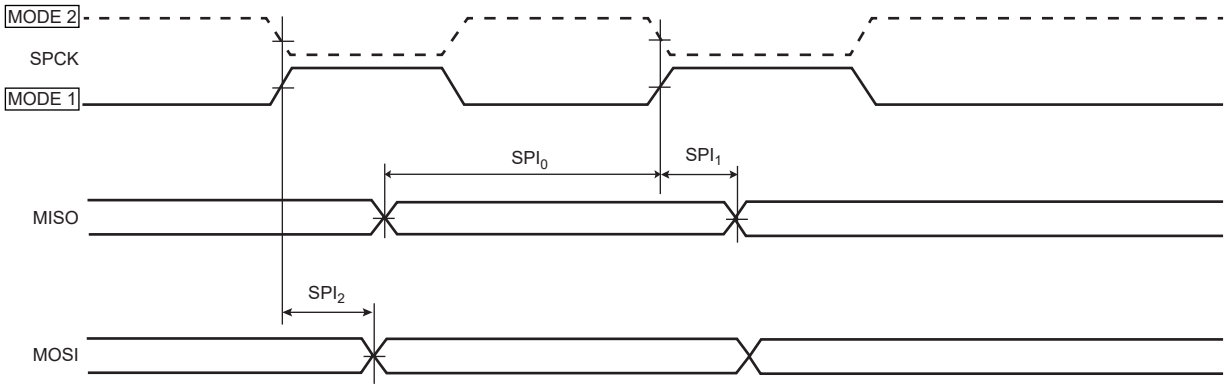


Figure 73.15. FLEXCOM in SPI Host Modes 0 and 3

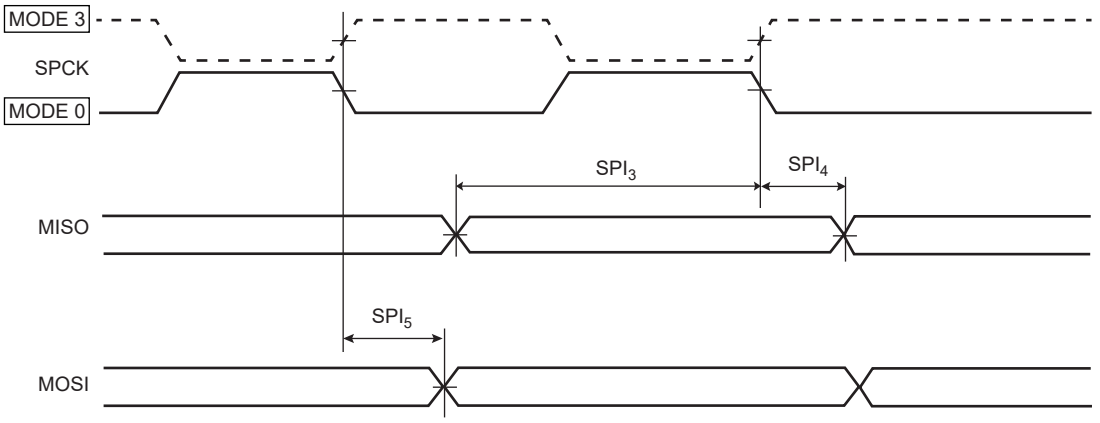


Figure 73.16. FLEXCOM in SPI Client Modes 0 and 3

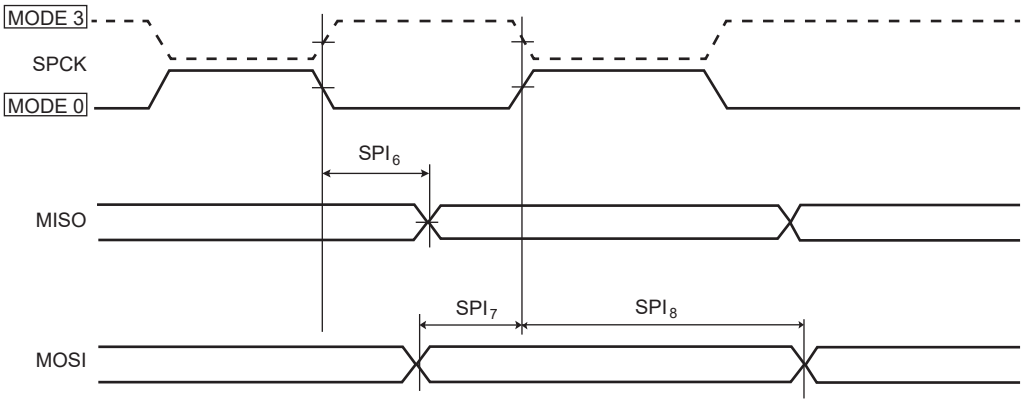


Figure 73.17. FLEXCOM in SPI Client Modes 1 and 2

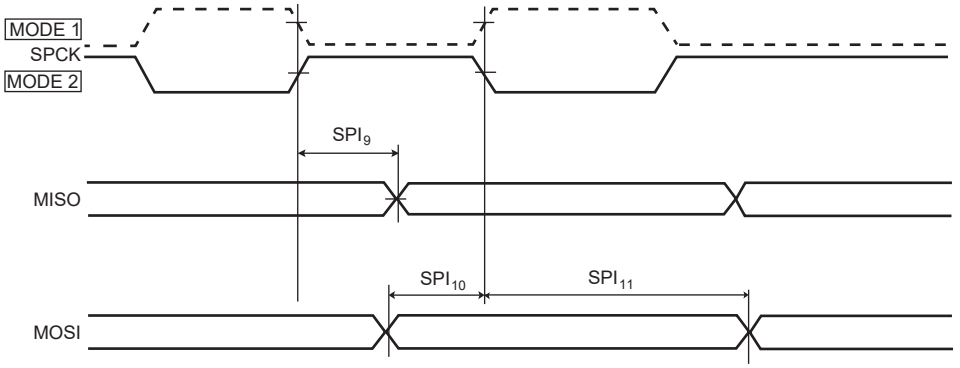
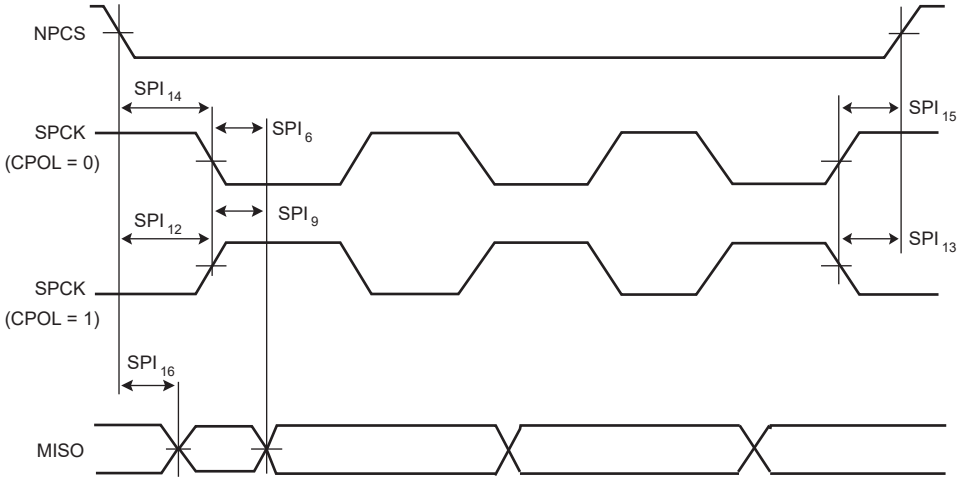


Figure 73.18. FLEXCOM in SPI Client - NPCS Timings



73.7.4.1. FLEXCOM SPI Timings

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 73.27. FLEXCOM SPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
Host Mode					
f_{SPCK}	SPCK operating frequency		–	44	MHz
t_{SPCK}	SPCK period	–	22.5	–	ns
SPI_0	MISO setup time before SPCK rises/falls	–	16.5	–	ns
SPI_1	MISO hold time after SPCK rises/falls	–	0.0	–	ns
SPI_2	SPCK rising to MOSI delay	–	-4.5	4.5	ns
SPI_3	MISO setup time before SPCK rises/falls	–	16.5	–	ns
SPI_4	MISO hold time after SPCK rises/falls	–	0.0	–	ns
SPI_5	SPCK falling to MOSI delay	–	-4.5	4.5	ns
Client Mode					
f_{SPCK}	SPCK operating frequency		–	25	MHz
t_{SPCK}	SPCK period	–	40.0	–	ns
SPI_6	SPCK falling to MISO delay	–	4.0	16.0	ns
SPI_7	MOSI setup time before SPCK rises/falls	–	2.0	–	ns
SPI_8	MOSI hold time after SPCK rises/falls	–	2.0	–	ns
SPI_9	SPCK rising to MISO delay	–	4.0	16.0	ns
SPI_{10}	MOSI setup time before SPCK rises/falls	–	2.0	–	ns
SPI_{11}	MOSI hold time after SPCK rises/falls	–	2.0	–	ns
SPI_{12}	NPCS setup to SPCK rising	–	42.0	–	ns
SPI_{13}	NPCS hold after SPCK rises/falls	–	42.0	–	ns
SPI_{14}	NPCS setup to SPCK falling	–	42.0	–	ns
SPI_{15}	NPCS hold after SPCK rises/falls	–	42.0	–	ns
SPI_{16}	NPCS falling to MISO valid	–	42.0	–	ns

73.7.5. Flexible Serial Communication Controller (FLEXCOM) - USART

Table 73.28. 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).

73.7.6. Flexible Serial Communication Controller (FLEXCOM) - TWI

The TWI communicates in Standard, Fast, Fast Mode Plus and High-Speed (HS) modes subject to the following:

- When the registers FLEX_TWI_CWGR, FLEX_TWI_HSCWGR and FLEX_TWI_MMR are programmed in the TWI Controller
- When pull-ups (Rp) are computed to achieve the required rise time according to the total Cbus capacitance.

Possible limitations are given in the table and figures below.

Table 73.29. Two-Wire Interface I/O Characteristics

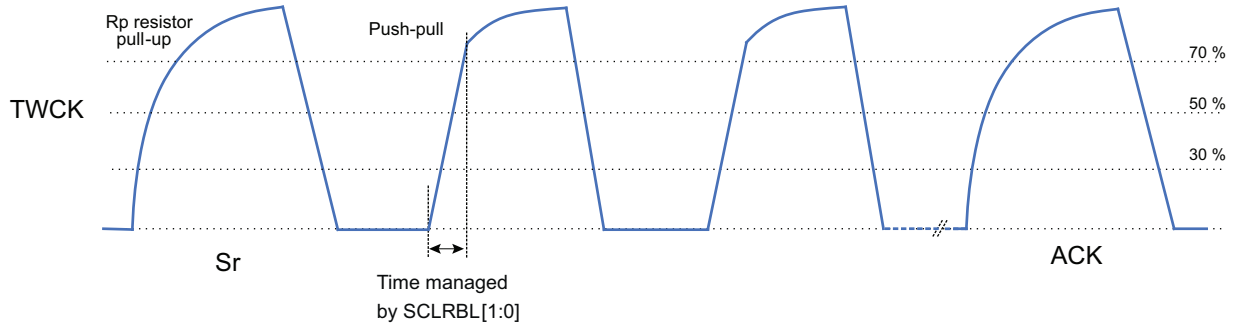
Symbol	Parameter	Conditions	Min	Max	Unit
V_{IL}, V_{IH}	Low-level/High-level Input Voltage	All modes	See note (1)		V
V_{OL}, V_{OH}	Low-level/High-level Output Voltage	All modes	See note (2)		V
I_{OL}	Low-level Output Current	Standard Fast Fast-Mode Plus	See notes (2, 3)		mA
		High-speed Mode	See note (2)		
t_f	Output fall time of both TWD and TWCK signals V_{IHmin} to V_{ILmax}	Standard Fast Fast-Mode Plus	See note (4)		ns
t_{fTWCK}, t_{fTWD}	Fall time of both TWD and TWCK signals	High-speed Mode			
t_{rTWCK}	Rise time of TWCK, rise boost mode (push-pull mode)	SCLRBL > 0	See note (5)		ns

Notes:

1. See the table Input DC Characteristics.
2. See the tables Output DC Characteristics (1.7V < VDD < 1.9V) and Output DC Characteristics (3.0V < VDD < 3.6V). If $R_{p(min)}$ as specified in the I2C specification must be overridden (stronger) to meet the timing specification, the IBIS model of the product can be used to extract $V_{OL(max)}$ vs I_{OL}.
3. 20 mA is not supported for Fast Mode Plus.
4. Fall time of FLEXCOM I/O buffers are not I2C-compliant. Fall time can be increased by adding series resistors (see series protection resistors in the I2C specification). Rs < 1 KΩ depending on Cbus and Rp to respect V_{OL}, V_{IL} on the bus. Use Drive disabled (set to 0) and Slew rate enabled (set to 1) on the corresponding I/O. In the I/O AC Characteristics section, see falling times given for a 10 pF load. IBIS models of the product can be used to choose an adequate Rs value by simulating different drive/slew rate combinations with respect to the Cbus load.

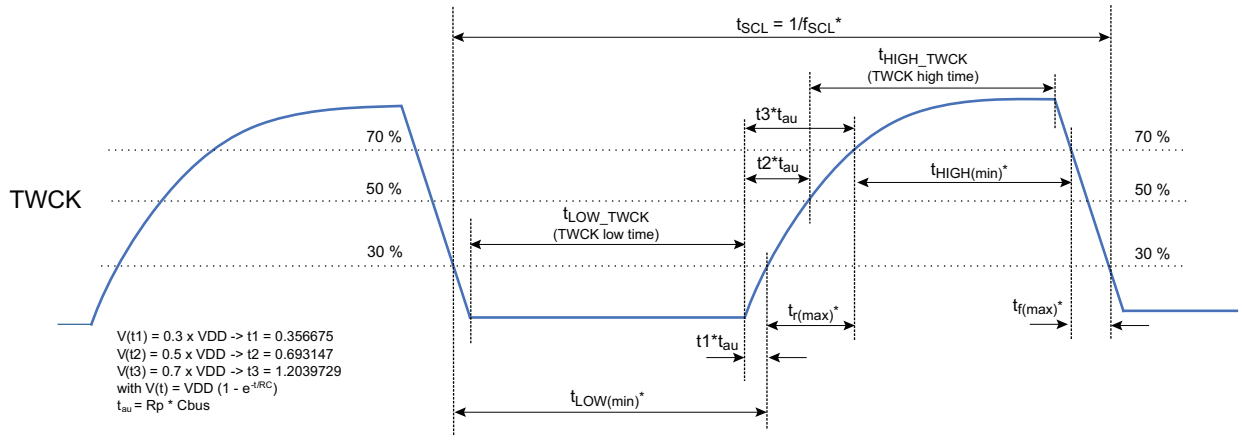
5. When in High-speed mode, the TWI Clock signal (TWCK) rise can be boosted to meet a very fast rise time of the parameter t_{rCL} of the I2C specification when the Rp value on TWCK is lower than $R_{p(min)}$.
   Bits in FLEX_TWI_MMR.SCLRBL must be programmed according to t_{MCKx} to respect these timings. This gives the number of peripheral clock periods during which the SCL rise is boosted, i.e. driven in Push-pull mode. Note that the Cbus value and the drive and slew rate settings still influence the rise time. Rise boost usage is more suited with high Cbus values.
   Note: MCKx refers to the MCK associated with the FLEXCOM. Refer to the Domain Clock column in the table “Peripheral Identifiers”.

Figure 73.19. TWCK High-Speed Mode with Rise Boost Enabled



The figure below shows programmable register fields (CLDIV/HSCLDIV, CHDIV/HSCHDIV and optional CKDIV/HSCKDIV) in relation to t_{LOW}, t_{HIGH} , t_r, t_f, f_{SCL}, t_{rCL}, t_{rCL1}, t_{fCL} from the I2C specification.

Figure 73.20. TWCK Timings Characteristics



Note: Symbols with an asterisk (*) are I2C specification symbols: t_{LOW}, t_{HIGH}, t_r , t_f.

Note: For High-speed mode, $t_r = t_{rCL}$ or t_{rCL1} and $t_f = t_{fCL}$ from the I2C specification.

Important: $t_{SCL} \cong t_{LOW(min)} + t_{r(max)} + t_{HIGH(min)} + t_{f(max)}$

73.7.6.1.FLEXCOM TWI Timings

Table 73.30. TWI Interface Host Mode Timings

Symbol	Parameter	Conditions	Min	Max	Unit
t_{LOW_TWCK}	TWCK low time	–	See notes (1, 3, 4)	–	µs
t_{HIGH_TWCK}	TWCK high time	–	See notes (2, 3, 4)	–	µs
f_{TWCK}	TWCK frequency(5)	Standard Fast Fast-Mode Plus High-speed Mode	0	0.1 0.4 1 3.4	MHz
$t_{HD(STA)}$	Hold time (repeated) START condition	Standard Fast Fast-Mode Plus	t_{HIGH_TWCK}	–	ns
		High-speed Mode	FLEX_TWI_CWGR.BRSRCCLK = 1: t_{LOW_TWCK}(6) FLEX_TWI_CWGR.BRSRCCLK = 0: 2 x t_{HIGH_TWCK}	–	ns
$t_{SU(STA)}$	Setup time for a repeated START condition	Standard Fast Fast-Mode Plus	t_{LOW_TWCK}	–	ns
		High-speed Mode	FLEX_TWI_CWGR.BRSRCCLK = 1: t_{LOW_TWCK} FLEX_TWI_CWGR.BRSRCCLK = 0: 2 x t_{LOW_TWCK} (6)	–	ns
$t_{HD(DAT)}$	Data hold time	Standard Fast Fast-Mode Plus	2 x t_{MCKx}	(HOLD + 3) × t_{MCKx}	ns
		High-speed Mode	0	150 @ f_{TWCK} max = 1.7 MHz 70 @ f_{TWCK} max = 3.4 MHz	ns
$t_{SU(DAT)}$	Data setup time	Standard Fast Fast-Mode Plus	t_{LOW} – (HOLD + 3) × t_{MCKx}	–	ns
		High-speed Mode	t_{LOW_TWCK}	–	ns
$t_{SU(STO)}$	Setup time for STOP condition	Standard Fast Fast-Mode Plus	FLEX_TWI_CWGR.BRSRCCLK = 1: t_{LOW_TWCK} FLEX_TWI_CWGR.BRSRCCLK = 0: t_{HIGH_TWCK}	–	ns
		High-speed Mode	FLEX_TWI_CWGR.BRSRCCLK = 1: t_{HIGH+1} FLEX_TWI_CWGR.BRSRCCLK = 0: t_{HIGH_TWCK}	–	ns
t_{BUF}	Bus free time between a STOP and a START condition	Standard Fast Fast-Mode Plus	t_{LOW_TWCK}	–	ns

Notes:

1. TWCK low time (t_{LOW_TWCK}) ≥ t_{LOWmin} - t1*t_{au} → HSCLDIV/CLDIV = ((t_{LOW_TWCK} / t_{PERIPH})-3) / $2^{CKDIV\ or\ HSCKDIV}$)
2. TWCK high time (t_{HIGH_TWCK}) ≥ $t_{HIGHmin}$ + (t3-t2)*t_{au} → HSCHDIV/CHDIV = ((t_{HIGH_TWCK} / t_{PERIPH})-3) / $2^{CKDIV\ or\ HSCKDIV}$)
3. t_{TWCK} = (t_{LOW_TWCK} + t1*t_{au}) + $t_{r(max}$ + (t_{HIGH_TWCK} + (t3-t2)*t_{au}) + $t_{f(max)}$
4. The TWCK low/high time formulae in notes 1 and 2 are for Bit Rate Source Clock (BRSRCCLK) = 0 for all modes and RX Digital Filter (FILT) = 0 for Standard, Fast and FM+ Modes only. See FLEX_TWI_CWGR, FLEX_TWI_HSCWGR and FLEX_TWI_FILTR registers for more details.
5. 1.7 MHz (Cbus 400 pF max), 3.4 MHz (Cbus 100 pF max). Timings in the above table involving t_{LOW_TWCK} and/or t_{HIGH_TWCK} may limit the TWCK maximum frequency when interfacing with an I2C client requiring strict I2C timings.
6. If computing HSCLDIV and/or HSCHDIV results in 0, $t_{SU(STA)}$ and/or $t_{HD(STA)}$ is equal to 6x t_{LOW_TWCK} and/or 6x t_{HIGH_TWCK}, respectively.
7. MCKx refers to the MCK associated with the FLEXCOM. Refer to column Domain Clock in the table “Peripheral Identifiers”.

Table 73.31. Two-Wire Interface Client High-Speed Mode Timings

Symbol	Parameter	Conditions	Min	Max	Unit
f_{TWCK}	TWCK clock frequency	–	–	3.4	MHz
$t_{HD(DAT)}$	Data hold time	f_{TWCK} = 1.7 MHZ	0	70	ns
		f_{TWCK} = 3.4 MHz	0	150	
$t_{SU(DAT)}$	Data setup time	f_{TWCK} = 1.7 MHZ f_{TWCK} = 3.4 MHz	10	–	ns

73.7.7. Synchronous Serial Controller (SSC)

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

Figure 73.21. SSC Transmitter, TK and TF in Output

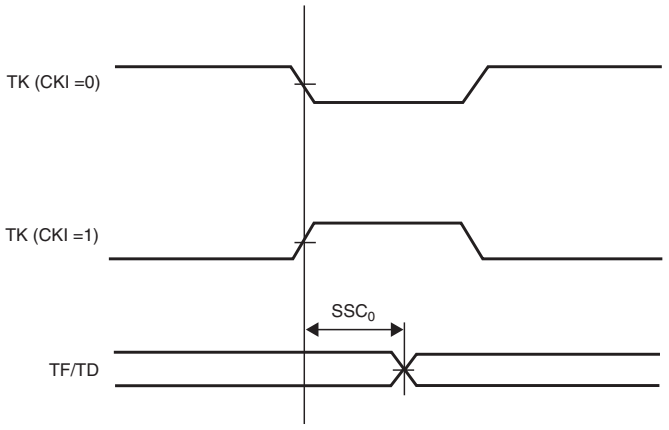


Figure 73.22. SSC Transmitter, TK in Input and TF in Output

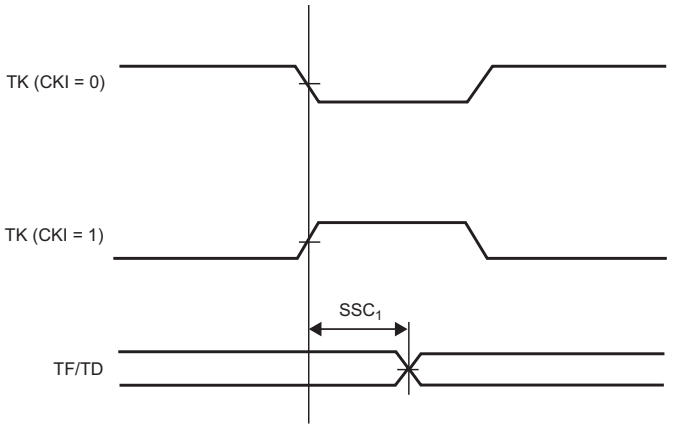


Figure 73.23. SSC Transmitter, TK in Output and TF in Input

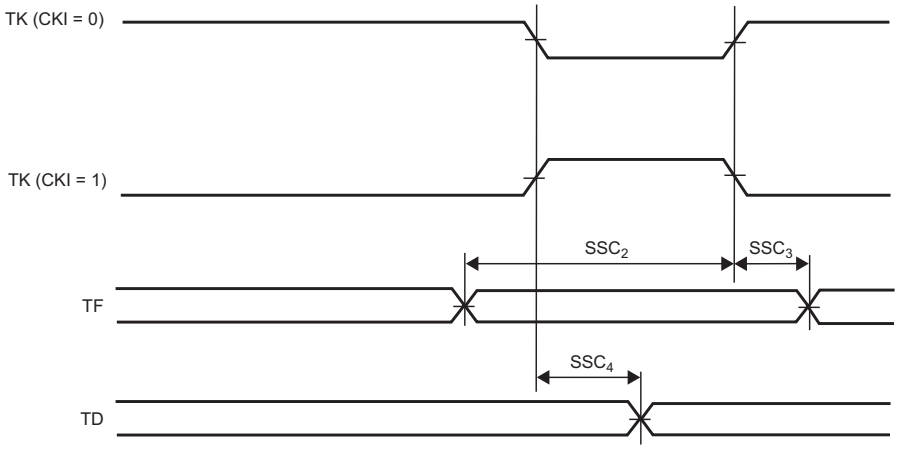


Figure 73.24. SSC Transmitter, TK and TF in Input

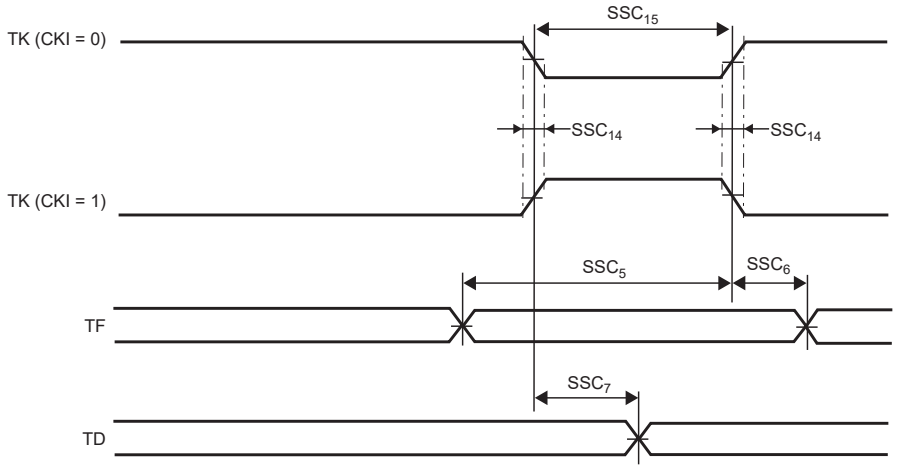


Figure 73.25. SSC Receiver RK and RF in Input

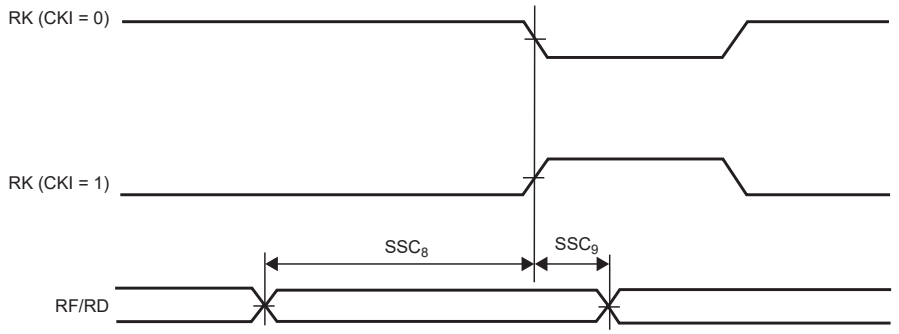


Figure 73.26. SSC Receiver, RK in Input and RF in Output

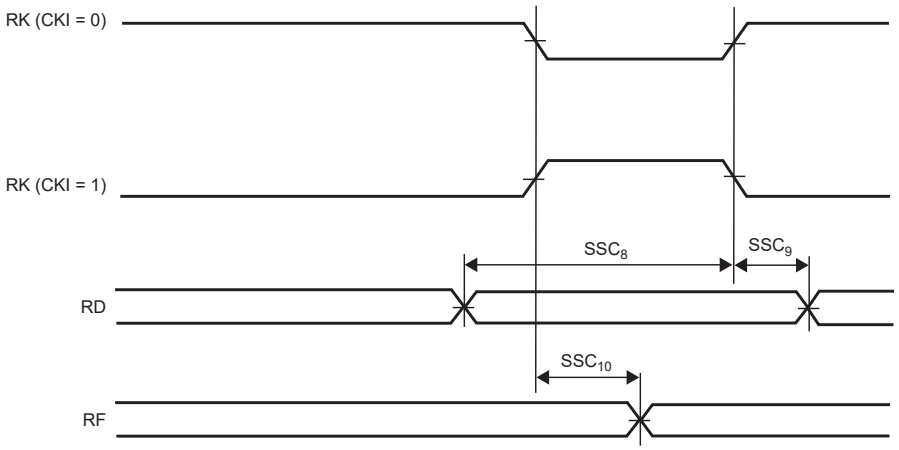


Figure 73.27. SSC Receiver, RK and RF in Output

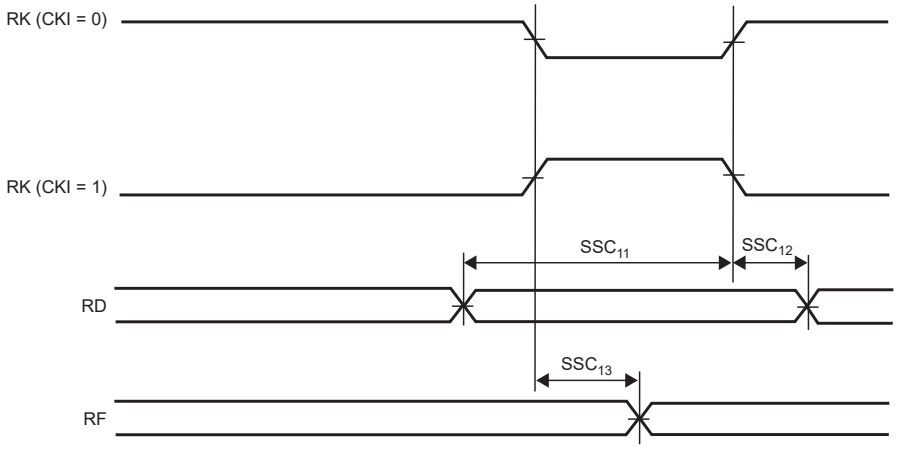


Figure 73.28. SSC Receiver, RK in Output and RF in Input

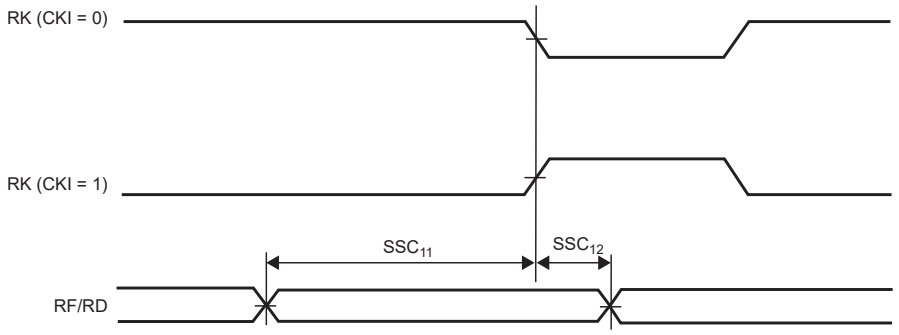


Table 73.32. SSC Timings

Symbol	Parameter	Conditions	Min	Max	Unit
SSC TX Timings - Client Mode (TK Input)					
f_{TK}	TK frequency (TK input)	–	–	25.0	MHz
t_{TK}	Minimum TK period	–	40.0	–	ns
SSC_1	TK edge to TF/TD	–	4.0	15.0	ns
SSC_5	TF setup time before TK edge	–	0.0	–	ns
SSC_6	TF hold time after TK edge	–	t_{MCK}	–	ns
SSC_7	TK edge to TF/TD with STDDLY = 0, START = 4,5,7	3 x t_{MCK} +	4.0	15.0	ns
SSC TX Timings - Host Mode (TK Output)					
f_{TK}	TK frequency (TK output)	–	–	26.7	MHz
t_{TK}	Minimum TK period	–	37.5	–	ns
SSC_0	TK edge to TF/TD	–	-3.8	9.4	ns
SSC_2	TF setup time before TK edge	–	13.1	–	ns
SSC_3	TF hold time after TK edge	–	5.3	–	ns
SSC_4	TK edge to TF/TD with STDDLY = 0, START = 4,5,7	2 x t_{MCK} +	-3.8	9.4	ns
SSC_{14}	TK rise time or fall time (TK input)	–	–	10.0	ns
SSC RX Timings - Client Mode (RK Input)					
f_{RK}	RK frequency (RK input)	–	–	25.0	MHz
t_{RK}	Minimum RK period	–	40.0	–	ns
SSC_8	RF/RD setup time before RK edge	–	0.0	–	ns
SSC_9	RF/RD hold time after RK edge	–	t_{MCK}	–	ns
SSC_{10}	RK edge to RF	–	4.0	15.0	ns
SSC RX Timings - Host Mode (RK Output)					
f_{RK}	RK frequency (RK output)	–	–	26.7	MHz
t_{RK}	Minimum RK period	–	37.5	–	ns
SSC_{11}	RD/RF setup time before RK edge	–	13.1	–	ns
SSC_{12}	RD/RF hold time after RK edge (RF input)	–	5.3	–	ns
SSC_{13}	RK edge to RF	–	-3.8	9.4	ns

73.7.8. Inter-IC Sound Multi-Channel Controller (I2SMCC)

I2SMCC 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

Figure 73.29. I2SMCC Timing Diagram in Host Mode

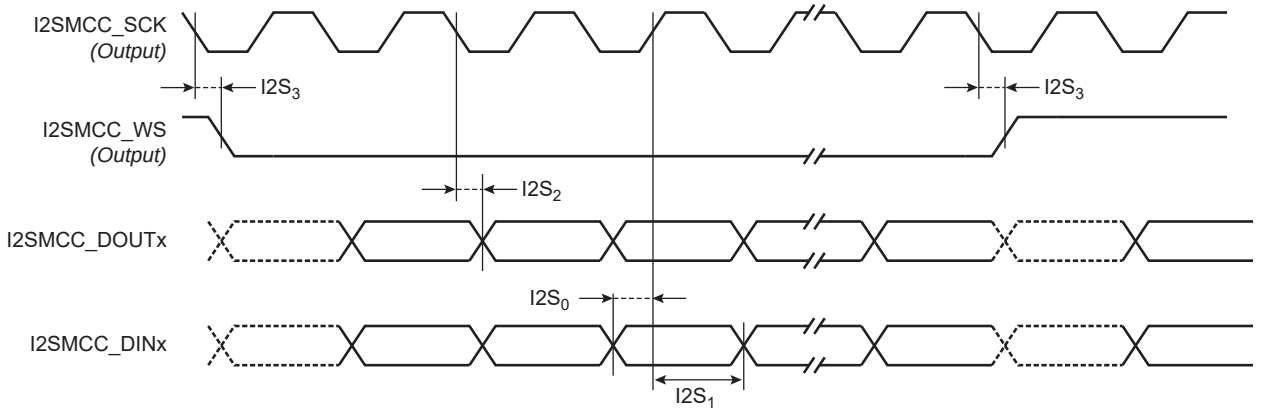


Figure 73.30. I2SMCC Timing Diagram in Client Mode

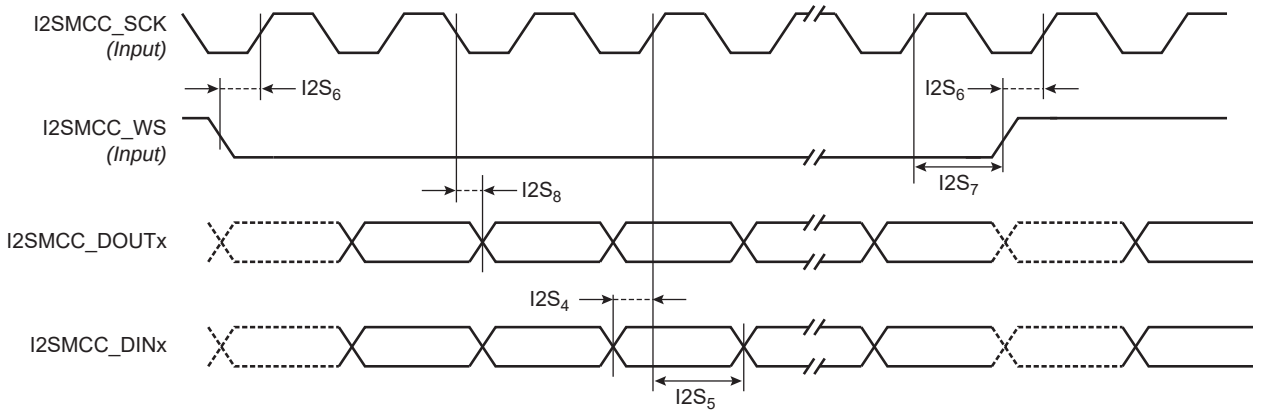


Table 73.33. I2SMCC Timings

Symbol	Parameter	Conditions	Min	Max	Unit
Host Mode					
f_{I2SMCC_SCK}	I2SMCC_SCK frequency	–	–	25.0	MHz
$I2S_0$	I2SMCC_DINx setup time before I2SMCC_SCK rises	–	14.0	–	ns
$I2S_1$	I2SMCC_DINx hold time after I2SMCC_SCK rises	–	2.0	–	ns
$I2S_2$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	–	-2.0	12.0	ns
$I2S_3$	I2SMCC_SCK falling to I2SMCC_WS delay	–	-2.0	12.0	ns
Client Mode					
f_{I2SMCC_SCK}	I2SMCC_SCK frequency	–	–	25.0	MHz
$I2S_4$	I2SMCC_DINx setup time before I2SMCC_SCK rises	–	14.0	–	ns
$I2S_5$	I2SMCC_DINx hold time after I2SMCC_SCK rises	–	2.0	–	ns
$I2S_6$	I2SMCC_WS setup time before I2SMCC_SCK rises	–	14.0	–	ns
$I2S_7$	I2SMCC_WS hold time after I2SMCC_SCK rises	–	2.0	–	ns
$I2S_8$	I2SMCC_SCK falling to I2SMCC_DOUTx delay	–	-2.0	12.0	ns

73.7.9. Ethernet MAC (GMAC)

GMAC 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

Figure 73.31. Ethernet MAC MDIO Interface Timings Characteristics

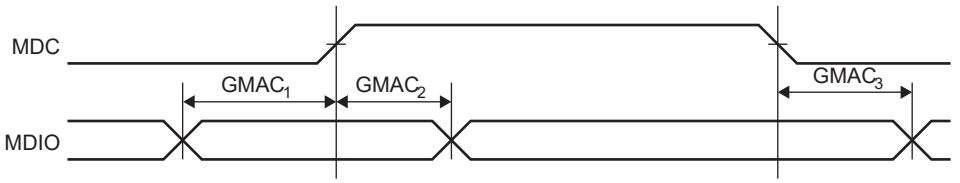


Table 73.34. Ethernet MAC Signals Relative to MDC (MDIO Mode)

Symbol	Parameters	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	0	–	ns
$GMAC_3$	MDC falling edge to MDIO output data valid	0	25	ns

Figure 73.32. Ethernet MAC RMII Mode Timing Diagram

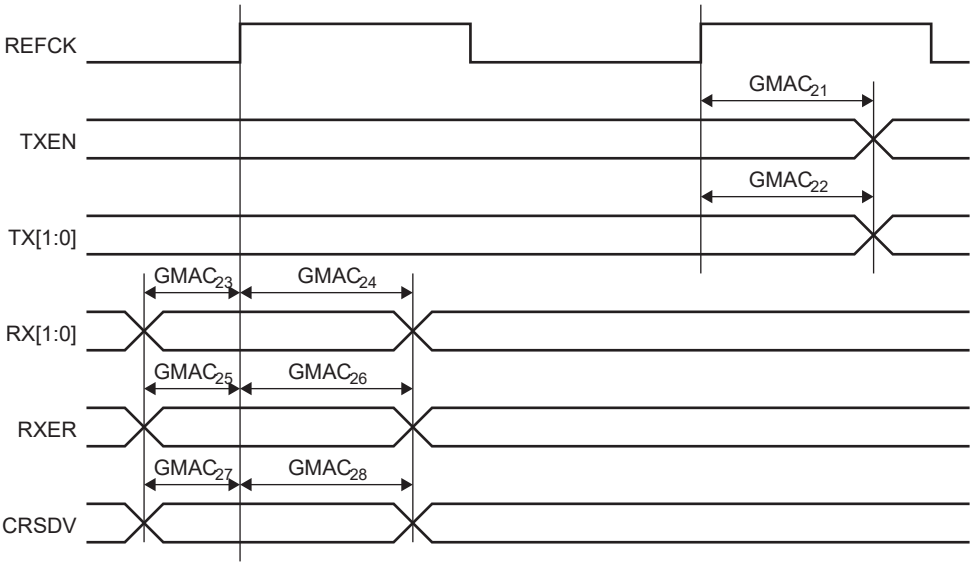


Table 73.35. Ethernet MAC RMII Mode Timing Characteristics

Symbol	Parameters	Conditions	Min	Typ	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
$GMAC_{28}$	Hold for CRSDV from REFCK rising	–	2	–	–	ns

Figure 73.33. Ethernet MAC RGMII Timing Diagram

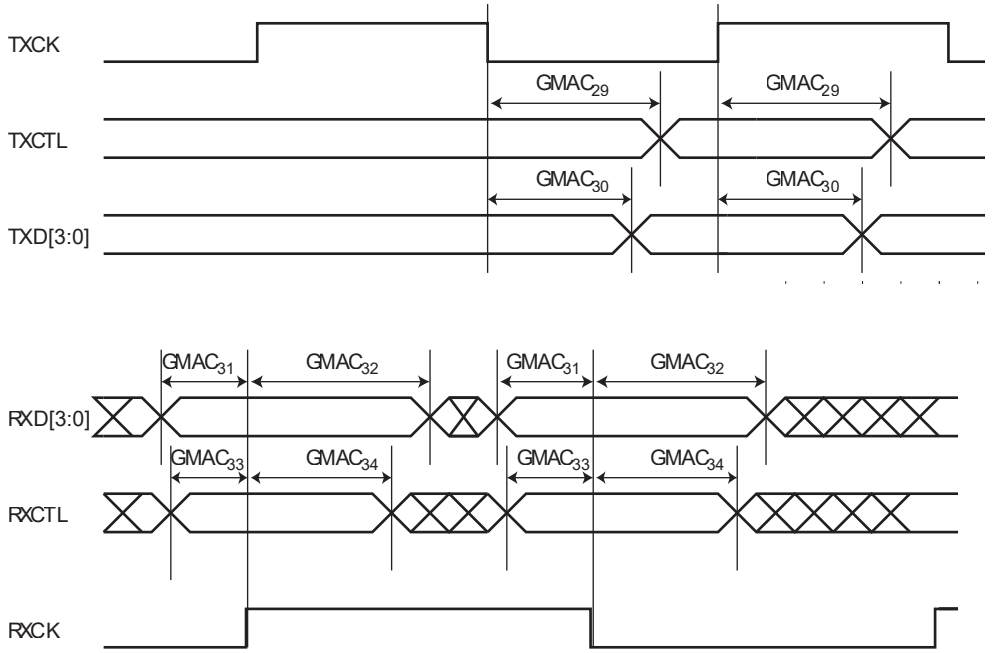


Table 73.36. Ethernet MAC RGMII Mode Timing Characteristics

Symbol	Parameter	Min	Max	Unit
$GMAC_{29}$	TXCTL toggling from TXCK edge	-500	500	ps
$GMAC_{30}$	TXD[3:0] toggling from TXCK edge	-500	500	ps
$GMAC_{31}$	RXD[3:0] Setup time before RXCK edge	1.0	–	ns
$GMAC_{32}$	RXD[3:0] Hold time after RXCK edge	1.0	–	ns
$GMAC_{33}$	RXCTL Setup time before RXCK edge	1.0	–	ns
$GMAC_{34}$	RXCTL Hold time after RXCK edge	1.0	–	ns

73.7.10. Image Sensor Controller (ISC)

ISC 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

Figure 73.34. ISC Timing Diagram

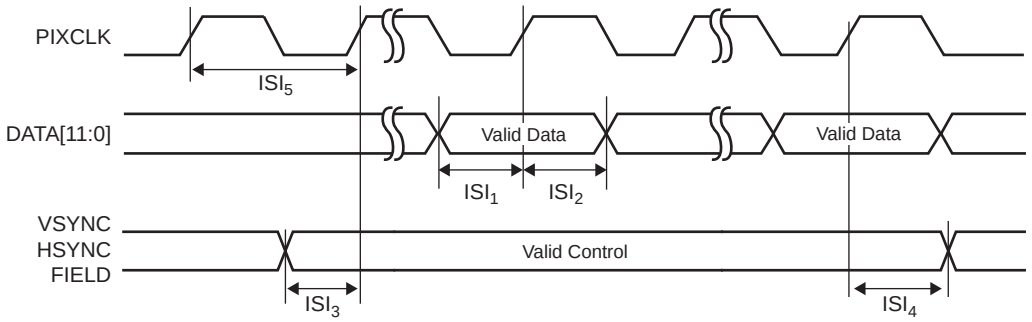


Table 73.37. ISC Timings

Symbol	Parameter	Conditions	Min	Max	Unit
f_{ISC_PCK}	ISC_PCK operating frequency	–	–	100	MHz
ISC1	DATA setup time before ISC_PCK rising edge	–	3.0	–	ns
ISC2	DATA hold time after ISC_PCK rising edge	–	5.0	–	ns
ISC3	VSYNC/HSYNC/FIELD setup time before ISC_PCK rising edge	–	3.0	–	ns
ISC4	VSYNC/HSYNC/FIELD hold time after ISC_PCK rising edge	–	5.0	–	ns

73.7.11. LCD Controller (LCDC)

LCD 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

Figure 73.35. LCD Timings

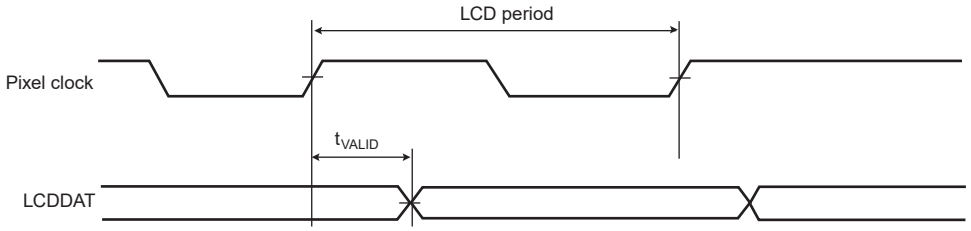


Table 73.38. LCD Timings

Symbol	Parameter	Conditions	Min	Max	Unit
f_{LCD}	Minimum LCD period	–	–	75	MHz
t_{VALID}	Clock to LCDDAT valid output	–	4.0	–	ns

73.8. Analog Peripheral Characteristics

73.8.1. VDDOUT25 Voltage Regulator

Table 73.39. 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+} POR VDDIN33		V
V_{STOP}	Input shutdown voltage	–	V_{IT-} POR VDDIN33		V
t_{START}	Start-up time(1)	From V_{DDIN33} > V_{START} to $V_{DDOUT25}$ > 95% of its final value	–	6	ms
I_{LOAD_EXT}	External DC output current(2)	–	–	1	mA
$V_{DDOUT25}$	VDDOUT25 accuracy	–	2.450	2.550	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 resistance(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. A 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's 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.

73.8.2. Power-On Reset (POR) Cells

Table 73.40. VDDCORE POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDCORE}	Supply voltage range (VDDCORE)	–	0.5	1.21	V
V_{IT-}	Negative-going input threshold voltage (VDDCORE)(1)	Measured with a -10V/s ramp rate	0.68	0.92	V
V_{IT+}	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	100 mV threshold overdrive	–	10	µs

Notes:

1. Simulation data
2. Vhys = VIT+ - VIT-

Table 73.41. VDDIN33 POR Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)	–	–	3.60	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	2.44	2.65	V
V_{hys}	Hysteresis voltage (1)(2)	–	40	60	mV
t_{RES}	Reset time(1)	–	1	6	ms
t_{DET-}	VIT- detection propagation time	100 mV threshold overdrive	–	30	µs

Notes:

1. Simulation data
2. Vhys = VIT+ - VIT-

Table 73.42. VDDBU POR Characteristics

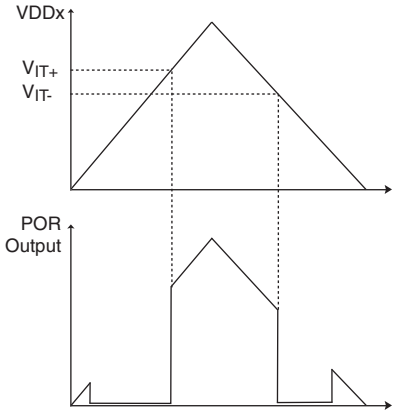
Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)	–	–	3.60	V
V_{IT-}	Negative-going input threshold voltage (VDDBU) (1)	Measured with a -10V/s ramp rate	1.38	1.51	V
V_{IT+}	Positive-going input threshold voltage (VDDBU)(1)	Measured with a +10V/s ramp rate	1.42	1.55	V
V_{hys}	Hysteresis voltage (1)(2)	–	25	45	mV
t_{RES}	Reset time(1)	–	1.2	3.3	ms

Table 73.42. VDDBU POR Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
t_{DET-}	VIT-detection propagation time	100 mV threshold overdrive	–	6	µs

1. Simulation data
2. Vhys = VIT+ - VIT-

Figure 73.36. VDDIN33/VDDCORE/VDDBU Power-On Reset Characteristics



73.8.3. Slow RC Oscillator

Table 73.43. Slow RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)	–	V_{IT-}(1)	3.60	V
f_{SLOWRC}	Output frequency range	V_{DDBU} = 3.3V, T_J = 0 to +50°C	31	33	kHz
		V_{DDBU} = V_{IT-} to 3.6V	29	35	kHz

Note:

1. In this table, V_{IT-} corresponds to the negative-going input threshold voltage of the VDDBU POR (see VDDBU POR Characteristics table). Operation of the device backup section, and in particular of the Slow RC oscillator, is granted down to the VDDBU POR V_{IT-} threshold.

73.8.4. Main RC Oscillator

Table 73.44. Main RC Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	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 f_o = 12MHz	0°C ≤ T_J ≤ +50°C	–	±2	%
		-40°C ≤ T_J ≤ +125°C	–	±5	%
df_0/dV	Output frequency drift with V_{DDIN33}(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

73.8.5. 32.768 kHz Crystal Oscillator

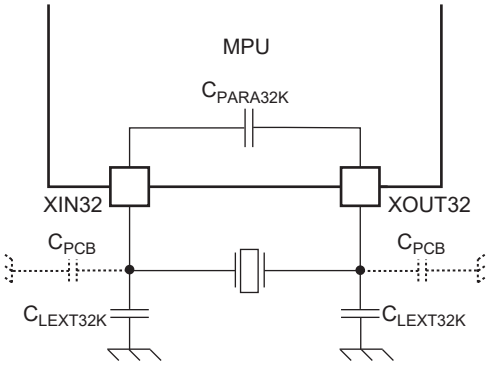
Table 73.45. 32.768 kHz Crystal Oscillator Characteristics

Symbol	Parameter	Conditions		Min	Max	Unit
V_{DDBU}	Supply voltage range (VDDBU)(1)	–		V_{IT-}	3.60	V
I_{DDBU}	Current consumption (VDDBU)(2)	–		–	3.1	µA
t_{START}	Start-up time(2)	R_S(3) < 50 kΩ	C_M = 0.6 fF	–	3.7	s
			C_M = 3 fF	–	0.8	s
		R_S(3) < 90 kΩ	C_M = 0.6 fF	–	5.0	s
			C_M = 3 fF	–	1.0	s
f_{OSC}	Operating frequency	–		32.7	32.8	kHz
duty	Output duty cycle(2)	–		40	60	%
$C_{PARA32k}$	Internal parasitic capacitance(2)	Between XIN32 and XOUT32		2		pF
R_F	Internal resistor(2)	Between XIN32 and XOUT32		6		MΩ

Notes:

1. In this table, VIT- refers to the negative-going input threshold voltage of the VBAT POR (see Table 73.42). The operation of the backup section of the device is granted down to the VBAT POR VIT- threshold.
2. Simulation data
3. R_S is the crystal's equivalent series resistor.

Figure 73.37. 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 table below summarizes recommendations for 32.768 kHz crystal selection.

Table 73.46. 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

Table 73.46. Recommended 32.768 kHz Crystal Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
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

73.8.6. Main Crystal Oscillator

Table 73.47. Main Crystal Oscillator Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	–	–	3.0	mA
t_{START}	Start-up time(2)	–	–	10	ms
f_{OSC}	Operating frequency range	–	20	50	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

The Main crystal oscillator supports a Bypass mode. See Input AC Characteristics.

Two sets of crystal characteristics are supported with this oscillator, corresponding to two operating frequency ranges:

- Set 1: Crystal frequency is between 20 and 30 MHz. See Table 73.48.
- Set 2: Crystal frequency is between 30 and 50 MHz. See Table 73.49.

When choosing a crystal, one and only one of these sets of characteristics must be completely satisfied.

Table 73.48. Recommended Crystal Characteristics (Set 1)

Symbol	Parameters	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	20	30	MHz
$C_{CRYSTAL}$	Crystal load capacitance	–	6	12.5	pF
R_S	Equivalent series resistor	–	–	100	Ω
C_M	Motional capacitance	–	1.3	3	fF
C_{SHUNT}	Shunt capacitance	–	–	3	pF
P_{ON}	Drive level	–	300	–	µW

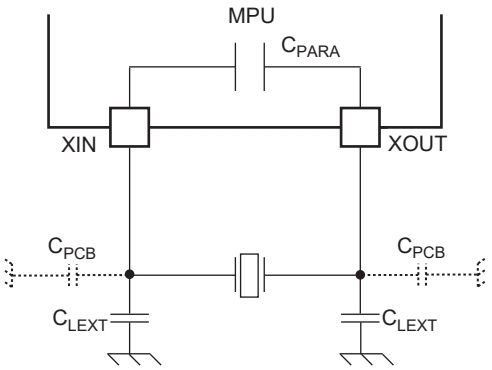
Table 73.49. Recommended Crystal Characteristics (Set 2)

Symbol	Parameters	Conditions	Min	Max	Unit
f_0	Fundamental frequency	–	30	50	MHz
$C_{CRYSTAL}$	Crystal load capacitance	–	6	10	pF
C_{SHUNT}	Shunt capacitance	With R_{S_MAX} = 60Ω	–	3	pF
		With R_{S_MAX} = 80Ω	–	1	pF
R_S	Equivalent series resistor	With C_{SHUNT_MAX} = 1pF	–	80	Ω
		With C_{SHUNT_MAX} = 3pF	–	60	Ω

Table 73.49. Recommended Crystal Characteristics (Set 2) (continued)

Symbol	Parameters	Conditions	Min	Max	Unit
C_M	Motional capacitance	–	1.3	3.2	fF
P_{ON}	Drive level	–	400	–	µW

Figure 73.38. Main Crystal Oscillator Schematic



$C_{LEXT} = 2 \times (C_{CRYSTAL} - C_{PARA} - C_{PCB} / 2)$.

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), $C_{LEXT} = 2 \times (12.5 - 1 - 0.5) = 22$ pF.

73.8.7. 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 leads 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 traces, it is recommended to minimize as much as possible their routing length. These traces must be kept away from noisy switching signals (clock, data, PWM, etc.). A good practice is to shield them with a quiet ground to avoid coupling to neighboring signals.

73.8.8. PLL Characteristics

Table 73.50. PLLA Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.03	1.21	V

Table 73.50. PLLA Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
I_{DDIN33}	Current consumption (VDDIN33)(2)	PLLACK = 1.6 GHz	–	2.0	mA
I_{DDCORE}	Current consumption (VDDCORE)(2)	PLLACK = 1.6 GHz	–	2.5	mA
t_{START}	Start-up time(2)	–	–	50	µs
f_{IN}	Input frequency range	–	20	50	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	800	1600	MHz
f_{PLLACK}	Output frequency range (PLLACK)	–	$f_{COREPLLCK}$ / 2		MHz

Notes:

1. This PLL is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data

Table 73.51. UPLL Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.03	1.21	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	–	–	2.4	mA
I_{DDCORE}	Current consumption (VDDCORE)(2)	–	–	2.8	mA
t_{START}	Start-up time(2)(3)	–	–	150	µs
f_{IN}	Input frequency range(4)(5)	–	20	50	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	960	MHz
f_{OUT}	Output frequency range(6)	–	$f_{COREPLLCK}$ / 2		MHz

Notes:

1. This PLL is powered by an internal dedicated voltage regulator, supplied from VDDIN33, that must be started by software before enabling this PLL. Refer to Clock Generator.
2. Simulation data
3. Covers the start-up time of the PLL and of its dedicated voltage regulator.
4. Only 24 or 48 MHz input frequencies are authorized to support USB-related features and in particular those of the bootloader program in ROM.
5. For optimal settings of UPLL, set the PMC_PLL_ACR as follows:
   PMC_PLL_ACR=0x12023010 for f_{IN}=[20 MHz, 32 MHz]
   PMC_PLL_ACR=0x1B023010 for f_{IN}=[32 MHz, 50 MHz]
6. The post divider is hardwired in a divide-by-2 configuration.

Table 73.52. LVDS PLL Characteristics

Symbol	Parameters	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.03	1.21	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	–	–	2.8	mA
I_{DDCORE}	Current consumption (VDDCORE)(2)	–	–	3.5	mA
t_{START}	Start-up time(2)	–	–	100	µs
f_{IN}	Input frequency range(3)	–	20	50	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	1200	MHz
$f_{LVDSPLLCK}$	Output frequency range (LVDSPLLCK)	–	175	550	MHz
f_{PIXCK}	Pixel clock frequency(4)	–	$f_{COREPLLCK}$ /7		MHz

Notes:

1. This PLL is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data.
3. For optimal settings of LVDS PLL, set the PMC_PLL_ACR as follows:
   PMC_PLL_ACR=0x12023010 for f_{IN}=[20 MHz, 32 MHz]
   PMC_PLL_ACR=0x1B023010 for f_{IN}=[32 MHz, 50 MHz]
4. This is the pixel clock feeding the LCD Controller when using the LVDS Controller.

Table 73.53. AUDIO PLL Characteristics

Symbol	Parameters	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33) (1)	–	3.0	3.60	V
V_{DDCORE}	Supply voltage range (VDDCORE)	–	1.03	1.21	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	–	–	2.8	mA
I_{DDCORE}	Current consumption (VDDCORE)(2)	–	–	3.45	mA
t_{START}	Start-up time(2)	–	–	100	µs
f_{IN}	Input frequency range(3)	–	20	50	MHz
$f_{COREPLLCK}$	COREPLLCK frequency range	–	600	1200	MHz
$f_{AUDIOPLLCK}$	AUDIOPLLCK frequency range	–	–	300	MHz
$f_{AUDIOCLK}$	AUDIOCLK Output frequency range(4)	–	–	50	MHz

Notes:

1. This PLL is powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data
3. For optimal settings of AUDIO PLL, set PMC_PLL_ACR as follows:
   PMC_PLL_ACR=0x12023010 for f_{IN} = [20 MHz, 32 MHz]
   PMC_PLL_ACR=0x1B023010 for f_{IN} = [32 MHz, 50 MHz]
4. AUDIOCLK corresponds to the AUDIOCLK pin.

73.8.9. 12-bit ADC Characteristics

Table 73.54. ADC Power Supply and Voltage Reference Input Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDANA}	Supply voltage range (VDDANA)	–	3.0	3.60	V
I_{DDANA}	Current consumption (VDDANA)(2)	Low speed – $f_S \le 500$ kS/s ADC_ACR.IBCTL = $(00)_2$	–	1.0	mA
		Full speed – $f_S \le 1$ MS/s ADC_ACR.IBCTL = $(01)_2$	–	1.8	mA
$V_{ADVREFP}$	ADVREFP input voltage range(1)	–	2.4	V_{DDANA}	V
$R_{ADVREFP}$	ADVREFP input resistance to ground(2)	ADC on	7.2	12	kΩ
		ADC off	1	–	MΩ
$C_{ADVREFP}$	Required bypass capacitor on ADVREFP	–	1	–	µF

1. The ADVREFN pin must be connected to the PCB ground plane.
2. Simulation data

Table 73.55. ADC Timing Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
f_{CKADC}	ADC clock frequency	Low speed – $f_S \le 500$ kS/s ADC_ACR.IBCTL = $(00)_2$	0.1	10	MHz
		Full speed – $f_S \le 1$ MS/s ADC_ACR.IBCTL = $(01)_2$	0.2	20	MHz
t_{CONV}	ADC conversion time(1)	–	20	–	t_{CKADC}
f_S	Sampling rate(2)	Low speed – $f_S \le 500$ kS/s ADC_ACR.IBCTL = $(00)_2$	–	0.5	MS/s
		Full speed – $f_S \le 1$ MS/s ADC_ACR.IBCTL = $(01)_2$	–	1	MS/s
t_{START}	Start-up time(3)	–	–	5	µs
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.

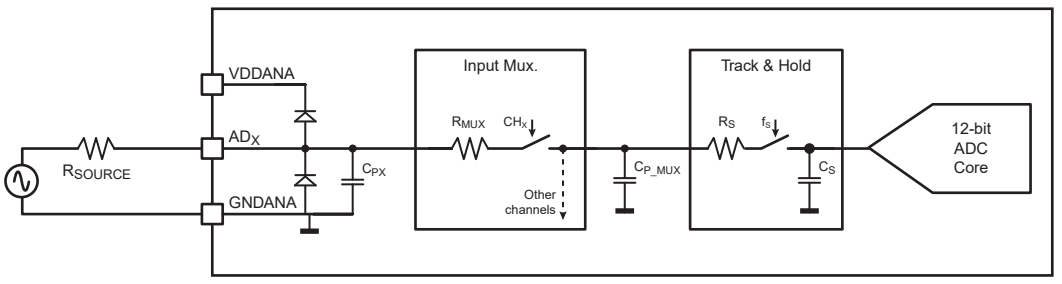
Table 73.56. ADC Analog Input Characteristics

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 mode(2)	ADC_CCR.DIFFx = 1	0.4 x V_{DDANA}	0.6 x * 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

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 Figure 73.40
5. Assuming conversion on one single channel

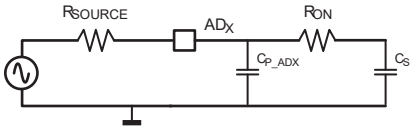
Figure 73.39. Acquisition Path Block Diagram



For tracking time calculation, during the sampling phase of the converter, this acquisition path can be reduced to the equivalent model provided in the following figure, where:

- $R_{ON} = R_{MUX} + R_S$
- $C_{P\_ADX} = C_{PX} + C_{P\_MUX}$

Figure 73.40. Equivalent Model of the Acquisition Path



See Track and Hold Time versus Source Impedance – Sampling Rate for further details on how to use 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 73.57. Static Performance Characteristics

Symbol	Parameter	Conditions	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	–	-4	4	LSB
GE	Gain error	–	-4	4	LSB

Notes:

1. In this table, errors are expressed in LSB where:
   - LSB = $V_{ADVREF} / 2^{12}$ in Single-ended mode (ADC_CCR.DIFFx = 0 and ADC_PDR.PDIFFx = 0)
   - LSB = $V_{ADVREF} / 2^{11}$ in Differential or Pseudo-differential mode (ADC_CCR.DIFFx = 1)
2. Error with respect to the best fit line method

73.8.9.1. Track and Hold Time versus Source Impedance – Sampling Rate

Referring to Figure 73.40, 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 sampling switch series resistor). In case of high output source resistance (with a low-power resistive divider, for example), the tracking time must be increased to ensure full settling of the sampling capacitor voltage. Of course, programming a long tracking time may impact the sampling frequency (f_S). The following formula calculates the minimum tracking time to ensure 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 using the ADC_MR.TRACKTIM and ADC_EMR.TRACKX4 fields. 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: $t_{CONV} = t_{TRACK} + 14 \times t_{CKADC} = 20 \times t_{CKADC}$
- The sampling rate is f_S = 8 MHz / 20 = 400 kS/s
- The maximum allowable source resistance is $R_{SOURCE\_MAX} = (6 \times t_{CKADC}) / (3\ pF \times 8) - 2\ k\Omega$= 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 $t_{CONV} = t_{TRACK} + 14 \times t_{CKADC} = 28 \times t_{CKADC}$
- The sampling rate is f_S = 20 MHz / 28 = 714.3 kS/s
- The maximum allowable source resistance is $R_{SOURCE\_MAX} = (14 \times t_{CKADC}) / (3\ pF \times 8) - 2\ k\Omega$= 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: $t_{CONV} = t_{TRACK} + 14 \times t_{CKADC} = 20 \times t_{CKADC}$
- The sampling rate is f_S = 20 MHz / 20 = 1 MS/s
- The maximum allowable source resistance is $R_{SOURCE\_MAX} = (6 \times t_{CKADC}) / (3\ pF \times 8) - 2\ k\Omega$= 10.5 kΩ

73.8.10. HS USB Transceiver Characteristics

The device complies with all voltage, power, and timing characteristics and specifications set forth in the USB 2.0 Specification. Refer to the USB 2.0 Specification for more information.

Table 73.58. USB 2.0 Transceiver Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range	–	3.0	3.6	V

Table 73.58. USB 2.0 Transceiver Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
I_{DDIN33}	HS transceiver current consumption(1)(2)	Bias circuits enabled only	–	15	mA
		HS(3)	–	20	mA
		LS/FS – 0m cable(3)	–	5	mA
		LS/FS – 5m cable(3)	–	30	mA

1. Simulation data
2. Including 1 mA due to pull-up/pull-down current
3. Additional current consumption for each port enabled

73.8.11. LVDS PHY Characteristics

The SAM9X7 Series complies with the LVDS standard TIA/EIA-644 for protocol and electrical specifications.

Table 73.59. LVDS PHY Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
I_{DDIN33}	Current consumption (VDDIN33)(2)(3)	Pre-emphasis disabled	–	30.0	mA
		Pre-emphasis enabled	–	50.0	mA
t_{START}	Start-up time(2)	–	–	50	µs
V_{OD}	Output differential voltage(2)	–	+/-250	+/-450	mV
V_{CM}	Output common mode voltage	–	1.06	1.44	V
V_{OHS}	Single-ended output level high(2)	–	1.21	1.64	V
V_{OLS}	Single-ended output level low	–	0.91	1.24	V
f_{LVDSCK}	LVDS_CLK1x output frequency(2)	–	57	79	MHz
BR_{LANE}	Bit rate per lane(4)	–	175	550	Mbps

Notes:

1. This LVDS PHY must be powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Simulation data
3. Four-lane data stream, 0.4 Mbps/lane, 100 Ohms load
4. Maximum LVDS PHY operating data rate. From a system point of view, the display resolution is limited to 1280x720.

73.8.12. MIPI D-PHY Characteristics

The SAM9X7 Series complies with the protocol and electrical specifications of the following standards:

- MIPI Alliance Specification for Display Serial Interface (DSI) Version 1.2
- MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2), Version 1.2
- MIPI Alliance Specification for D-PHY, Version 1.2

Table 73.60. MIPI PHY DSI Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
I_{DDIN33}	Current consumption (VDDIN33)(2)	HS mode, 4 lanes	–	30.0	mA
t_{START}	Start-up time(3)	–	–	60	µs

Table 73.60. MIPI PHY DSI Characteristics (continued)

Symbol	Parameter	Conditions	Min	Max	Unit
High-Speed Characteristics					
V_{CMTXDC}	Output common mode voltage(2)	–	150	250	mV
V_{OD}	Output differential voltage(2)	–	140	270	mV
V_{OHHS}	Output high voltage(2)	–	–	360	mV
Z_{OS}	Output impedance(3)	–	40	62.5	Ω
DZ_{OS}	Output impedance mismatch(3)	–	–	10	%
f_{DSICK}	MIPI_CLKx output frequency(3)	–	40	500	MHz
BR_{LANE}	Bit rate per lane(3)	–	80	1000	Mbps
Low-Power Characteristics					
V_{OH}	High-level output(3)	–	1.10	1.30	V
V_{OL}	Low-level output(3)	–	-50	50	mV
Z_{OLP}	Output impedance(3)	–	110	–	Ω
t_{RLP} / t_{FLP}	15% to 85% rise time and fall time(3)	C_L < 70 pF	–	25	ns

Notes:

1. This MIPI D-PHY must be powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.
2. Lane load impedance is 80 to 125 Ω.
3. Simulation data

Table 73.61. MIPI PHY CSI Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDIN33}	Supply voltage range (VDDIN33)(1)	–	3.0	3.60	V
I_{DDIN33}	Current consumption (VDDIN33)	HS mode, 4 lanes	–	15.0	mA
t_{START}	Start-up time(2)	–	–	60	µs
V_{IN}	Recommended input voltage range(2)	On MIPI_CLKx and MIPI_Dx	-50	1350	mV
I_{LEAK}	Input leakage current(2)	On MIPI_CLKx and MIPI_Dx	-10	10	µA
High-Speed Characteristics					
V_{CMRXDC}	Input common mode voltage range(1)	–	70	330	mV
V_{IDTH}	Differential input high voltage threshold(2)	–	–	70	mV
V_{IDTL}	Differential input low voltage threshold(2)	–	-70	–	mV
V_{IHHS}	Input high voltage(2)	–	–	460	mV
V_{ILHS}	Input low voltage(2)	–	-40	–	mV
Z_{ID}	Differential input impedance(2)	–	80	125	Ω
f_{CSICK}	MIPI_CLKx output frequency(2)	–	40	500	MHz
BR_{LANE}	Bit rate per lane(2)	–	80	1000	Mbps
Low-Power Characteristics					
V_{IH}	Input high level(2)	–	880	–	mV
V_{IL}	Input low level(2)	–	–	550	mV
V_{HYST}	Input hysteresis(2)	–	25	–	mV
V_{IHF}	Input high fault threshold(2)	–	450	–	mV
V_{ILF}	Input low fault threshold(2)	–	–	200	mV

1. This MIPI D-PHY must be powered by the 2.5V regulated output of the VDDOUT25 regulator, which is supplied from VDDIN33.

2. Simulation data

73.9. Power Consumption in Active Mode

Table 73.62 and Table 73.64 report active power consumption data measured on a few typical samples of SAM9X7 Series. These data do not provide maximum power consumption specifications.

73.9.1. Processor Power Consumption in Active Mode

Table 73.62 provides the processor power consumption in the following conditions:

- f_{CPU_CLK} = from 200 MHz to 800 MHz
- f_{MCK} = 100 MHz to 266 MHz
- L1 caches enabled
- The Arm926EJ-S core executes a Coremark benchmark from the (internal) SRAM0
- Code compiled with speed optimization
- Peripheral clocks disabled
- Current measured as per the following figure

Figure 73.41. Current Measurement on VDDCORE

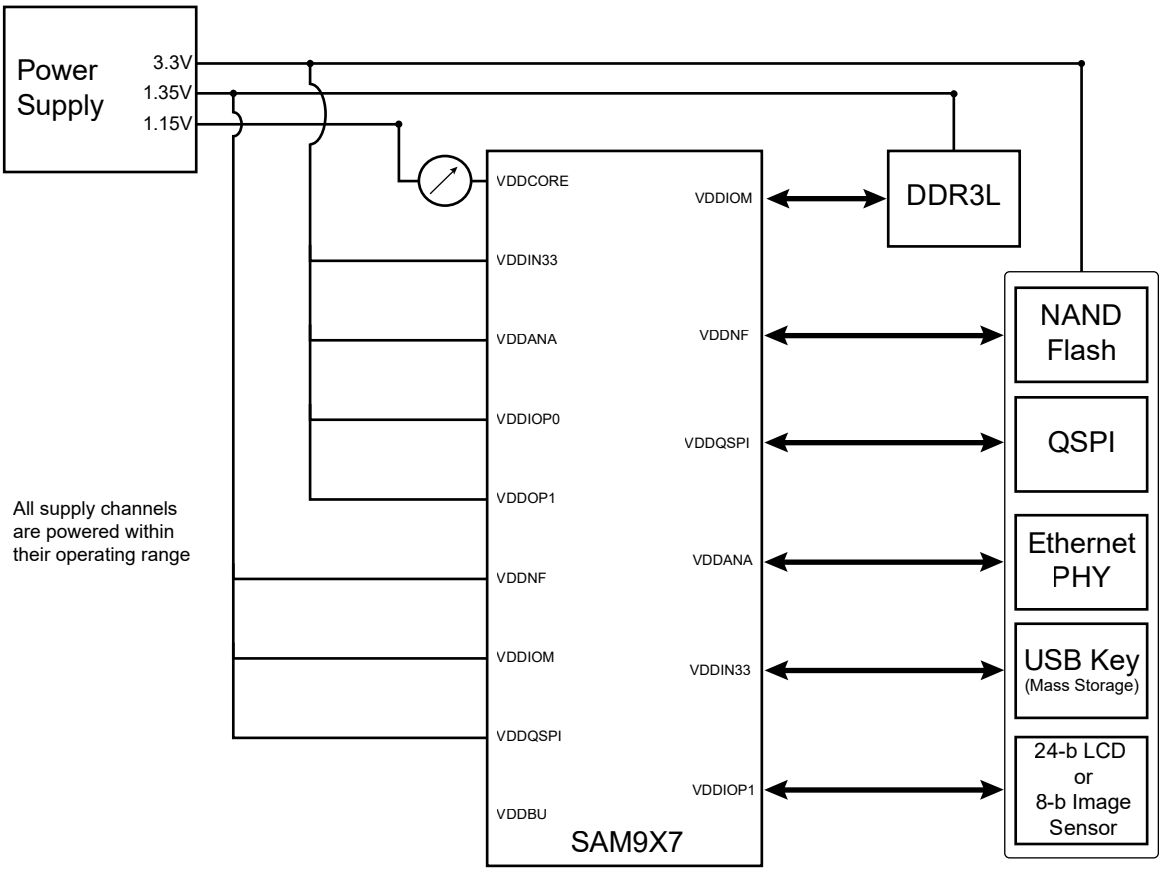


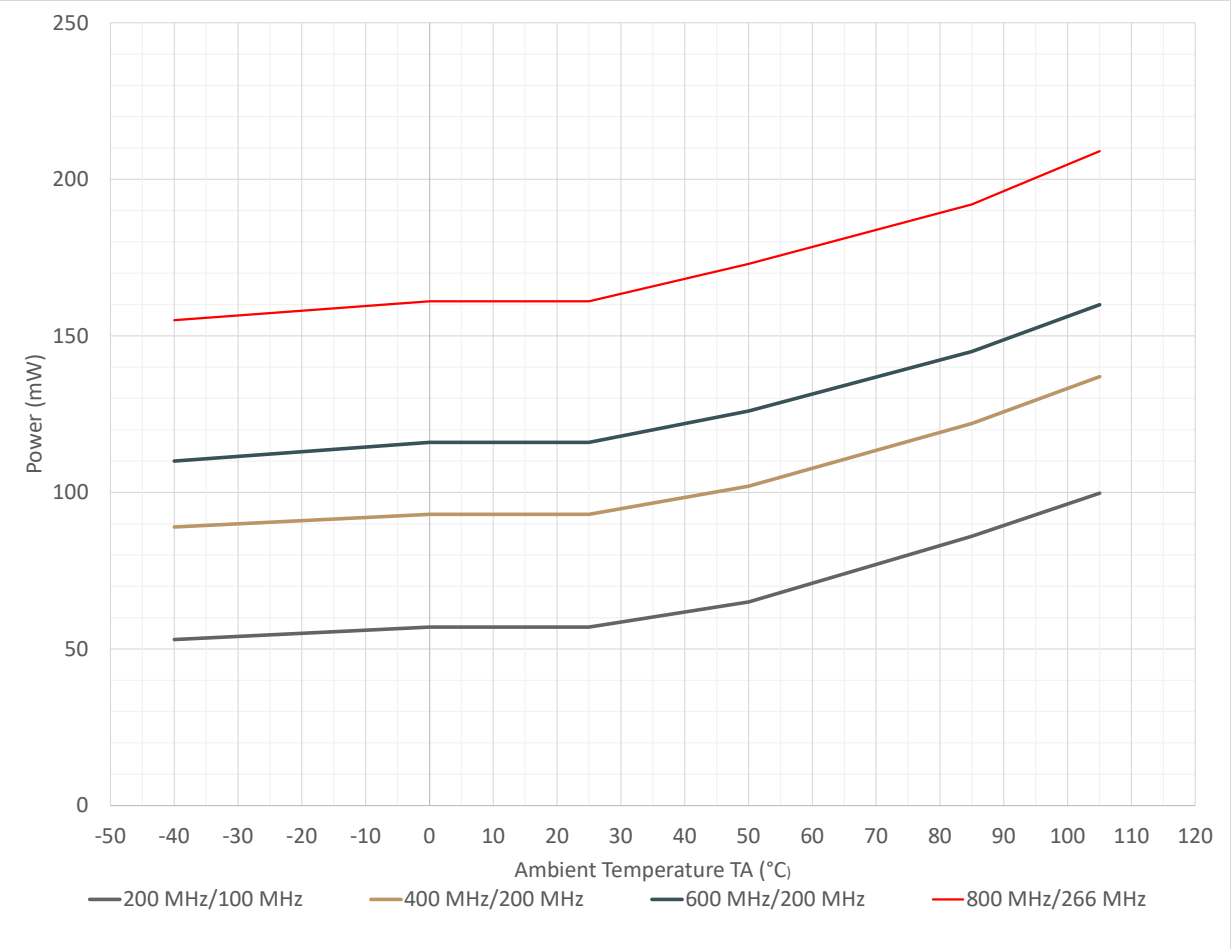
Table 73.62. Processor Power Consumption Running a Coremark Benchmark from SRAM0

f_{CPU_CLK}/f_{MCK}	Power (mW) vs T_A					
	-40°C	0°C	25°C	50°C	85°C	105°C
200 MHz/100 MHz	53	57	57	65	86	99.75

Table 73.62. Processor Power Consumption Running a Coremark Benchmark from SRAM0 (continued)

f_{CPU_CLK}/f_{MCK}	Power (mW) vs T_A					
	-40°C	0°C	25°C	50°C	85°C	105°C
400 MHz/200 MHz	89	93	93	102	122	137
600 MHz/200 MHz	110	116	116	126	145	160
800 MHz/266 MHz	155	161	161	173	192	209

Figure 73.42. Processor Power Consumption Running a Coremark Benchmark from SRAM0



73.9.2. System Power Consumption in Applicative Use Cases

Table 73.64 provides the processor power consumption in the following conditions:

- f_{CPU_CLK} = 800 MHz
- f_{MCK} = 266 MHz
- I & D caches enabled
- External SDRAM memory: DDR3L operating at 266 MHz
- Use cases are run on Linux
- Current consumptions are measured according to Figure 73.43. Note that external component current consumptions are not counted. Either the 24-bit display or the 8-bit image sensor is connected to the SAM9X7 Series device, but not both.

Table 73.63. Use Case Definition

Use Case	Description
1	After Linux boot root prompt
2	Running 2D graphics benchmark with EGT monitor (800x480 DSI panel @ 46 fps) - 22% CPU usage
3	SAM9X70 running as iPerf server
4	Run Bonnie++ on USB mass storage
5	Image sensor video recording with QVGA resolution - 10 fps

Figure 73.43. Current Measurement for Applicative Use Cases

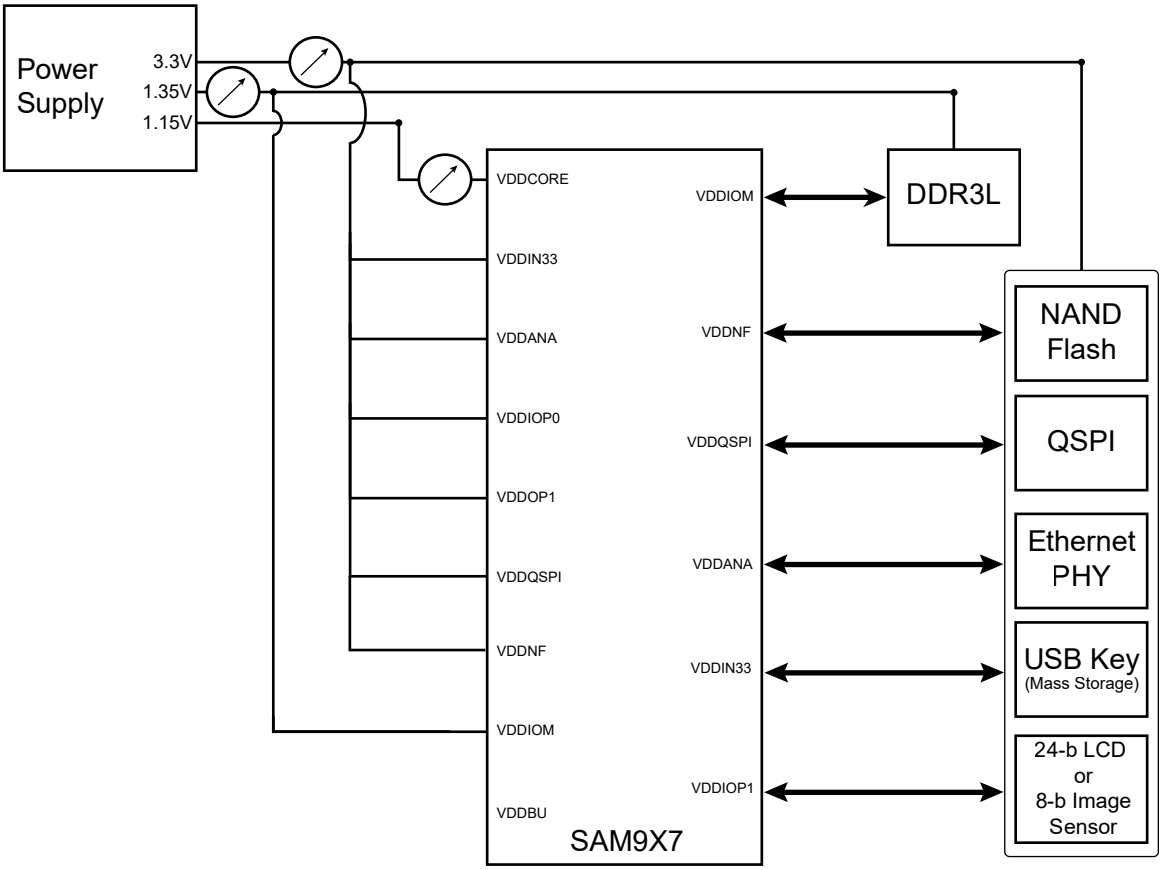


Table 73.64. Power Consumption in Applicative Use Cases at T_A = 25°C

Power Input	Voltage Level (V)	Power Consumption vs Use Case (mW)				
		#1	#2	#3	#4	#5
VDD_3V3	3.3	165	125	165	116	330
VDDCORE	1.15	132	132	167	207	196
VDDIOM	1.35	27	43	95	41	53

73.10. Operation and Power Consumption in Low-Power Modes

The SAM9X7 Series features four low-power modes summarized in Table 73.65. A detailed description of each mode is provided in the following sections.

Table 73.65. Low-Power Modes Summary

		Backup Mode	ULP1	ULP0	Idle
VDDBU power		✓	✓	✓	✓
Other power supply inputs		✗	✓	✓	✓
DDR self-refresh		–	✓	✓	✗
Main crystal oscillator		–	✓	✓	✓
Main RC oscillator		–	✓	✓	✓
MAINCK (main clock)	Source	–	Main RC osc.	User-defined	User-defined
	Status	–	✗	User-defined	✓
MCK (main system bus clock)	Source	–	MAINCK	User-defined (MD_SLCK, MAINCK, etc.)	User-defined
	Status	–	✗	✓	✓
HCLK (CPU clock)		–	✗	✓	✓
CPU state		–	Clocks stopped	WFI	WFI
Mode entry		Use SHDWC to shutdown all power rails except VDDBU	DDR-saved context DDR self-refreshed ULP1 bit	DDR-saved context DDR self-refreshed WFI	WFI
Wake-up	Sources	RTC alarm WKUP0 pin	RTC/RTT alarm USB resume Wake-On-LAN WKUP[13:0]	Any non-masked interrupt	Any non-masked interrupt
	Event or interrupt sampling clock	MD_SLCK	Asynchronous	User-defined (MCK, periph_clk, etc.)	User-defined
MCK at wake-up		MAINCK (defaults to Main RC osc.)	MAINCK (Main RC osc.)	User-defined	User-defined

Note:

1. ✓/✗ means either powered/unpowered for a power source, on/off for an oscillator, active/inactive for a clock signal, or in/out of Self-refresh mode for the external SDRAM.

73.10.1. Backup Mode

73.10.1.1.Operation

The 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 VDDBU is maintained (RTC, GPBR, SHDWC, BSCR). This mode is entered by shutting down all the power rails of the device except VDDBU. It is good practice to use the SHDN output of the Shutdown Controller (SHDWC) for this purpose.

The SAM9X7 Series exits Backup mode when an active wake-up event is triggered by the Shutdown Controller (SHDWC). Upon this wake-up event, the SHDWC automatically toggles its SHDN output back to VDDBU 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 (level transition, configurable debouncing)
- RTC alarm

73.10.1.2.Power Consumption

Backup mode configuration and measurements are defined as follows:

- POR backup on VDDBU enabled

- RTC running
- Force wake-up (FWUP) enabled
- Current measurement as per the following figure

Figure 73.44. Measurement on VDDBU

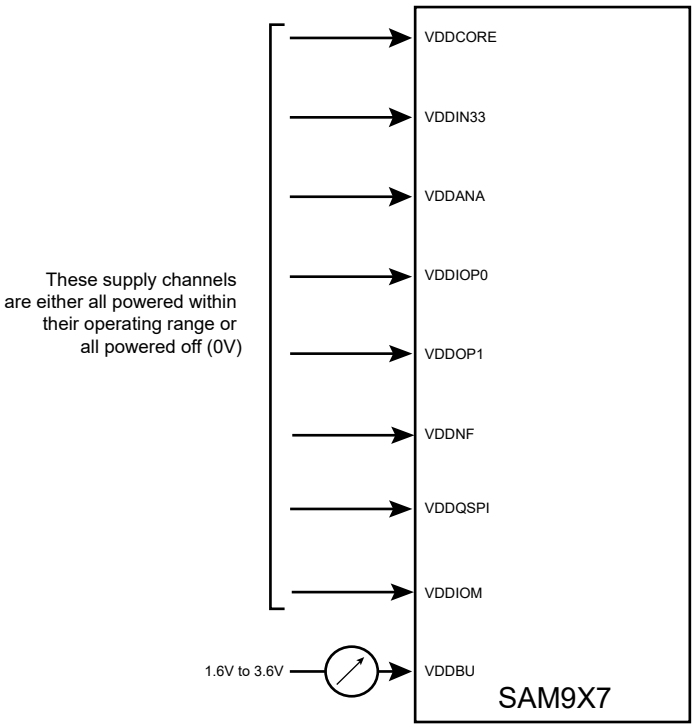
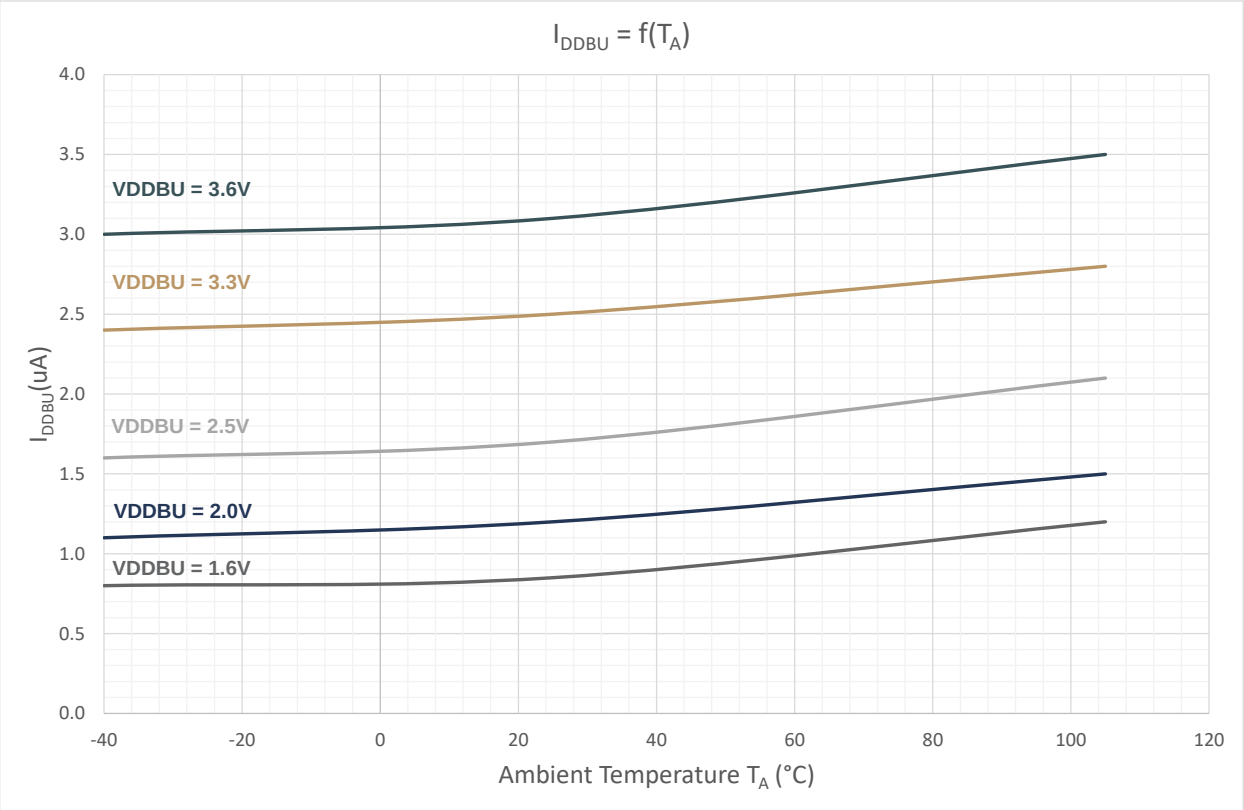


Table 73.66. Current Consumption in Backup Mode at T_A = 105°C

Symbol	Parameter	Conditions	Min	Max	Unit
I_{VDDBU}	Current consumption in Backup mode (VDDBU)	VDDBU = 3.3V	–	5	µA

The following figure shows the average Backup mode current consumption measured on a few typical devices.

Figure 73.45. Typical Current Consumption in Backup Mode



73.10.2. ULP0, ULP1 and Idle Modes

In these three low-power modes, the SAM9X7 Series power supplies are all applied within their operating range. The power reduction is achieved by reducing the frequency or even stopping the clock signals of the processor and/or its peripherals.

- In Idle mode, only the processor clock is affected when in ULP0 and ULP1 modes; the clocks feeding both the processor and its peripherals are slowed down.
- In ULP0 or ULP1 mode, the SAM9X7 Series is placed in Retention mode and is able to resume on wake- up events (any interrupt or hardware event). This mode is a combination of the Wait for Interrupt mode of the Arm core and the system clocks frequency reduction or shut-off.

A detailed description of each mode is provided in the following sections and is followed by a power consumption section dedicated to those modes.

ULP0 Operation

The ULP0 mode maintains very low frequency clocks (MCK, CPU_CLK) in the system to wake up on any interrupt. The 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 SRAM0.

1. Set the DDR to Self-refresh mode.
2. Set the interrupts to wake up the system.
3. Disable all peripheral clocks.
4. Set the I/Os to an appropriate state and disable the USB transceivers.

5. Switch the system clock to Slow Clock.
6. Set the SRAM memories to Light Sleep mode in SFR_LS, except LS6 (internal SRAM).
7. Disable the PLLs, the main oscillator and the 12 MHz RC oscillator.
8. Enter the Wait for Interrupt mode and disable the CPU_CLK clock in PMC_SCDR.

Wake-up from ULP0 mode is triggered by any enabled interrupt. When resuming, the software reconfigures the system (oscillator, PLL, etc.) in the same state as before WFI. Disable Light Sleep mode for SRAM memories in the SFR_LS register prior to returning to full speed operation.

ULP1 Operation

Unlike the ULP0 mode, all the clocks are off in the ULP1 mode, and the number of wake-up sources is limited to the list below:

- WKUP[13:0] pins (level transition, configurable debouncing)
- RTC/RTT alarm
- USB Resume from Suspend mode
- Wake-On-LAN events from EMAC peripherals

The sequence to enter ULP1 mode is detailed below. The code used to enter this mode must be executed out of the internal SRAM0.

1. Set the DDR to Self-refresh mode.
2. Set the events to enable a system wake-up.
3. Disable all peripheral clocks.
4. Set the I/Os to an appropriate state and disable the USB transceivers.
5. Switch the system clock to the 12 MHz RC oscillator.
6. Set the SRAM memories to Light Sleep mode in SFR_LS, except LS6 (internal SRAM).
7. Disable the PLLs and the main oscillator.
8. Enter ULP1 mode by setting the ULP1 bit in CKGR_MOR and wait for the PMC_SR.MCKRDY bit to be set.

When resuming, the software reconfigures the system (oscillator, PLL, etc.) in the same state as before entering ULP1. In particular, the SRAM memories Light Sleep mode must be disabled as soon as possible in the wake-up process.

Idle Mode Operation

The purpose of Idle mode is to optimize power consumption of the device 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 Wait for Interrupt (WFI) instruction.

The processor can be awakened from an interrupt. The system resumes where it was before entering WFI mode.

73.10.3. Power Consumption in ULP0, ULP1 and Idle Modes

- Figure 73.47, Figure 73.48, Table 73.67 and Table 73.68 report the SAM9X7 Series average power consumption in Idle, ULP0 and ULP1 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.

- There is no consumption on the device I/Os. To reach the best possible power consumption figures, it is of prime importance 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 transceivers are disabled.
- All peripheral clocks are disabled.
- Current measurement as per the following figure.

Figure 73.46. Current Measurement in ULP0, ULP1 and Idle Modes

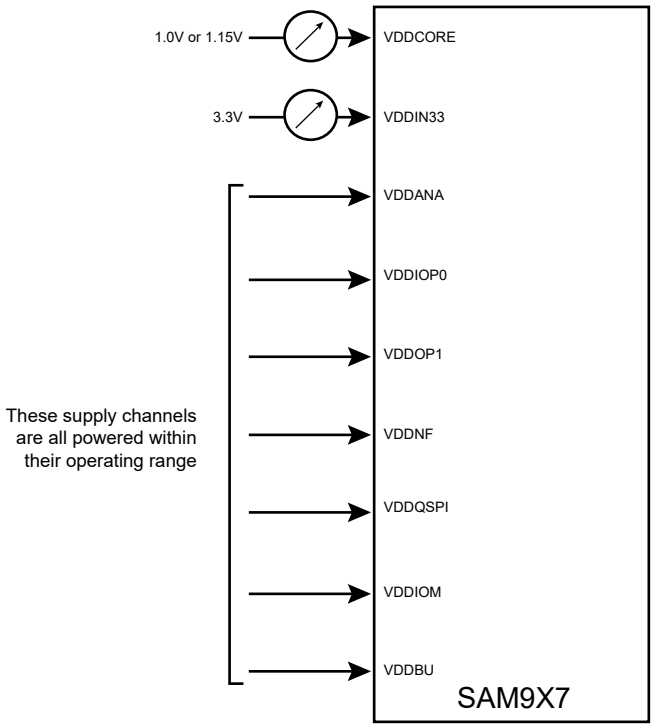


Table 73.67. Typical Power Consumption in ULP0, ULP1 and Idle Modes

Low-Power Mode (mW)	P_{VDD3v3}	$P_{VDDCORE}$	P_{VDD3v3}	$P_{VDDCORE}$	P_{VDD3v3}	$P_{VDDCORE}$	P_{VDD3v3}	$P_{VDDCORE}$	P_{VDD3v3}	$P_{VDDCORE}$	P_{VDD3v3}	$P_{VDDCORE}$	Wake-up Time (µs)
T_A (°C)	-40		25		50		70		85		105		
Idle (MCK @ 266 MHz)	0.66	37.95	0.83	44.85	0.86	51.75	0.89	59.80	0.89	70.15	0.92	86.25	10x t_{CPU_CLK}
ULP1	0.66	1.58	0.83	4.73	0.86	8.93	0.89	15.75	0.89	23.10	0.92	33.60	1515
ULP1 @ 1.0V	0.66	1.00	0.83	3.00	0.86	6.00	0.89	10.00	0.89	14.50	0.92	23.00	
ULP0 @ 32 kHz	0.66	1.26	0.83	7.04	0.86	12.50	0.89	19.11	0.89	25.73	0.92	36.96	5625
ULP0 @ 187.5 kHz	0.99	1.37	1.12	7.09	1.16	12.60	1.19	19.22	1.19	25.73	1.22	36.96	960
ULP0 @ 750 kHz	0.99	1.47	1.12	7.14	1.16	12.71	1.19	19.22	1.19	25.73	1.22	36.96	240
ULP0 @ 12 MHz	0.99	2.52	1.12	8.19	1.16	14.18	1.19	20.16	1.19	26.67	1.22	37.70	15

The following figures plot the power consumption of the SAM9X7 Series in ULP0, ULP1 and Idle modes. These figures account for the power consumption in VDDCORE and VDDIN33: Power (mW) = I_{DDIN33} x 3.3V + I_{DDCORE} x 1.15V, with I_{DDIN33} and + I_{DDCORE} as measured in the above table.

Figure 73.47. Typical Power Consumption in Idle Mode

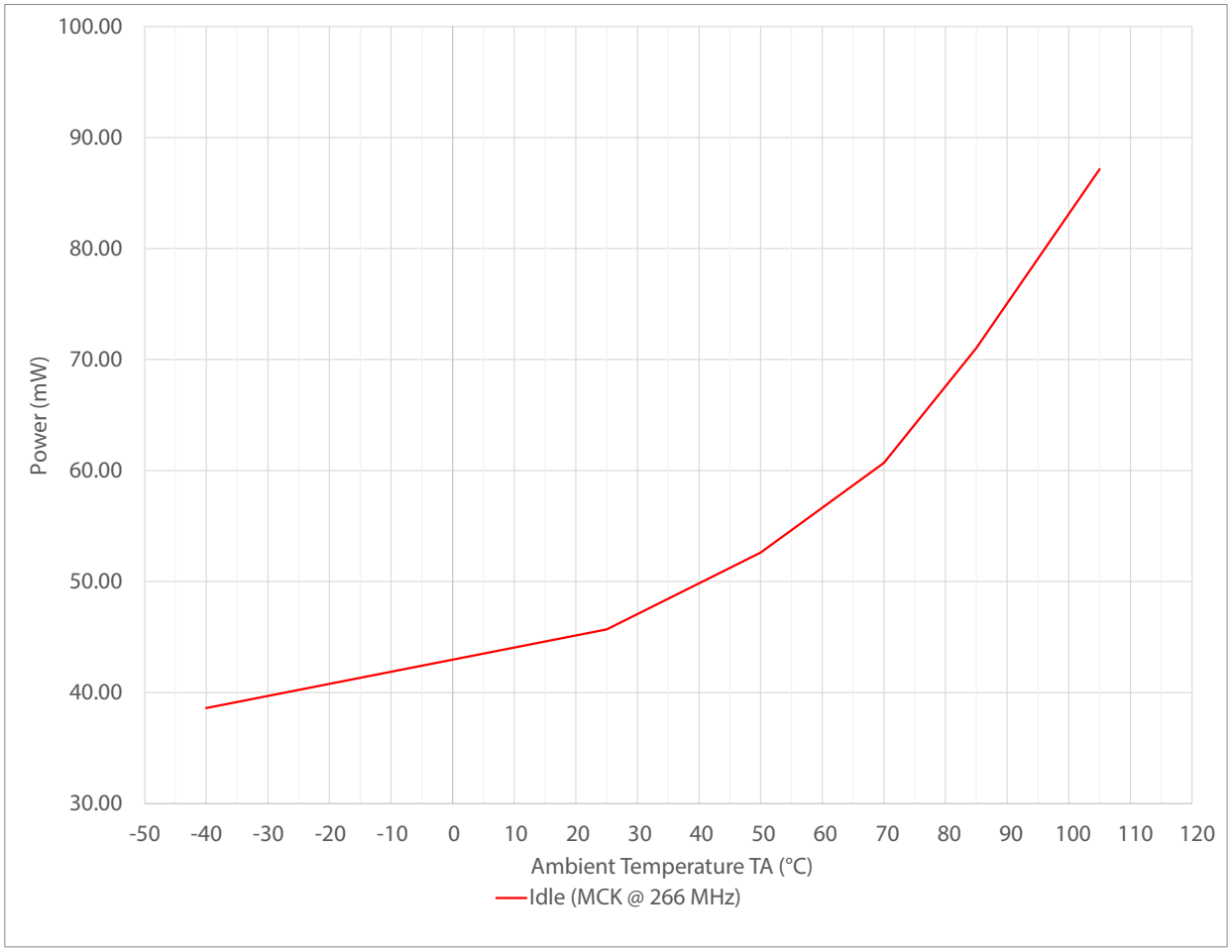
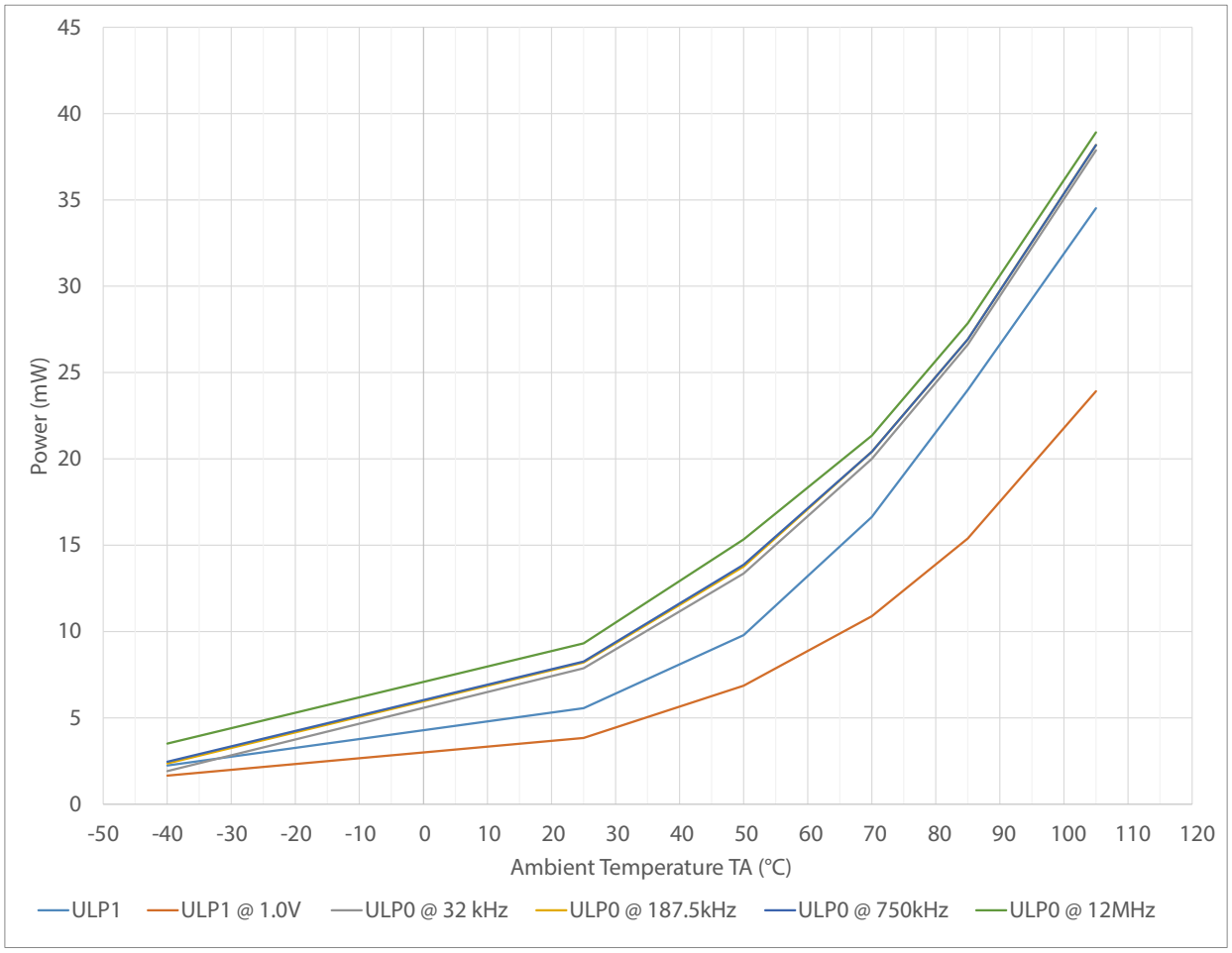


Figure 73.48. Typical Power Consumption in ULP0 and ULP1 Modes



The following table shows the specification for ULP1 mode power consumption on VDDCORE and VDDIN33.

Table 73.68. Power Consumption in ULP1 Mode

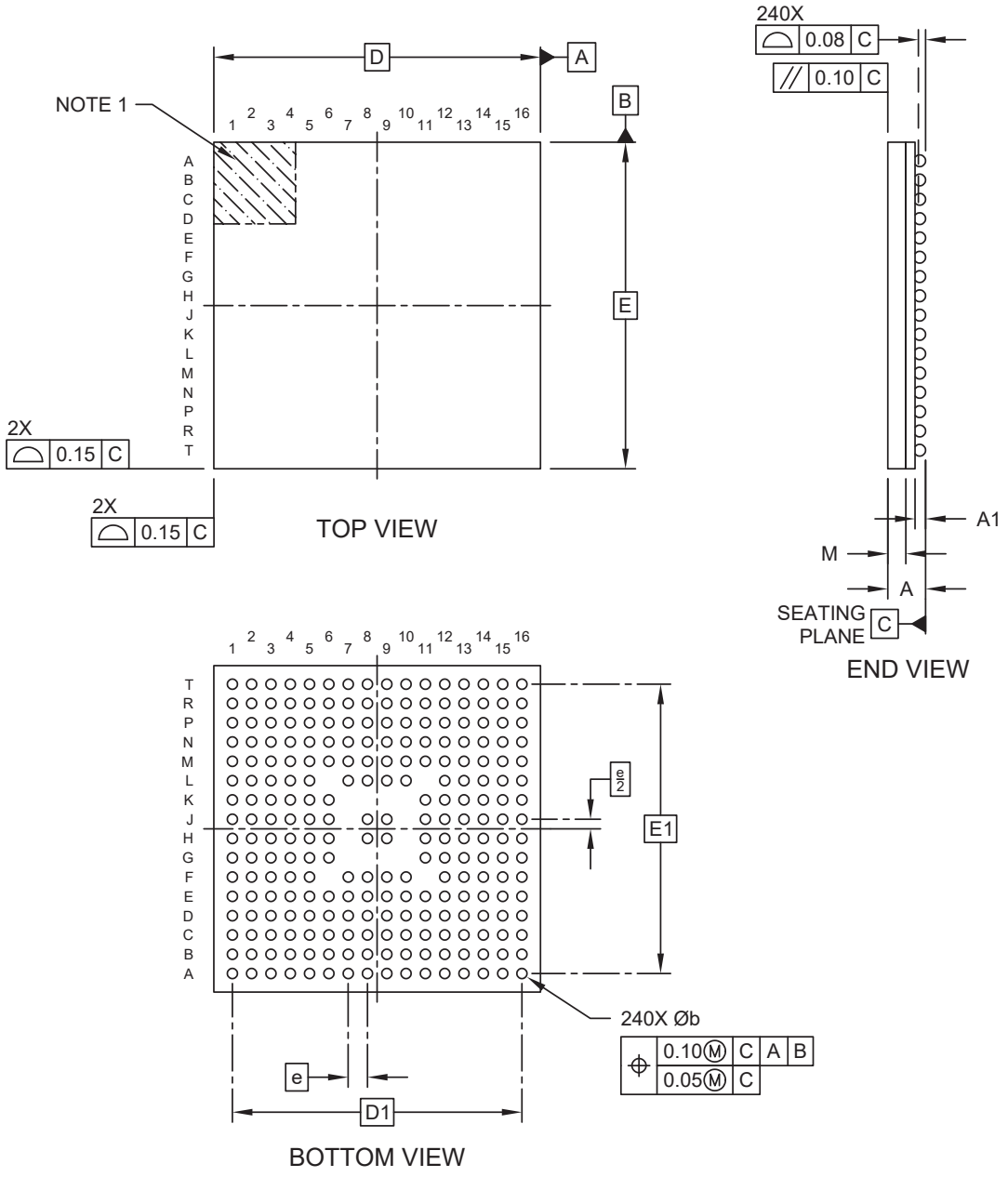
Symbol	Parameter	Conditions	Min	Max	Unit
$P_{VDDCORE}$	Power consumption in ULP1 mode (VDDCORE)	VDDCORE = 1.0V, T_A = 105°C	–	50	mW
	Power consumption in ULP1 mode (VDDCORE)	VDDCORE = 1.0V, T_A = 85°C	–	30	mW
$P_{VDDIN33}$	Power consumption in ULP1 mode (VDDIN33)	VDDIN33 = 3.3V	–	1	mW

74. Mechanical Characteristics

74.1. 240-Ball TFBGA Mechanical Characteristics

240-Ball Thin Fine-Pitch Ball Grid Array (4PB) - 11x11x1.217 mm Body [TFBGA]

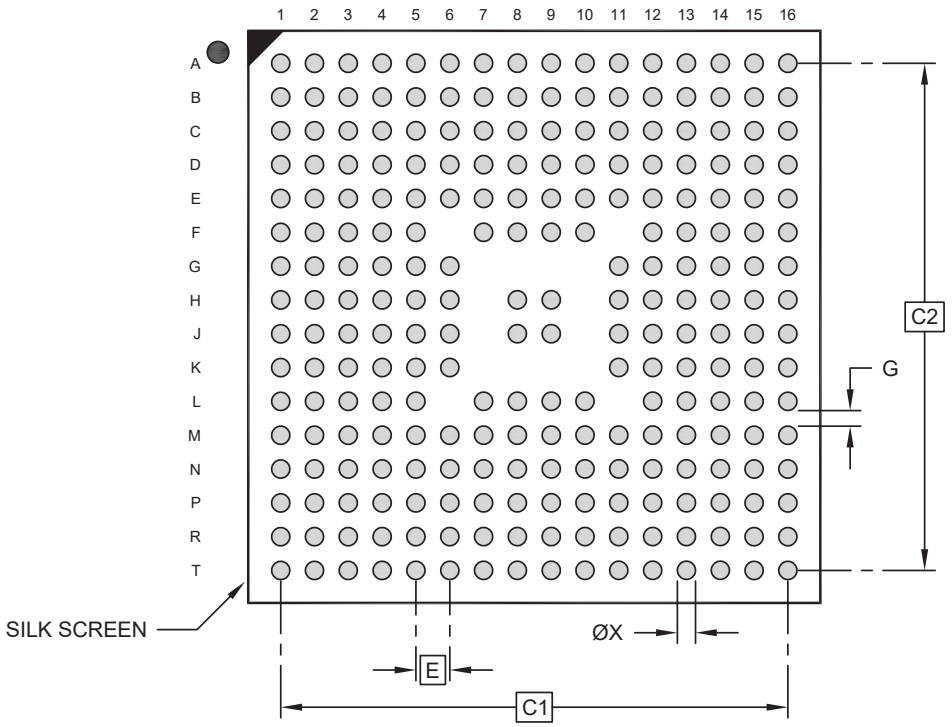
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-21532 Rev C Sheet 1 of 2

240-Ball Thin Fine-Pitch Ball Grid Array (4PB) - 11x11x1.217 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		9.75 BSC	
Contact Pad Spacing	C2		9.75 BSC	
Contact Pad Diameter	X			0.35
Contact Pad to Contact Pad	G	0.30		

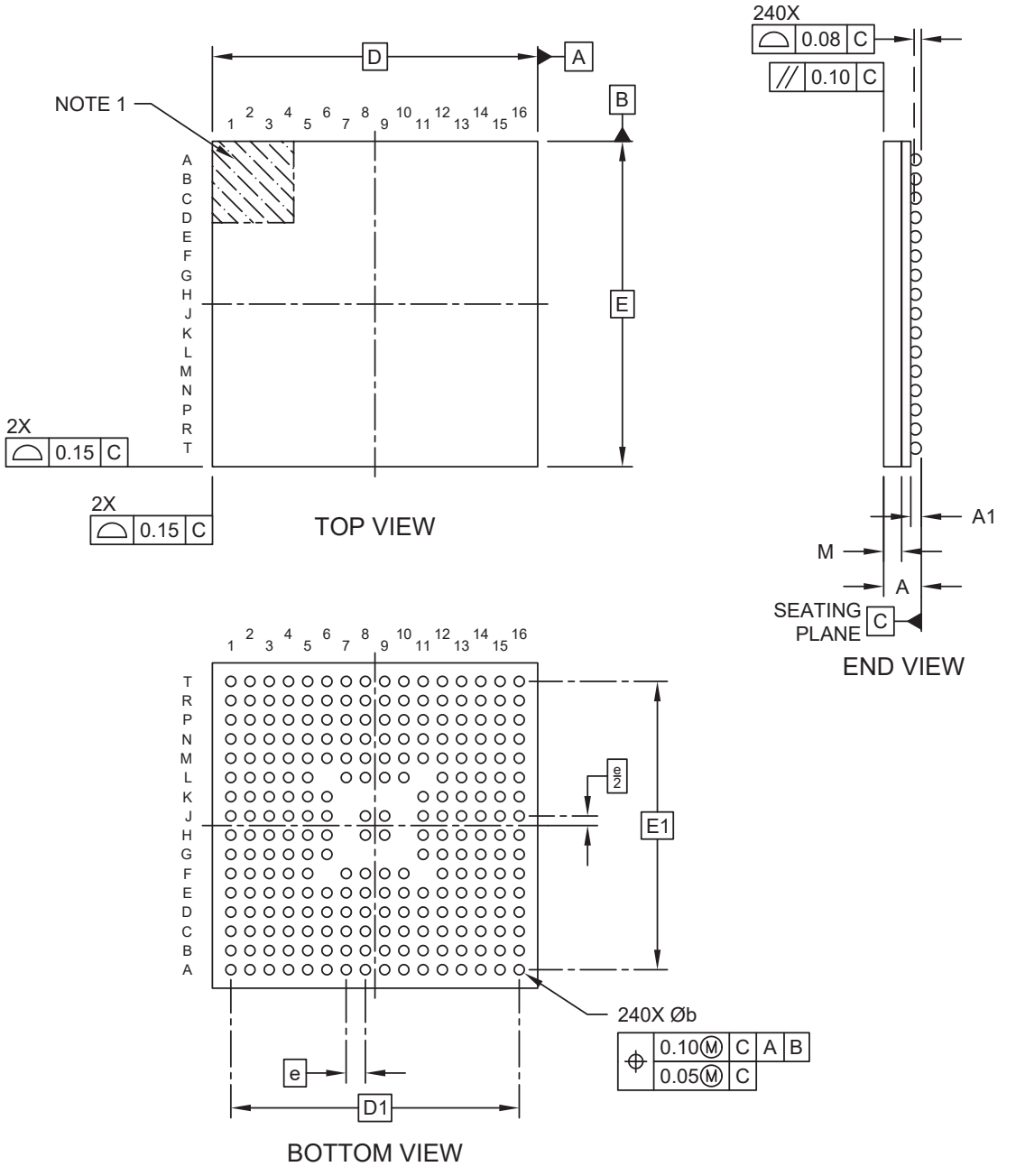
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-23532 Rev C

240-Ball Thin Fine-Pitch Ball Grid Array (4PB) - 11x11x1.217 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 C04-21532 Rev C Sheet 1 of 2

Table 74.1. 240-ball TFBGA Package Characteristics

Moisture Sensitivity Level	3

Table 74.2. Device and 240-ball TFBGA Package Weight

250	mg

Table 74.3. Package Reference

J-STD-609 Classification	e8

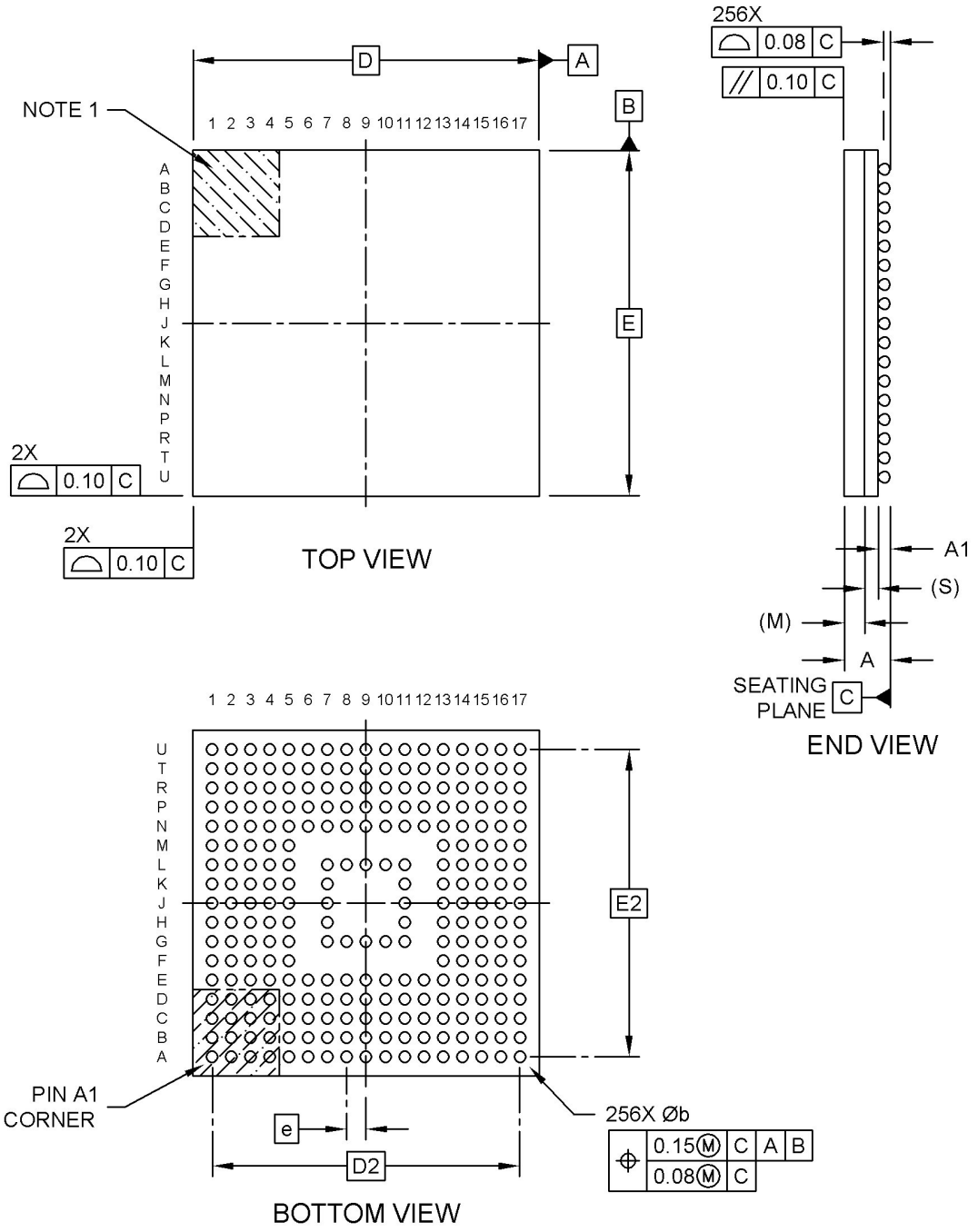
Table 74.4. 240-ball TFBGA Package Information

Ball Land	0.45 mm
Nominal Ball Diameter	0.35 mm
Solder Mask Opening	0.250 mm
Solder Mask Definition	SMD
Solder	LF35

74.2. 256-Ball TFBGA Mechanical Characteristics

256-Ball Thin Fine-Pitch Ball Grid Array Package (6GW) - 9x9x1.2 mm Body [TFBGA]

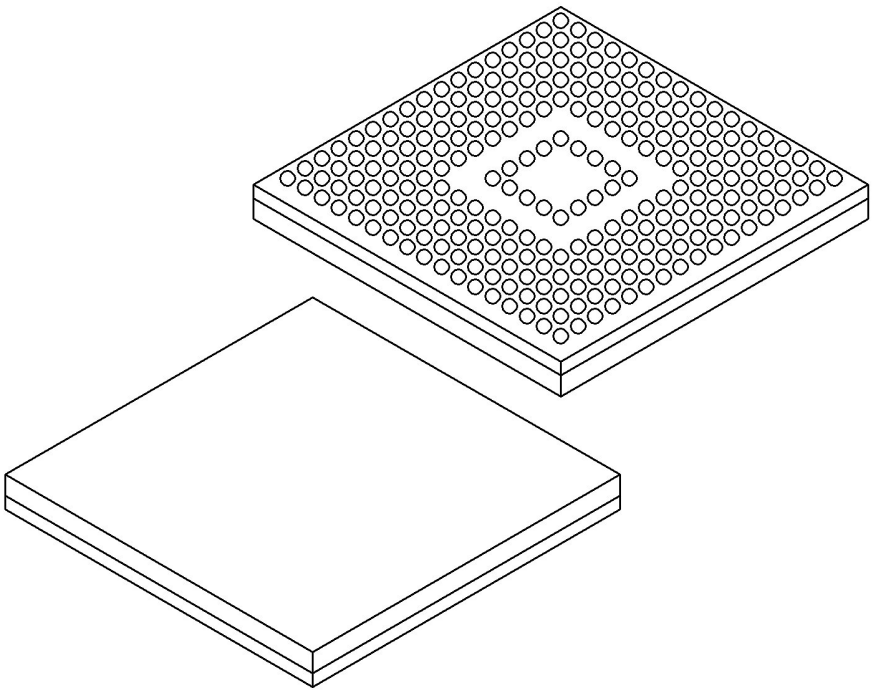
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-622 Rev A Sheet 1 of 2

256-Ball Thin Fine-Pitch Ball Grid Array Package (6GW) - 9x9x1.2 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		256	
Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Ball Height	A1	0.16	0.21	0.26
Mold Thickness	M		0.53 REF	
Substrate Thickness	S		0.35 REF	
Overall Length	D		9.00 BSC	
Ball Array Length	D2		8.00 BSC	
Overall Width	E		9.00 BSC	
Ball Array Width	E2		8.00 BSC	
Ball Diameter	b	0.27	0.32	0.37

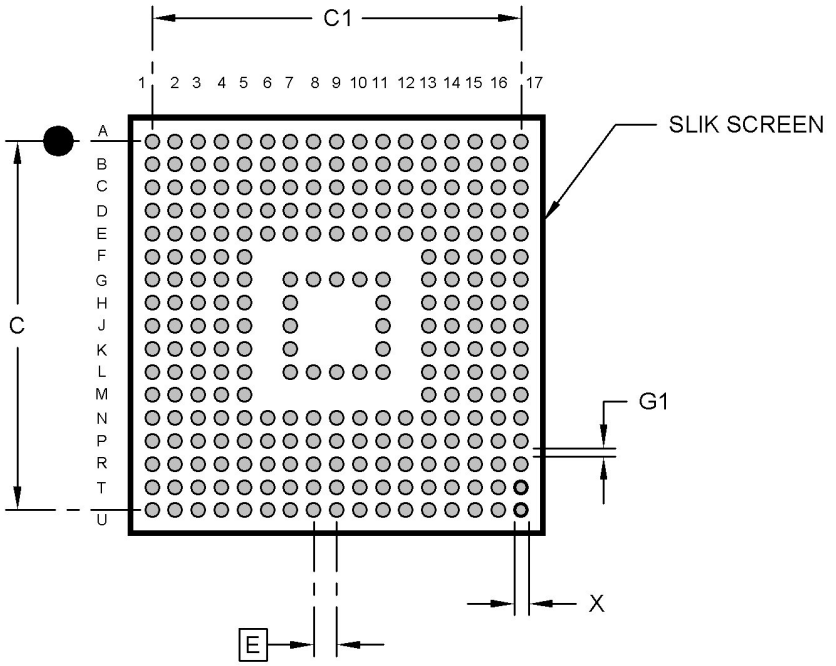
Notes:

1. Pin 1 visual index feature may vary but must be located within the hatched area.
2. Package is saw singulated.
3. 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 C04-622 Rev A Sheet 2 of 2

256-Ball Thin Fine-Pitch Ball Grid Array Package (6GW) - 9x9x1.2 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		0.50 BSC	
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Diameter (X256)	X			0.25
Contact Pad to Contact Pad	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2622 Rev A

Table 74.5. 256-ball TFBGA Package Characteristics

Moisture sensitivity level	MSL3

Table 74.6. Device and 256-ball TFBGA Package Weight

182	mg

Table 74.7. 256-ball TFBGA Package Reference

JEDEC drawing reference	N/A
J-STD-609 classification	e8

Table 74.8. 256-ball TFBGA Package Information

Ball land	0.400 mm
Nominal ball diameter	0.320 mm
Solder mask opening	0.275 mm
Solder mask definition	Solder Mask Defined (SMD)
Solder	LF35

Revision History

75. Revision History

75.1. DS60001813D - 03/2025

Section	Changes
General	Added content related to: • SAM9X75(T)-V/4PB(-SLx)VAO Automotive grade device • SAM9X72 and SAM9X70 fine pitch package devices
Features	Added Qualification feature
Ordering Information	Updated ordering codes
Product Identification System	Updated ordering code description
Security and Cryptography Subsystem	Updated Cryptography Subsystem Keybus
Flexible Serial Communication Controller (FLEXCOM)	FLEX_US_FIDI (LON_MODE), FLEX_US_LONB1TX, FLEX_US_LONB1RX, FLEX_US_IDTTX, FLEX_US_IDTRX: added note in field description
Quad Serial Peripheral Interface (QSPI)	Added End of Frame Procedure
Secure Digital MultiMedia Card Controller (SDMMC)	SDMMC_PVRx: corrected the number of register instances from 4 to 3
USB Device High Speed Port (UDPHS)	Updated Transfer Without DMA
Electrical Characteristics	Added ESD Ratings section, Figure 73.46 Updated Table 73.4, Table 73.6, Table 73.7, Table 73.50, Table 73.51, Table 73.52, Table 73.53, , Table 73.62, Table 73.64, Table 73.68 Updated Figure 73.13, Figure 73.42, Figure 73.44, Figure 73.47, Figure 73.48

75.2. DS60001813C - 12/2024

Section	Changes
General	• Added content related to SAM9X7 Series fine pitch package devices (SAM9X75(T)-V/6GW) • Added content related to Software License Level (SLx) • Data sheet format reworked
Ordering Information	Updated ordering codes
Product Identification System	Updated ordering codes
Boot Strategies	Added Software Considerations Updated Hardware Considerations
Debug Unit (DBGU)	Chip Identifier: updated reset value information
LCD Controller (LCDC)	Updated Embedded Characteristics, Vertical Scaler, Active Mode Output Pin Assignment
Low Voltage Differential Signaling Controller (LVDSC)	LVDSC_WPSR: corrected WPVSRC field size
CSI-2 Demultiplexer Controller (CSI2DC)	YUV 420 8-bit Legacy Mode: updated Figure 49.6; added Note YUV 420 8-bit Mode: updated Table 49.12 and Table 49.13
Image Sensor Controller (ISC)	Updated Defective Pixel Correction (DPC), Green Disparity Correction (GDC), White Balance (WB) Module Black Level Correction (BLC) Rounding, Limiting and Packing (RLP) Module: updated DATY10 table row ISC_INTEN, ISC_INTDIS, ISC_INTMASK, ISC_INTSR: removed WPE

DS60001813C - 12/2024 (continued)	
Section	**Changes**
Inter-IC Sound Multi-Channel Controller (I2SMCC)	Updated Figure 53.7
Flexible Serial Communication Controller (FLEXCOM)	Updated Baud Rate Generator, Baud Rate in Synchronous Mode, FIFO Overflow/Underflow Error, FIFO Overflow/Underflow Error, FIFO Overflow/Underflow Error Updated TXFPTEF and RXFPTEF definitions in FLEX_US_FESR, FLEX_SPI_SR, FLEX_TWI_FSR FLEX_TWI_SMBTR: added detail on WPEN FLEX_SPI_MR: added LSBHALF
Quad Serial Peripheral Interface (QSPI)	Embedded Characteristics: added supported standards Memory Registers/Commands Access: added note about the RBSYERR flag Added Peripheral Bus Access Errors Updated HyperFlash Mode, Twin-Quad Mode Instruction Frame Transmission Examples: corrected address range used in examples QSPI_IFR: updated PROTTYP description Updated Initialization Procedure, Figure 64.23. Added Figure 64.14, Figure 64.16, Figure 64.18, Figure 64.20, Figure 64.22
Analog-to-Digital Controller (ADC)	Updated Table 72.4
Electrical Characteristics	Updated Table 73.4 Added Power Consumption in Active Mode Table 73.9, Table 73.10: updated Note

75.3. DS60001813B - 02/2024

Section	Changes
Reference Document	New section
Configuration Summary	Updated Table 5.1
Block Diagram	Updated Figure 6.1
Package and Pinout	Updated Table 8.1
DMA Controller (XDMAC)	Updated Figure 16.5
Overview	Updated Power Supplies
Real-Time Clock (RTC)	All occurrences of Persian mode deleted Updated Waveform Generation, Figure 33.8, RTC Accurate Clock Calibration RTC_MR: index 1 now 'reserved'; UTC description updated RTC_SR: updated reset value
Power Management Controller (PMC)	Throughout: changed "FSTP" to "WIP" Updated Main Crystal Oscillator Failure Detection
External Bus Interface (EBI)	Added whole chapter
DDR-SDRAM Controller (MPDDRC)	DDR3-SDRAM/DDR3L-SDRAM Initialization: corrected DDR3(L)-SDRAM value
OTP Memory Controller (OTPC)	Updated Power Management
LCD Controller (LCDC)	Updated Embedded Characteristics, Functional Description LCDC_LCDISR: updated WP description
Display Serial Interface (DSI)	Offsets 0x000000E0–0x000000E8 and 0x000000F8 now 'reserved' DSI_INT_FORCE0, DSI_INT_FORCE1: access now Write-only
CSI-2 Demultiplexer Controller (CSI2DC)	CSI2DC_SSPISR, CSI2DC_GSPISR, CSI2DC_GLPISR, CSI2DC_IDSISR, CSI2DC_DPISR, CSI2DC_VPISR: updated bit descriptions
Image Sensor Controller (ISC)	Added section Scaler Function ISC_INTEN, ISC_INTDIS, ISC_INTMASK, ISC_INTSR: added WPE

DS60001813B - 02/2024 (continued)	
Section	**Changes**
Inter-IC Sound Multi-Channel Controller (I2SMCC)	Updated Common Registers TDM Reception and Transmission Sequence: added note I2SMCC_MRA: updated TDMFS description I2SMCC_MRB: updated I2SLINESIZE description I2SMCC_ISRA: updated RXLRDYx and TXRRDYx descriptions
Synchronous Serial Controller (SSC)	Updated Embedded Characteristics
Ethernet MAC (GMAC)	Updated Embedded Characteristics, GMAC Connections to PHY in Different Modes, Media Access Controller, Jumbo Frames, PHY Interface GMAC_EMAC_SCR2_RATE_LIMIT1: added register at offset 0x1B44 GMAC_EMAC_DCFGR: FBLDO now at index [4:0] GMAC_RJFML: updated reset value GMAC_ISR, GMAC_IER, GMAC_IDR, GMAC_IMR: corrected offsets of TXLCK and RXLCK GMAC_DCFGR: corrected offset of CRCERRREP GMAC_RX_WATERMARK, GMAC_EMAC_RX_WATERMARK, GMAC_TQSA, GMAC_EMAC_TQSA, GMAC_SCR2_RATE_STATUS, GMAC_EMAC_TSCTL, GMAC_EMAC_TQBWRL, GMAC_EMAC_ENST_CR, GMAC_EMAC_SCR2_RATE_STATUS, GMAC_NCFGR, GMAC_EMAC_NCFGR: updated bit descriptions GMAC_FRER_STAT_Bx, GMAC_EMAC_FRER_STAT1_B, GMAC_EMAC_FRER_STAT2_B: bits 25:24 now 'reserved' GMAC_SCR2_RATE_LIMITx, GMAC_EMAC_ENST_ON, GMAC_EMAC_ENST_START, GMAC_EMAC_ENST_OFF, GMAC_FRER_CTRL_Ax, GMAC_FRER_CTRL_Bx, GMAC_FRER_STAT_Ax, GMAC_FRER_STAT_Bx: modified index range of registers
Flexible Serial Communication Controller (FLEXCOM)	Updated I/O Lines Description, Baud Rate in Synchronous Mode FLEX_TWI_SR (DEFAULT_MODE), FLEX_TWI_SR (FIFO_ENABLED): index 17 now 'reserved'
Timer Counter (TC)	Updated Block Diagram TC_BMR: updated TCxXCxS descriptions
Electrical Characteristics	Updated Table 73.34, Table 73.16, Digital Peripheral Characteristics System Power Consumption in Applicative Use Cases: updated Table 73.64 Operation and Power Consumption in Low-Power Modes: updated Table 73.65
Ordering Information	Updated ordering codes

75.4. DS60001813A - 03/2023

Changes
First issue.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-0790-5

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 INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

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 products are 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.

Product Page Links

SAM9X70, SAM9X72, SAM9X75