FLC
TECHNOLOGY GROUP

FLC Memory Expander

Technical Reference Manual

April 1, 2025

Document Number XXXXXX

Revision 1.66

Preface

Note for Readers

This document is under development. More details are being added.

About the Manual

FLC Technology Group Memory Expander Hardware Technical Reference Manual. This manual contains documentation for the Memory Expander, the programmer’s model, instruction set, registers, memory map, debug support and many other features.

Documentation Dependencies

The technical reference manual details about the FLC memory expander alongside its dependencies on third party IPs from Cadence, Synopsys and RISC-V to align the functionality.

Audience

This manual is intended to help system designers, system integrators, verification engineers, and software programmers who are using Memory Expander.

Revision History

Version	Date	Comments
V 1.66	April 2025	Document cleanup
V 1.65	February 2025	Updated Registers
V 1.64	12/17/2024	Add 3.19.4
V 1.63	11/23/2024	Tables 1 and 42 Updated Add 2.3.3
V 1.62	10/17/2024	3.18 Updated
V 1.61	10/10/2024	Rearrange Chapters
V 1.60	September 2024	
V 1.59	August 2024	Added Sections
V 1.58	June 2024	Review Draft version
V 1.57	May 2024	Updated previous version
Initial Version	April 2024	Creating the initial draft

Table of Contents

List of Figures

List of Tables

1.0 Introduction

This chapter describes the processor, its application domain and its significant features.

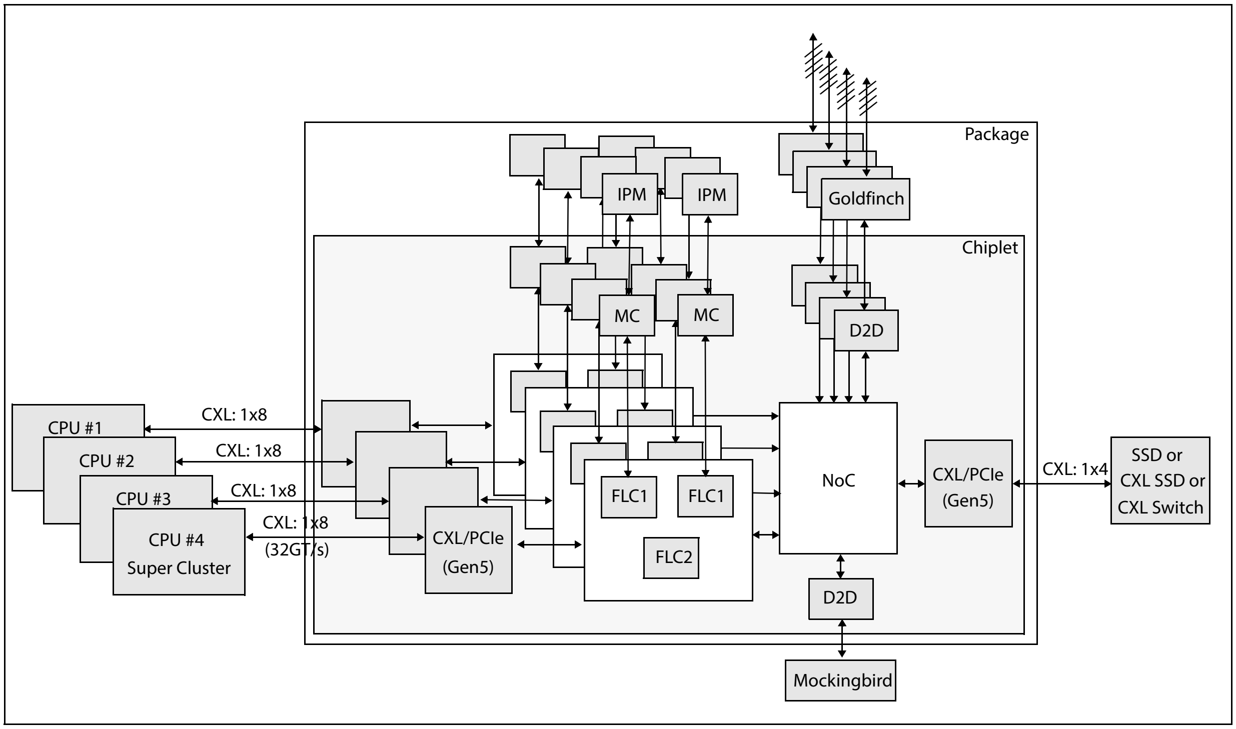
1.1 Overview

The main block diagram includes:

- 4x 1x8 CXL Channel for CPU Sockets
- 4x 2-Level FLC Controller
- 8x Custom IP Dies for FLC Level-1
- 4 Channel DDR4 DIMM or 8 Channel DDR5 DIMM through D2D for FLC Level-2 Caching and Shared Memory Pool
- 2x2 (or 4x4) CXL Channel to PCIe SSD or Switch Fabric
- D2D inter-chiplet Interface to Extend via Clam-Shell Structure (optional feature)

The Figure 1 illustrates the Memory Expander Main Block Diagram

Figure 1 FLC Memory Expander Block Diagram



1.2 Features

This chapter lists out the features present in the FLC Memory Expander.

1.2.1 FLC Memory Controller and In-Package Memory (IPM)

- Fully Associative Look-up Engine for Cache
- 2 (or more) Levels

1.2.2 Final Level Cache (FLC)

- Gigantic entries: Millions
- Cache Line: 4KB
- Number of Entries: 32K/Gb
- Hit Rate: Very High (~99%)

1.2.3 Cache & System Memory

1.2.3.1 In-Package Memory (IPM)

- 1Gb density
- 2CH x 64b @2133Mbps BW 32Gb/s
- Access time: ~33ns
- Access power: ~2pJ/bit
- Support parallel ECC 128 + 8 SECDED

1.2.3.2 DDR4

- 1CH x 64 @3200Mbps BW 25.6Gb/s
- Support parallel ECC 64 + 8 SECDED

1.2.4 High-end Memory Controller

1.2.4.1 Reliability, Availability, Serviceability (RAS)

- End-to-end protection
  - Accepts error sideband signal from upstream masters and report errors for master
  - Parity protected SRAMs
- Supports ECC for all supported DDR bus widths
  - SECDED
  - Error logging for correctable and non-correctable errors
  - Error reporting interrupt
- Row Hammer Protection
- Memory scrubbing
  - Automatically scans the system memory to correct errors and report uncorrectable errors

  - Programmable auto timer between scrubbing or software-triggered scrubber with programmable start and end system address

1.2.4.2 Performance

- Sophisticated DRAM scheduling
- Large write buffer with built-in heuristics
- Agile E2E Quality of Service arbitration with improved latency and bandwidth for critical transactions
- Bandwidth regulators among multiple masters
- Bandwidth monitors
- Read data forwarding
- Write coalescing

1.2.4.3 Debug

- Error poisoning
  - 1-bit or 2-bit ECC errors injections
  - CRC data error injection
- DRAM status monitoring
- Memory controller status monitoring

1.2.4.4 DDR Repair

- Supports DDR4 post-package repair (hPPR and sPPR)
- Shadow memory repair
- Hashed memory repair

1.2.4.5 Performance and Activity Monitoring

- Event based performance monitors
  - AXI transactions
  - Empty / full status for all internal FIFO / buffer / pools
  - DDR commands counter
- Fully user-defined start and end conditions-based monitoring
- Critical visibility to AXI port

1.2.4.6 Power Saving

- Auto power saving control to place DRAM into self-refresh or when no activity detected for extended time frame
- Aggressive clock gating signals PMU to partially or fully gate off MC clock
- Supports firmware or software based dynamic frequency change (DFC) to adjust speed of memory controller and DRAM in order to balance between performance and power consumption

1.2.5 Micro controller Subsystem Components

- Processor: SiFive E21
- Processor Speed: 200MHz
- Operating Mode: **Machine mode** (privilege mode), **User mode**
- Physical Memory Protection: 4 regions (configurable up to 16). PMP is equivalent to ARM’s PMU
- Debug Interface: JTAG
- Peripheral Bus: AHB and APB for embedded peripherals / memory / CSR. Interface to NoC / FLC / CXL
- Interrupt CLIC (63 interrupt sources)
- Timer: Local Timer (x2) and processor embedded Timer. Watchdog Timer
- SRAM (TCM): 356KB, 2 banks (2x128KB)
- Mask ROM: 32KB
- XIP NOR Serial Flash: x4 data bus with SRAM cache
- OTP
- UART (x1): console terminal and firmware download
- I2C (x1)
- SPI (x2): master and device modes
- GPIO: 20 pins
- DMA: 8 channels
- Security Engine: AES and SHA
- Run-time Temperature and Performance Monitoring

2.0 Signal Description

This chapter explains the various signals present in the processor and their usage.

2.1 Pin List

The FLC Memory Expander (Mockingbird) signals are listed and described in Table 1, *Mockingbird Pin List*.

Table 1 Mockingbird Pin List

Type	Pin Name	Category	Comment
input	CB_nRST	Reset	Function reset for chiplet, internal pull-up
input	CB_nPOR	Reset	Power on reset
input	CORERSTn	Reset	CPU subsystem reset
output	PLL_LOCKED	Status	1= SOC PLL locked; clock ready
input	OSCCLK [0]	Clock	Reference clock for SOC PLL, 25MHz
input	ref_100m_p [1:0], ref_100m_n [1:0]	Clock	Reference clock for D2D PLL (LVDS)
input	ref_ipm [3:0]	IPM Clock	Reference clock for IPM PLL, 133.33MHz
inout	PAD_GPIO_[19:0]	GPIO	General purpose IO
output	sf_clk	SF-NOR	Serial flash clock
output	sf_cs	SF-NOR	Serial flash chip select
inout	sf_io [3:0]	SF-NOR	Serial flash data
input	boot_xip	Boot Option	0 = internal ROM 1 = serial flash
inout	I3c_master_scl	I3C	I3C clock
inout	I3c_master_sda	I3C	I3C data
input	I3c_slave_scl	I3C	I3C clock
inout	I3c_slave_sda	I3C	I3C data
inout	ts_an_io [1:0]	TSensor	Temperature sensor Analogue Access[1]
inout	vs_an_io [1:0]	VSensor	Voltage sensor Analogue Access
CXL/PCIE EP, 1x8			
input	ep_perstn [3:0]	CXL_PCIe	Endpoint [3:0] PCIe bus reset
inout	ep_bu_atest [3:0]	CXL_PCIe	
input	ep_cxl_ref_p [3:0], ep_cxl_ref_n [3:0]	CXL_PCIe Clock	Reference clock for host side CXL/PCIe PHY, 100MHz
input	ep_cxl_rx_p [3:0][7:0], ep_cxl_rx_n [3:0][7:0]	CXL_PCIe	RX lane for host side CXL/PCIe
output	ep_cxl_tx_p [3:0][7:0], ep_cxl_tx_n [3:0][7:0]	CXL_PCIe	TX lane for host side CXL/PCIe
output	rc_perstn [1:0]	CXL_PCIe	RC port bus reset
output	rc_x4_perstn [1:0]	CXL_PCIe	
inout	rc_bu_atest [1:0]	CXL_PCIe	

Table 1 Mockingbird Pin List (Continued)

Type	Pin Name	Category	Comment
input	rc_cxl_ref_p [1:0], rc_cxl_ref_n [1:0]	CXL_PCIe Clock	Reference clock for RC side CXL/PCIe PHY, 100MHz
input	rc_cxl_rx_p [1:0][7:0], rc_cxl_rx_n [1:0][7:0]	CXL_PCIe	RX lane for RC side CXL/PCIe
output	rc_cxl_tx_p [1:0][7:0], rc_cxl_tx_n [1:0][7:0]	CXL_PCIe	TX lane for RC side CXL/PCIe
IPM interface for FLC slice [3:0], 2xIPM each slice			
inout	ipm0_DQ [3:0][1:0][1:0][33:0]	IPM0	DQ to/from DRAM
input	ipm0_RDQSB [3:0][1:0][3:0]	IPM0	
input	ipm0_RDQS [3:0][1:0][3:0]	IPM0	
output	ipm0_WDQSB [3:0][1:0][3:0]	IPM0	
output	ipm0_WDQS [3:0][1:0][3:0]	IPM0	
output	ipm0_RESETN_PHY_OUT [3:0][1:0]	IPM0	RESET_n to DRAM
output	ipm0_ADDR [3:0][1:0][9:0]	IPM0	CA to DRAM
output	ipm0_CLK [3:0][1:0]	IPM0	CK_t to DRAM
output	ipm0_CLKN [3:0][1:0]	IPM0	CK_c to DRAM
output	ipm0_CKE [3:0][1:0]	IPM0	CKE to DRAM
output	ipm0_CSN [3:0][1:0]	IPM0	CS to DRAM
inout	Ipm1_DQ [3:0][1:0][1:0][33:0]	IPM1	DQ to/from DRAM
input	Ipm1_RDQSB [3:0][1:0][3:0]	IPM1	
input	Ipm1_RDQS [3:0][1:0][3:0]	IPM1	
output	Ipm1_WDQSB [3:0][1:0][3:0]	IPM1	
output	Ipm1_WDQS [3:0][1:0][3:0]	IPM1	
output	Ipm1_RESETN_PHY_OUT [3:0][1:0]	IPM1	RESET_n to DRAM
output	Ipm1_ADDR [3:0][1:0][9:0]	IPM1	CA to DRAM
output	Ipm1_CLK [3:0][1:0]	IPM1	CK_t to DRAM
output	Ipm1_CLKN [3:0][1:0]	IPM1	CK_c to DRAM
output	Ipm1_CKE [3:0][1:0]	IPM1	CKE to DRAM
output	Ipm1_CSN [3:0][1:0]	IPM1	CS to DRAM
D2D interface for DDR4 chiplet (4x1x16)			
inout	d2d_ddr_tx0_d [3:0][15:0]	D2D 1x16	
output	d2d_ddr_tx0_aux [3:0]	D2D 1x16	
output	d2d_ddr_tx0_fec [3:0]	D2D 1x16	
inout	d2d_ddr_clk_tx0_p [3:0]	D2D 1x16	
inout	d2d_ddr_clk_tx0_n [3:0]	D2D 1x16	
inout	d2d_ddr_rx0_d [3:0][15:0]	D2D 1x16	
inout	d2d_ddr_rx0_aux [3:0]	D2D 1x16	
inout	d2d_ddr_rx0_fec [3:0]	D2D 1x16	

Table 1 Mockingbird Pin List (Continued)

Type	Pin Name	Category	Comment
inout	d2d_ddr_clk_rx0_p [3:0]	D2D 1x16	
inout	d2d_ddr_clk_rx0_n [3:0]	D2D 1x16	
MKB D2D interface for inter-chiplet (2x1x16)			
inout	d2d_tx0_d [1:0][15:0]	D2D 1x16	
output	d2d_tx0_aux [1:0]	D2D 1x16	
output	d2d_tx0_fec [1:0]	D2D 1x16	
inout	d2d_clk_tx0_p [1:0]	D2D 1x16	
inout	d2d_clk_tx0_n [1:0]	D2D 1x16	
inout	d2d_rx0_d [1:0][15:0]	D2D 1x16	
inout	d2d_rx0_aux [1:0]	D2D 1x16	
inout	d2d_rx0_fec [1:0]	D2D 1x16	
inout	d2d_clk_rx0_p [1:0]	D2D 1x16	
inout	d2d_clk_rx0_n [1:0]	D2D 1x16	
EMAC			
input	mtx_clk		
output	mtx_d [3:0]		
output	mtx_en		
output	mtx_err		
input	mrx_clk		
input	mrx_d [3:0]		
input	mrx_dv		
input	mrx_err		
input	mcol		
input	mcrs		
output	mdc		
inout	mdio		
Debug and Trace			
input	nTRST	JTAG	JTAG reset
input	TCK	JTAG	JTAG clock
input	TMS	JTAG	JTAG mode select
input	TDI	JTAG	JTAG data input: TDI_IPM_PHY, TDI_PCIE_PHY, TDI_D2D
output	TDO	JTAG	JTAG data output: TDO_IPM_PHY, TDO_PCIE_PHY, TDO_D2D
input	nTRST_E21	JTAG	JTAG reset
input	TCK_E21	JTAG	JTAG clock
input	TMS_E21	JTAG	JTAG mode select
input	TDI_E21	JTAG	JTAG data input: TDI_E21
output	TDO_E21	JTAG	JTAG data output: TDO_E21
input	nTRST_RISCV	JTAG	JTAG reset
input	TCK_RISCV	JTAG	JTAG clock

Table 1 Mockingbird Pin List (Continued)

Type	Pin Name	Category	Comment
input	TMS_RISCV	JTAG	JTAG mode select
input	TDI_RISCV	JTAG	JTAG data input: TDI_RISCV
output	TDO_RISCV	JTAG	JTAG data output: TDO_RISCV
input	scan_mode	BSCAN	Scan mode
input	BSCAN_TRST	BSCAN	BSCAN reset
input	BSCAN_TCK	BSCAN	BSCAN clock
input	BSCAN_TMS	BSCAN	BSCAN mode select
input	BSCAN_TDI	BSCAN	BSCAN data input
output	BSCAN_TDO	BSCAN	BSCAN data output
Power and ground			
Power	vdd_io	PWR	IO power, 1.8 V +/- 10%
Ground	vss_io	GND	IO combined ground
Power	vdd_core	PWR	Core power, 0.8 V +/- 10%
Ground	vss_core	GND	Core combined ground
Power	ipm_vdd2	IPM-PWR	IPM PHY 1.1V IO/PLL power
Power	ipm_vddq	IPM-PWR	IPM PHY 0.3V IO power
Ground	ipm_vss	IPM-GND	
Power	pcie_phy_p_vdd	PCIe-PWR	
Ground	pcie_phy_p_vss	PCIe-GND	
Power	pcie_phy_pa_vddh	PCIe-PWR	PCIe PHY analog 1.2V power
Power	pcie_phy_pa_vddl	PCIe-PWR	PCIe PHY analog 0.75V power
Ground	pcie_phy_pa_vss	PCIe-GND	PCIe PHY analog ground
Power	pll_vddhv	SOC-PLL-PWR	SOC PLL Analog Supply Voltage, 1.8 V +/- 10%
Power	pll_vddpost	SOC-PLL-PWR	SOC PLL Digital Supply Voltage, 0.8 V +/- 10%
Power	pll_vddref	SOC-PLL-PWR	SOC PLL Digital Supply Voltage, 0.8 V +/- 10%
Ground	pll_vss	SOC-PLL-GND	
Power	hs_pll_vddhv	D2D-PLL-PWR	D2D PLL Analog Supply Voltage, 1.8 V +/- 10%
Power	hs_pll_vddpost	D2D-PLL-PWR	D2D PLL Digital Supply Voltage, 0.8 V +/- 10%
Power	hs_pll_vddref	D2D-PLL-PWR	D2D PLL Digital Supply Voltage, 0.8 V +/- 10%
Ground	hs_pll_vss	D2D-PLL-GND	
Power	d2d_vddq_s	D2D-PWR	D2D PHY VDDQ Supply Voltage, 0.3/0.5/0.825 V
Power	d2d_vddclk_s	D2D-PWR	D2D PHY VDDCLK Supply Voltage, 0.8 V +/- 10%
Power	d2d_vdda_s	D2D-PWR	D2D PHY VDDA Supply Voltage, 1.2 V +/- 10%
Power	d2d_vdd_s	D2D-PWR	
Ground	d2d_vss_s	D2D-GND	
Power	otp_vdd	OTP	0.8 V
Power	otp_vdd2	OTP	OTP power, 1.8 V +/- 10%

Table 1 Mockingbird Pin List (Continued)

Type	Pin Name	Category	Comment
Ground	otp_vss	OTP	
Power	VDDA_TS		Temperature Sensor Analog Supply Voltage, 1.8 V +/- 10%
Power	VDDA_VS		Voltage Monitor Analog Supply Voltage, 1.8 V +/- 10%

1.

2.2 GPIO Pin Multiplex

The Microcontroller GPIO pins are multiplexed with embedded peripheral functions. Table 2, *GPIO Pin Multiplex* lists the GPIO Pin Multiplex

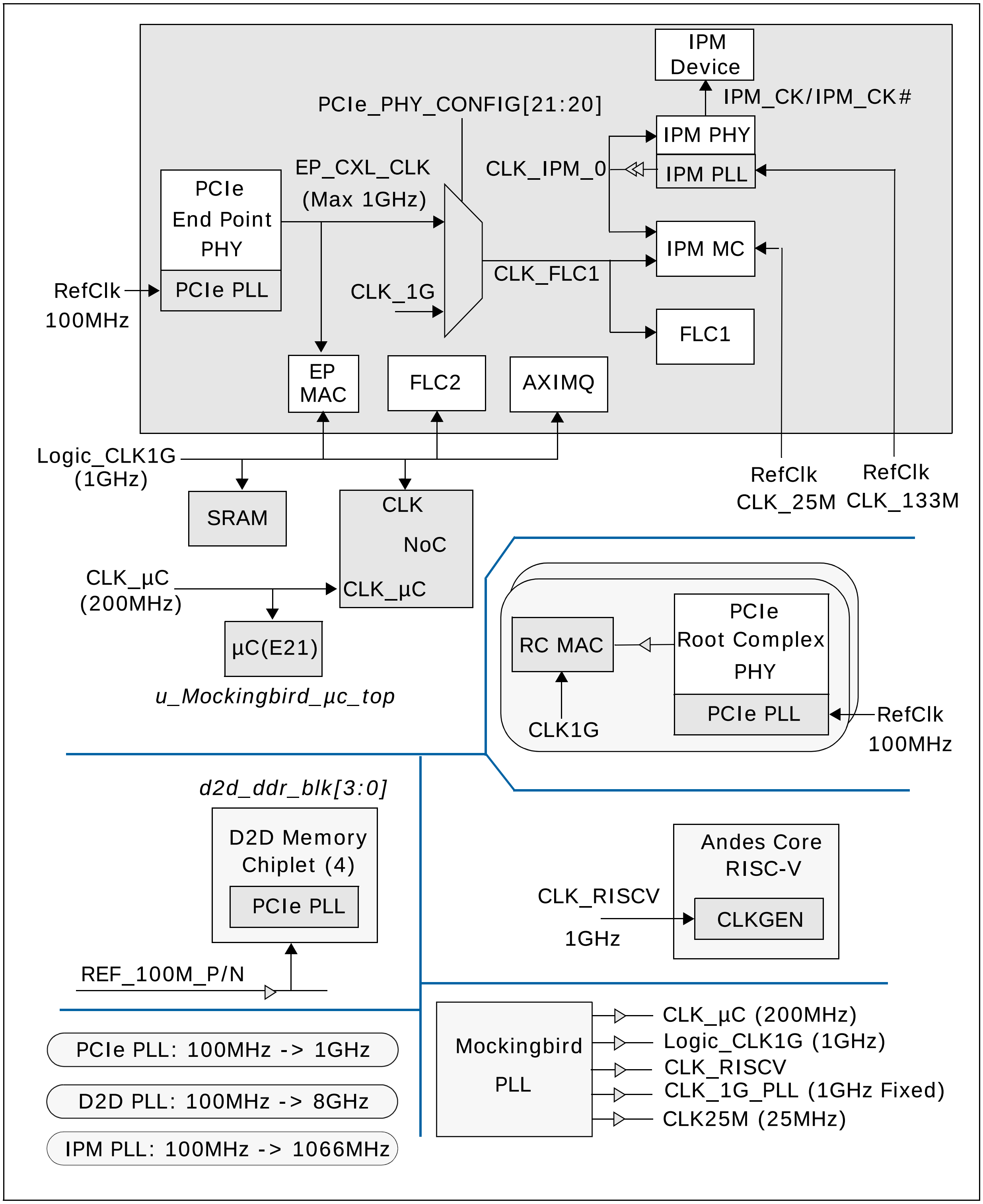
Table 2 GPIO Pin Multiplex

GPIO#	Function Select	Function Pins	Direction	Pull Up	Pull Down	Comment
0	1	uart_rts	o	n	n	riscv_txd
1	1	uart_cts	i	n	n	riscv_rxd
2	1	uart_txd	o	n	n	
3	1	uart_rxd	i	n	n	
4	2	spi_mosi	o if master mode	n	n	master mode=0x80.0
5	2	spi_miso	i if master mode	n	n	swap miso/mosi=0x80.1
6	2	spi_ss_o	o if master mode	n	n	
7	2	spi_clk	o if master mode	n	n	
8	3	i2c_scl	bi	y		
9	3	i2c_sda	bi	y		
10	0	gpio_reg	bi			CHIP_ID/spi1_mosi
11	0	gpio_reg	bi			CHIP_ID/spi1_miso
12	0	gpio_reg	bi			BOOT_LED[0]/spi1_ss_o
13	0	gpio_reg	bi			BOOT_LED[1]/spi1_clk
14	0	gpio_reg	bi			BOOT_LED[2]/config_spiclk
15	0	gpio_reg	bi			I3C_ready (GFH3's GPIO18)/config_spi_di
16	0	gpio_reg	bi			I3C_ready (GFH2's GPIO18)/config_spi_do
17	0	gpio_reg	bi			I3C_ready (GFH1's GPIO18)
18	0	gpio_reg	bi			I3C_ready (GFH0's GPIO18)
19	0	gpio_reg	bi			FW_MODE

2.3 Clock Domain

Figure 2 illustrates the Clock domain of the FLC Memory Expander Controller.

Figure 2 Clock Domains



2.3.1 FLC Slice Clocks

This section explains the clock distribution present within the FLC slices.

The flc1_axi_slv_config bits in register PCIE_PHYCONFIG (0x2601003C) defines the clock source selection.

- Register: PCIE_PHYCONFIG, Address: 0x2601003C

Bits	Bit Name	Default	Type	Comment
21:20	flc1_axi_slv_config	2’b0	R/W	00 = CXL.mem -> flc1 01 = CXL.io -> flc1 1x = D2D (host) -> flc1

- Register: PMU_CTRL, Address: 0x26000004

Bits	Bit Name	Default	Type	Comment
4	flc1_axi_slave_sync_mode	1’b1	R/W	1 = flc1's axi slave interface runs in synchronous mode
3	ipm_mc_axi_sync_mode	1’b0	R/W	1 = ipm_mc's axi interface runs in synchronous mode

Table 3 Clock Source Selection when flc1_axi_slv_config == 2’b00

ipm_mc_axi_p0_sync	flc1_axi_slave_sync	flc1_axi_slave	flc1
x	1	ep_cxl_clk	ep_cxl_clk

ep_cxl_clk is the PCIe PHY generated clock. Frequency variations are based on the mode negotiation.

Table 4 Clock Source Selection when flc1_axi_slv_config != 2’b00

ipm_mc_axi_p0_sync	flc1_axi_slave_sync	flc1_axi_slave	flc1
x	1	clk_nic (clk1g)	clk_nic (clk1g)

2.3.2 SOC Clocks

Table 5 lists the clocks present in the SOC.

Table 5 SOC Clocks

Name	Working Frequency	Source	Comment
clk1g (clk1g_slice[3:0])	1GHz	Mockingbird PLL	Main clock for each slice
clk1g_noc, clk1g_noc_lp	1GHz	Mockingbird PLL	NOC clock
clk_riscv	1GHz	Mockingbird PLL	Andes CPU
clk1g_axi4tg	1GHz	Mockingbird PLL	AXI4 traffic gen
clk_uc	200MHz	Mockingbird PLL	uC subsystem
clk_ipm	1066MHz	IPM PHY	
clk_flc1_slave	Up to 1Ghz	Mockingbird PLL or PCIe EP PHY	FLC1 slave port
clk_flc1	1Ghz	Mockingbird PLL	FLC1 core clock
clk_flc2	1Ghz	Mockingbird PLL	FLC2 core clock
clk_rc_x8[1:0]	Up to 1Ghz	RC x8 PHY	
clk_rc_x4[1:0]	Up to 1Ghz	RC x4 PHY	
ref_100m_p [1:0], ref_100m_n [1:0]	100MHz	External	100MHz differential clock for D2D PLL

Table 5 SOC Clocks (Continued)

Name	Working Frequency	Source	Comment
ref_ipm [3:0]	133.33MHz	External	IPM PLL reference clock
ep_cxl_ref_p[3:0], ep_cxl_ref_n[3:0]	100MHz	External	PCIe/CXL EP PHY reference clock
rc_cxl_ref_p[1:0], rc_cxl_ref_n[1:0]	100MHz	Externa	PCIe/CXL RC PHY reference clock
clk25m	25MHz	Externa	

Table 6 lists the external clock sources examples.

Table 6 External Clock Sources Examples

Frequency	Part Number	Comment
25MHz	ASD3-25.000MHZ-LR-T	±25ppm
133.33MHz	SIT8009BI-13-18E-133.333333	±50ppm
100MHz	Si52204-A01BGMR	0.085ps rms, PCIe Gen 6

2.3.3 MKB main PLL Configuration

By changing the MKB main PLL control registers MKB_PLL_CTRL3~MKB_PLL_CTRL1 (mkb_pll_ctl [95:0], 0x2600_0108~0x2600_0100), different output frequencies can be reached. Note that pll_en (0x2600_0100[0]) = 1'b0 by default, the input refclk is bypassed to all outputs, need to set it to 1'b1 to enable the PLL.

For example, if Input frequency (refclk) = 25MHz and

mkb_pll_ctl[95:64], 0x2600_0108 = 32’hF000_0000 (Default) and

mkb_pll_ctl[31:0], 0x2600_0100 = 32’h0000_0011 (Default and pll_en = 1’b1),

mkb_pll_ctl [63:32], 0x2600_0104	clk1g	clk_riscv	clk_uc	clk25m
32’h7110_1401	1GHz	1GHz	200MHz	25MHz
32’h8220_1801	800MHz	800MHz	200MHz	25MHz
32’h8330_1801	600MHz	600MHz	200MHz	25MHz
32’h7330_1401	500MHz	500MHz	200MHz	25MHz
32’ha340_2801	800MHz	1GHz	200MHz	25MHz
32’hB460_3001	600MHz	960MHz	200MHz	25MHz
32’h7130_1401	500MHz	1GHz	200MHz	25MHz

2.4 Reset and Power Up

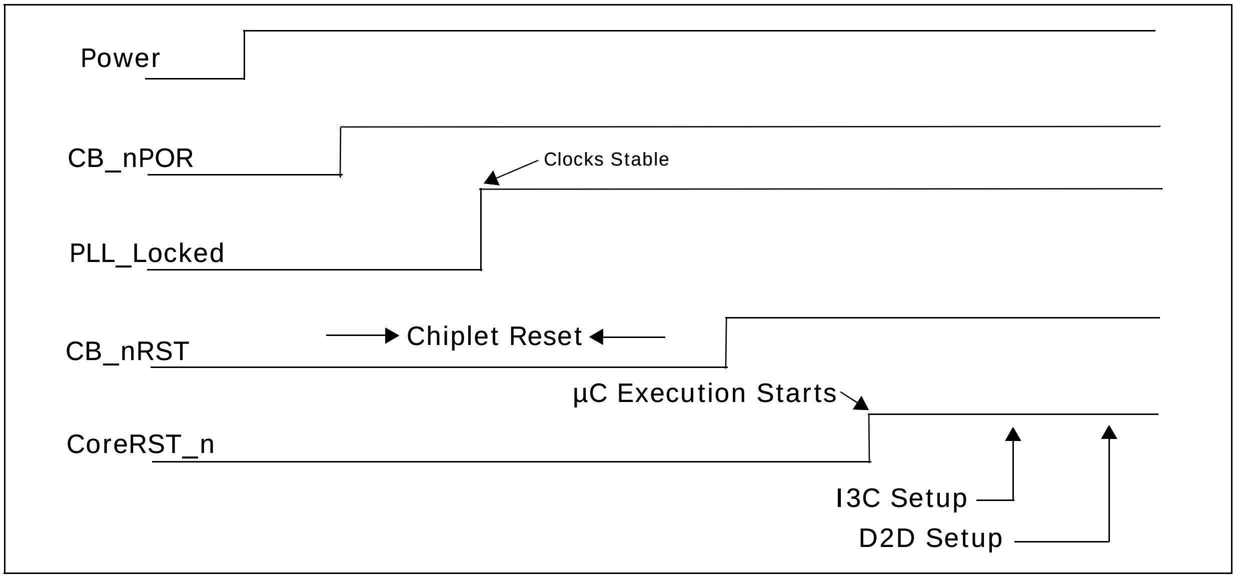
- Power applied. System clock stable
- E21 boot
- PHY (IPM, PCIe, DDR, D2D) FW download
- PHY (IPM, PCIe, DDR, D2D) init
- PHY output clock stable (IPM, D2D, PCIe clock)
- I3C, D2D LL sync

- Functional unit init
  - PCIe MAC
  - IPM, DDR MC
  - FLC
- RISC-V system bring up
  - Code download
  - ETM

2.4.1 Chip Reset and Clock Ports

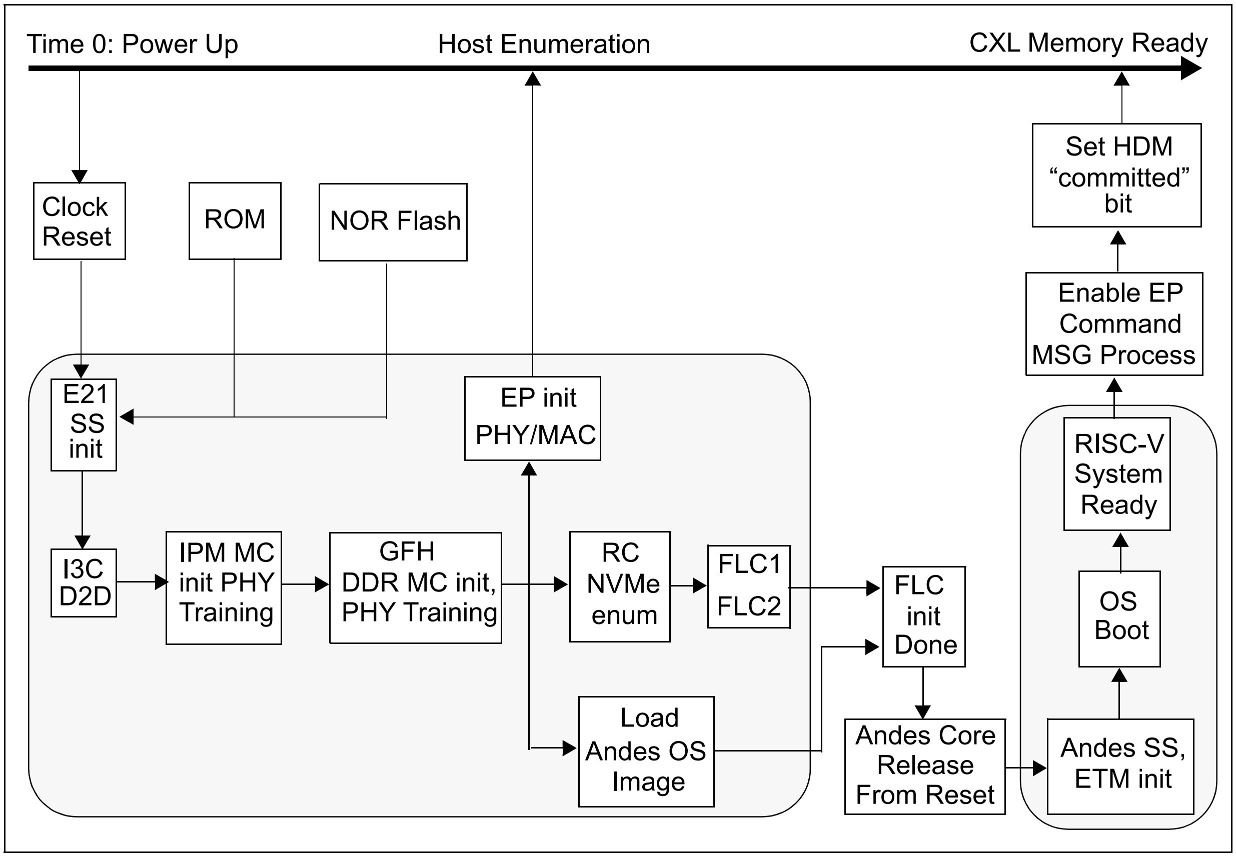
- CB_nPOR: Active low, power on reset. Reset PLL and clock circuitry
- CB_nRST: Active low, global chip reset
- CORERSTn: Active low, reset embedded µC
- External reset chip required to detect stable voltage level and reset timing
- OSC: 25MHz single ended reference clock for SOC PLL
- ref_100m_p/n [1:0]: 100MHz differential reference clocks for D2D PLL
- ref_ipm [3:0]: 133.33Mhz single ended reference clock for IPM PLL

Figure 3 Reset and Clock Timing Diagram



2.4.2 Startup Timeline for MKB in CXL.mem Mode

Figure 4 MKB Startup Timeline (CXL.mem Mode)



3.0 Chiplet Modules

This chapter describes the various functional modules in the Memory Expander.

3.1 NOC and AHB / APB Bus Matrix

Figure 5 NOC and AHB / APB Bus Matrix

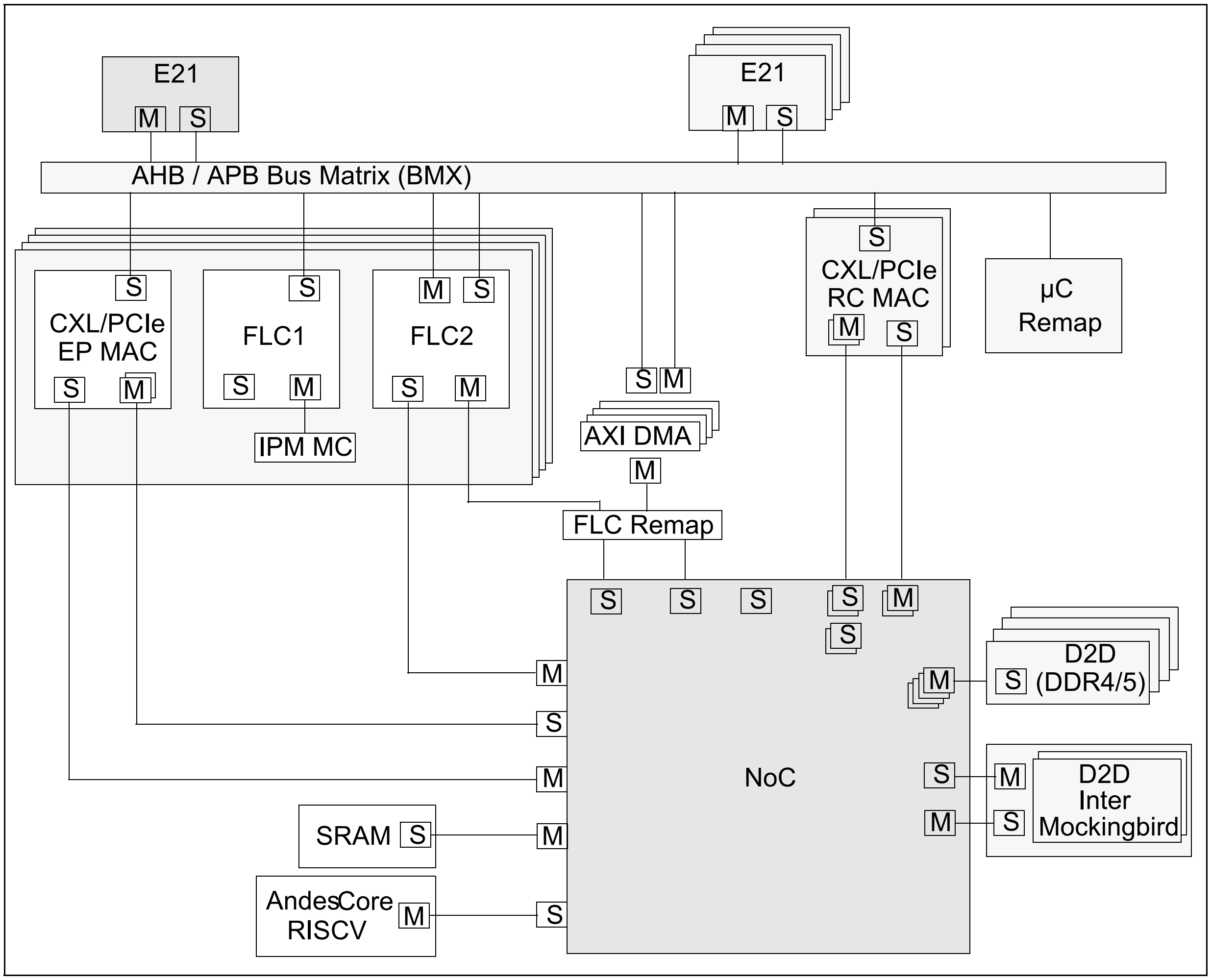


Table 7 lists the AXI target and initiator present in NoC interface.

Table 7 Mockingbird 1x16 D2D

Region	D2D	NoC Interface AXI Target	NoC Interface: AXI Initiator	Peak BW
GFH	4x(1x16)	4	0	4x32GB/s
Inter-Chiplet	2x(1x16)	2	2	4x32GB/s
Total	6x(1x16)	6	0	192GB/s

3.2 FLC

FLC handles hierarchy memory with FLC controller between each tier is illustrated below. In the chiplet, conventional memory is expanding into a 3-tier hierarchy with IPM serves as the Level 1 cache, DDR serves as the Level 2 cache and main memory resides in SLC NAND.

FLC1 controller manages data between IPM and DDR. FLC2 controller manages data between DDR, NAND, or remote memories (CXL, PCIe, NVMe). µC programs module CSRs within FLC during initialization.

FLC2 issues AXI cycle to access DDR, NAND or remote through NOC.

FLC1 controller, FLC2 controller, IPM memory controller and LP4X memory controller are on the same Mockingbird slice. IPM will be on a different die but same package. Both DDR and NAND are off-chip.

There are total of 4 Mockingbird slices per generation 1 chiplet.

Figure 6 illustrates the MKB chiplet.

Figure 6 MKB Chiplet

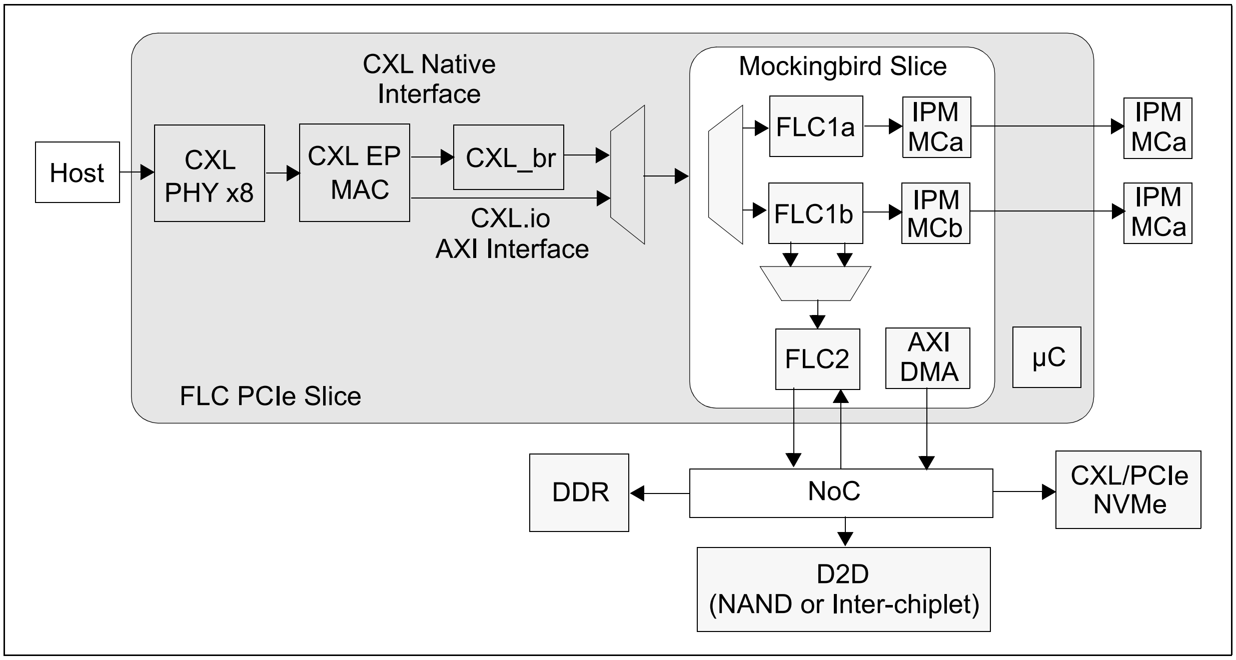
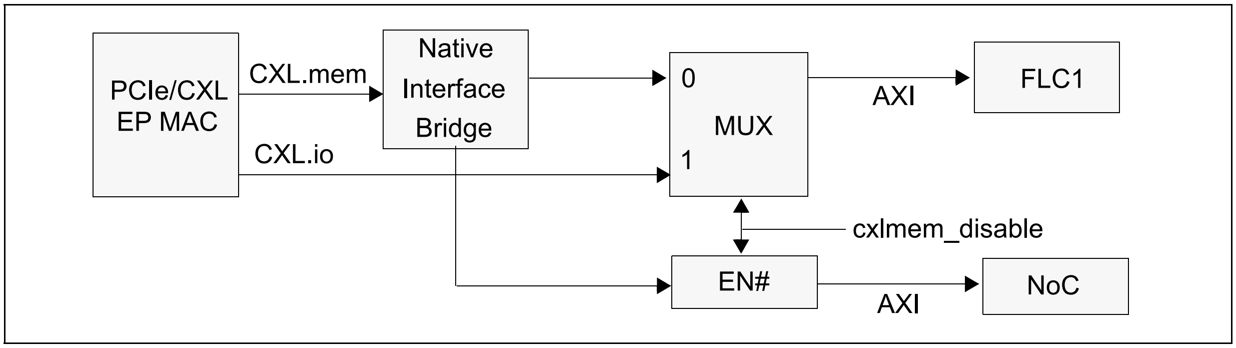


Figure 7 Data Transaction within Chiplet



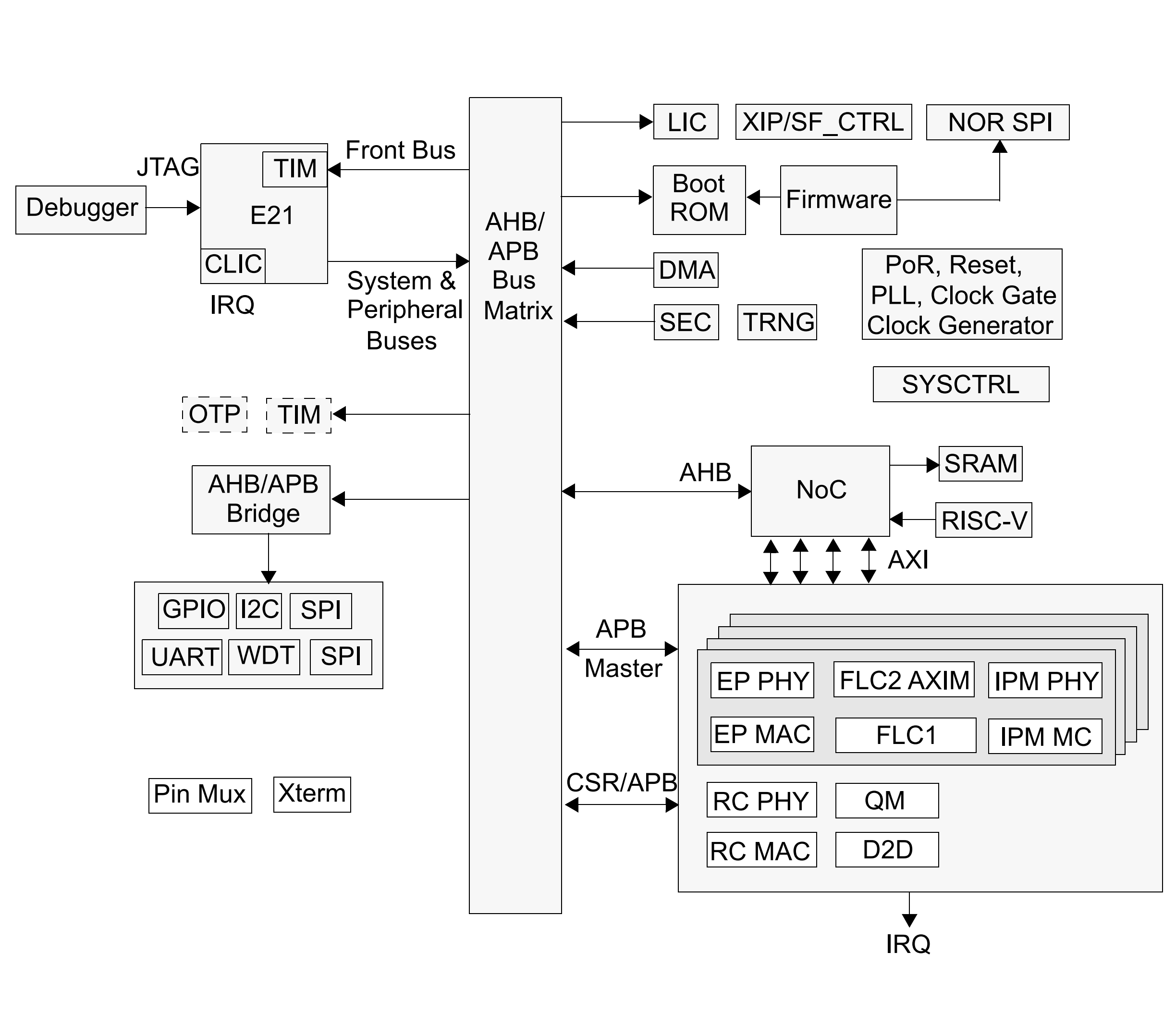
3.3 Microcontroller Subsystem

The micro controller subsystem design is based on SiFive E21 RISCV core and Buffalo Lab peripheral IPs. Firmware is stored in on-chip mask ROM and off-chip XIP NOR flash. An AHB/APB bus matrix is used to connect CPU, memory and IO. The µC subsystem also interfaces with chiplet's NOC via AHB to access chiplet resources (SRAM, DDR).

After reset, µC performs system initialization through AHB/APB bridge (refer to Table 48). There are two local timers, a watchdog timer and core embedded timer for system timing control. Interrupt via E21's CLIC module with total of 63 maskable interrupt sources and 1 non-maskable NMI (Refer to Table 49).

Two on-core tightly couple memories (TIM), 128KB each are included for fast execution. Figure 8 illustrates the microcontroller subsystem.

Figure 8 Microcontroller Subsystem



3.3.1 Embedded IO Summary

Table 8 lists the embedded IO summary.

Table 8 Embedded IO Summary

Name	Mode	FIFO	DMA	Interrupt	Comment
UART	Slave	32	Yes	Yes	Interface for debug and firmware download
I2C	Master	2	Yes	Yes	
SPI0	Master/Slave	4	Yes	Yes	
SPI1	Master/Slave	4	Yes	Yes	
TIMER	Slave			Yes	2 programmable timers, 1 watchdog timer
GPIO	Slave	NA		Yes	20 GPIO pins with configurable driving mode and pull up/down. Refer to section 2.2 GPIO Pin Multiplex and section 6.13 GPIO Control Register
DMA				Yes	
SEC	AHB Master			Yes	Security engine supports hardware based AES and SHA
OTP	Slave	SA		Yes	Only in MKB chiplet

3.3.2 AHB / APB Bus Matrix

The µC bus matrix supports 10 AHB masters and 4 AHB slaves with hard-coded address range. There are also subsection decoders for individual module's CSR. To program multiple D2D or IPM or PCIe PHY slices, the `*_phy_sel` bits in register `INTERFACE_SEL (0x26000038)` need to be set with target unit ID. Table 9 lists the matrix for masters.

Table 9 .Matrix for Masters

Name	Comment
µC System Bus	µC (E21) system port for code fetch
FLC APB master 2	APB master from (EP) PCIe ELBI
DMA	Connect to DMA engine's AHB master port
EMAC	EMAC AHB master
CONFIG_SPI	SPI BFM port (RTL simulation only)
µC peripheral bus 1	µC (E21) peripheral port 1
µC peripheral bus 2	µC (E21) peripheral port 2
AES	Security engine (AES) master port
SHA	Security engine (SHA) master port
FLC APB master	FLC APB master for hardware based NVMe queue management

Table 10 Matrix for Slaves

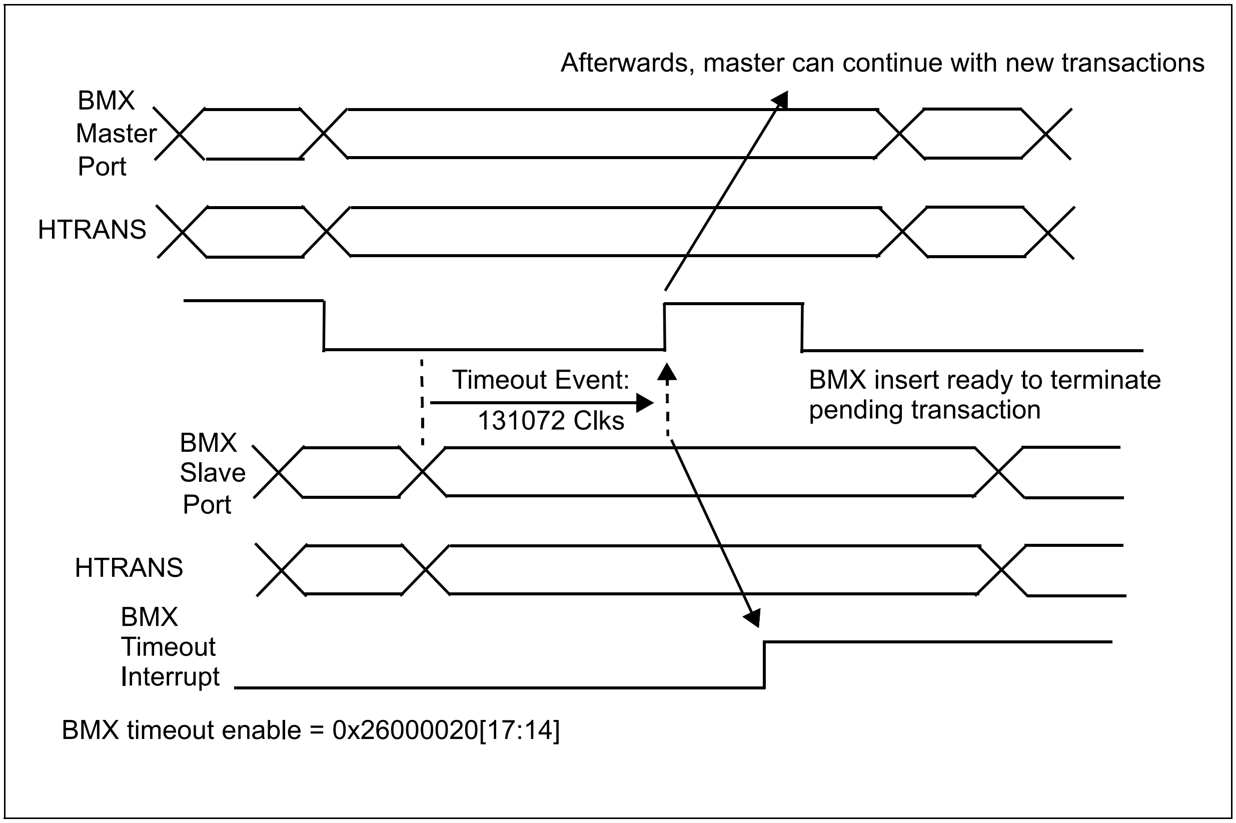
Name	Comment
Mem	Memory region, mask ROM and XIP NOR

Name	Comment
Peripheral	Peripheral region, includes µC embedded IO and FLC, PCIe controller CSRs
NoC	NoC AHB slave port for DDR, NoC-SRAM access
TIM	µC embedded memory (SRAM)

3.3.3 Bus Timeout

Figure 9 illustrates the bus matrix timeout

Figure 9 Bus Matrix Timeout


Afterwards, master can continue with new transactions
BMX Master Port
HTRANS
Timeout Event:
131072 Clks
BMX insert ready to terminate pending transaction
BMX Slave Port
HTRANS
BMX Timeout Interrupt
BMX timeout enable = 0x26000020[17:14]

3.3.4 PHY Select

Steps to access D2D/IPM/PCIe PHY

1. For D2D, set `mkb_d2d_resetn[1:0]` and/or `d2d_ddr_resetn[3:0]` in register `PMU_CTRL (0x26000004)` to enable the D2D interface.
2. Set `*_phy_sel` bits in register `INTERFACE_SEL (0x26000038)` to enable one of the D2D or IPM or PCIe PHY unit.
3. Access unit CSR based on Table 48, *E21 Address Map*

- Register: PMU_CTRL, Address: 0x26000004

Bits	Bit Name	Default	Type	Comment
31:28	flc_slice_rst[3:0]	4'h0	R/W	1 = FLC PCIe slice [3:0] hardware reset Note: IPM_PHY and CSR do not reset by this bit
17:16	mkb_d2d_resetn[1:0]	1'h0	R/W	0 = MKB_D2D interface in reset state 1 = MKB_D2D interface is enabled [17]: MKB_D2D #1 [16]: MKB_D2D #0
11:8	d2d_ddr_resetn[3:0]	4'h0	R/W	0 = D2D_DDR (Goldfinch D2D interface) in reset state 1 = D2D_DDr interface is enabled [11]: Goldfinch #3 [10]: Goldfinch #2 [9]: Goldfinch #1 [8]: Goldfinch #0

- Register: INTERFACE_SEL, Address: 0x26000038

Bits	Bit Name	Default	Type	Comment
19:12	d2d_io_a31x24	8'h2A	R/W	D2D APB (A5L) access upper address bits: A[31:24]. Default points to apbx (0x2Axxxxxx)
11:8	d2d_phy_sel	4'h0	R/W	Select D2D for configuration 0~3: 1st 1x16 D2D_DDR[0:3] 4: MKB_D2D0 5: MKB_D2D1 6~14: Reserved (default D2D_DDR0) 15: Broadcast all, this field combines with address windows 0x27xx_xxxx for D2D decoding
7:4	ipm_phy_sel	4'h0	R/W	Select IPM PHY for configuration. (register: 0x2600_0064) 0: FLC slice 0, ipm0 1: FLC slice 0, ipm1 2: FLC slice 1, ipm0 3: FLC slice 1, ipm1 4: FLC slice 2, ipm0 5: FLC slice 2, ipm1 6: FLC slice 3, ipm0 7: FLC slice 3, ipm1 8: Reserved for ipm9 9~14: Reserved 15: Broadcast all
[3:0]	pcie_phy_sel	4'h0	R/W	Select CXL/PCIe interface for configuration. 0: RCx8[0] MAC [NVMe host/CXL RC] 1~4: host side slice pcie_ep[0:3] 5: RCx4[0] MAC 6: RC1_x8[1] MAC 7: RC1_x4[1] MAC 8~13: Reserved 14: Broadcast (RCx8[0:1]) 15: Broadcast all

3.4 SPI Module

3.4.1 SPI Overview

Serial Peripheral Interface (SPI) bus is a synchronous serial communication interface specification for short-range communication. Devices communicate in the full duplex mode, which is a master-slave mode where one master controls one or more slaves.

SPI completes full-duplex communication with four signal lines, namely chip select (CS), SCLK (clock), master output slave input (MOSI), and master input slave output (MISO).

3.4.2 SPI Features

- Can be used as SPI master or slave
- Both master and slave support four operating modes (CPOL and CPHA)
- Both master and slave support 1/2/3/4-byte transfer mode
- The sending and receiving channels each have a FIFO with a depth of 32 bytes
- The adaptive FIFO depth variation characteristic suits high-performance applications
  - — When the frame is 32-bits, the depth of FIFO is 8
  - — When the frame is 24-bits, the depth of FIFO is 8
  - — When the frame is 16-bits, the depth of FIFO is 16
  - — When the frame is 8-bits, the depth of FIFO is 32
- Adjustable byte transfer sequence
- Flexible clock configuration
- Configurable MSB/LSB transfer priority
- Receive ignore function: You can set to ignore the reception of data from the specified location
- Supports timeout mechanism in the slave mode
- Supports DMA transfer mode

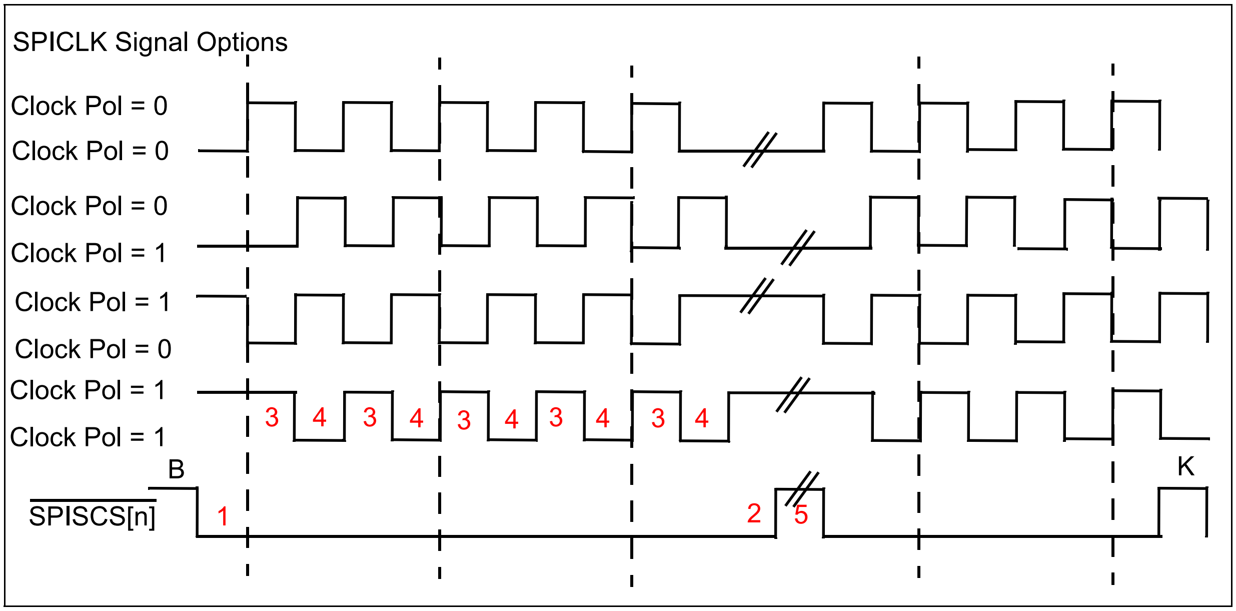
3.4.3 SPI Functional Description

3.4.3.1 SPI Clock Control

Due to different clock phases and polarity settings, the SPI clock has four modes, which can be set by `cr_spi_sclk_pol(CPOL)` and `cr_spi_sclk_ph(CPHA)` in the register `SPI_CONFIG`. The CPOL determines the level of SCK clock signal when it is idle. If `CPOL=0 (cr_spi_sclk_pol=0)`, the idle level is low and if `CPOL=1 (cr_spi_sclk_pol=1)`, the idle level is high. The CPHA determines the sampling time. If `CPHA=0 (cr_spi_sclk_ph=1)`, sampling is made at the first lock edge of each cycle and if `CPHA=1 (cr_spi_sclk_ph=0)`, that is made at the second clock edge of each cycle.

By setting the registers SPI_PRD_0 and SPI_PRD_1, you can also adjust the duration of the start and end levels of the clock, time of phase 0/1, and interval between frames of data. The settings of the four modes are shown in Figure 10.

Figure 10 SPT Timing



The meanings of numbers are as follows:

- "1" denotes the length of the START condition, which is configured by cr_spi_prd_s in the register SPI_PRD_0.
- "2" denotes the length of the STOP condition, which is configured by cr_spi_prd_p in the register SPI_PRD_0.
- "3" denotes the length of phase 0, which is configured by cr_spi_prd_d_ph_0 in the register SPI_PRD_0.
- "4" denotes the length of phase 1, which is configured by cr_spi_prd_d_ph_1 in the register SPI_PRD_0.
- "5" denotes the interval between frames of data, which is configured by cr_spi_prd_i in the register SPI_PRD_1.

3.4.3.2 Master-Slave: Transfer and Receive Data

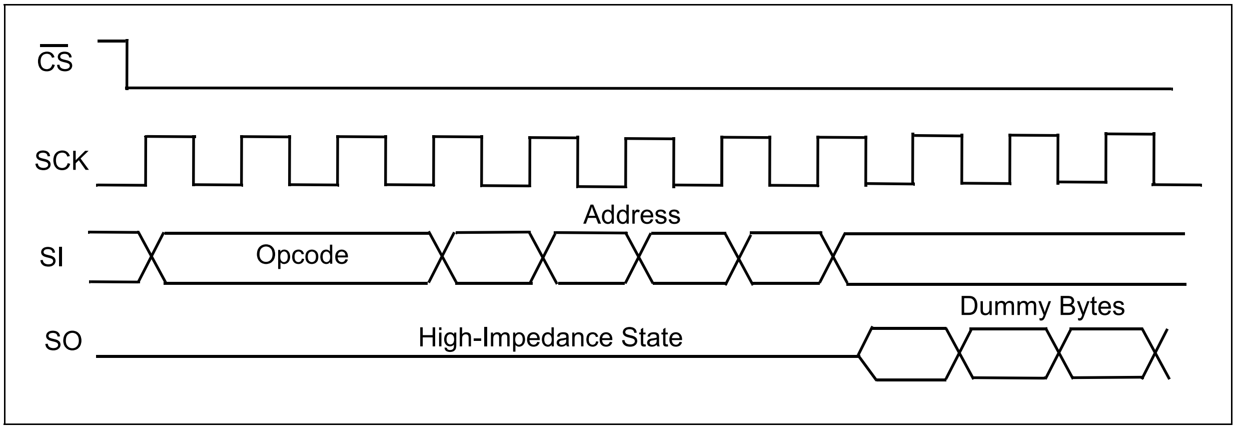
The same frame_size shall be set for the master and slave for transferring and receiving data by configuring the cr_spi_frame_size in the register SPI_CONFIG. When the master and slave agree to communicate at a 32-bits frame_size, if the clk of the master does not meet 32-bits due to an exception in a frame of data, the following symptoms occur.

- The data sent by the master cannot be transferred to the RX FIFO of the slave. The slave cannot receive data from the master.
- When the slave sends data, it will skip this frame of data and continue to send the next frame of data when the master's clk is normal again.

3.4.3.3 Receive Ignore Function

When the start and end bits to be filtered out are set, SPI will discard the corresponding data segments in the received data as shown in Figure 11.

Figure 11 Ignore Waveform



You can enable this function by configuring the `cr_spi_rxd_ignr_en` in the register `SPI_CONFIG`. The start bit of this function is set by configuring the `cr_spi_rxd_ignr_s` in the register `SPI_RXD_IGNR`. The end bit of this function is set by configuring the `cr_spi_rxd_ignr_p` in the register `SPI_RXD_IGNR`.

In the above Figure 11, the start bit to be filtered is set to 0, if the end bit is set to 7, Dummy Byte will be received; if the end bit is set to 15, Dummy Byte will be discarded.

3.4.3.4 Filtering Function

When this function is enabled and a threshold is set, SPI will filter the data less than or equal to the width threshold. Assuming that the SPI top clock is 160 MHz and the threshold is set to 4, any data with a width below (4/160 MHz = 25 ns) will be filtered out.

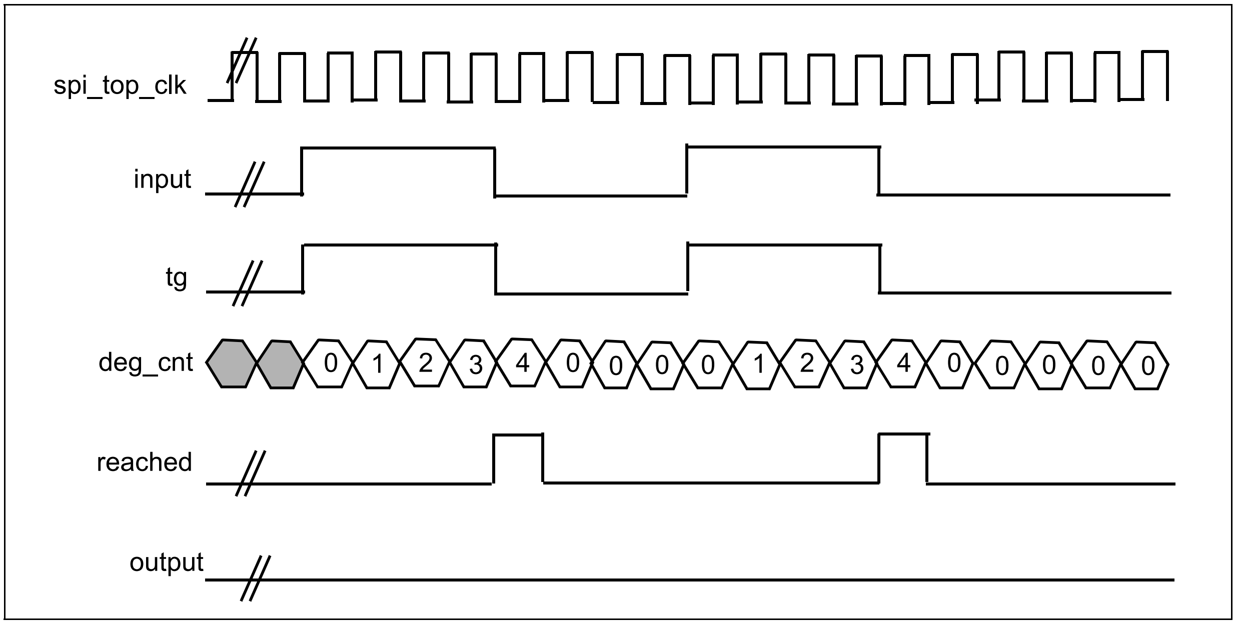
When this function is enabled by setting the `cr_spi_deg_en` in the register `SPI_CONFIG` and the threshold is set by configuring the `cr_spi_deg_cnt`, SPI will filter out the data that cannot reach the width threshold. As shown in the Figure 12 below, the data width is 4, Input is the initial data and Output is the filtered data.

Filtering logic process:

- `Tgl` is the exclusive XOR result of input and output.
- `Deg_cnt` counts from 0, and the counting condition is that `tgl` is at the high level and reached is at the low level.
- Reached means whether the current `deg_cnt` count reaches the set `cr_spi_deg_cnt` and it is at a high level once reached.
- When reached is at a high level, input is output to output.

Note: User-defined condition for deg_cnt: tgl is at a high level and reached is at a low level. In other cases, deg_cnt will be cleared to 0.

Figure 12 SPI Filter Waveform



3.4.3.5 Configurable MSB / LSB Transfer

The configurable MSB/LSB transfer mode is limited to the priority transfer sequence of 8-bits in one byte and the transfer sequence of bits in one byte is set by configuring the cr_spi_bit_inv bit in the register SPI_CONFIG. 0 indicates MSB and 1 indicates LSB. For example, for data transfer where the frame size is 24-bits, the data format is Data[23:0] = 0x123456.

When MSB transfer is set, the transfer sequence is: 01010110 (binary, 1st byte: 0 × 56); 00110100 (binary, 2nd byte: 0 × 34); 00010010 (binary, 3rd byte: 0 × 12). When LSB transfer is set, the transfer sequence is: 01101010 (binary, 1st byte: 0 × 56); 00101100 (binary, 2nd byte: 0 × 34); 01001000 (binary, 3rd byte: 0 × 12).

3.4.3.6 Adjustable Byte Transfer Sequence

The adjustable byte transfer sequence is limited to the priority transfer sequence between different bytes in FIFO. The transfer sequence of bytes in FIFO is set by configuring the cr_spi_byte_inv bit in the register SPI_CONFIG. 0 means sending LSB first, and 1 means sending MSB first.

For example, for data transfer where the frame size is 24-bits, the data format is Data[23:0] = 0 × 123456.

When LSB transfer priority is set, the transfer sequence is 0 × 56 (1st byte: LSB); 0 × 34 (2nd byte: intermediate byte); 0 × 12 (3rd byte: MSB). When MSB transfer priority is set, the transfer sequence is 0 × 12 (3rd byte: MSB); 0×34 (2nd byte: intermediate byte); 0 × 56 (1st byte: LSB).

Adjustable byte transfer can be used in conjunction with configurable MSB/LSB transfer.

3.4.3.7 Slave Mode Timeout Mechanism

When a timeout threshold is set, an interrupt will be triggered when SPI in the slave mode receives no clock signal after the threshold exceeds.

3.4.3.8 DMA Transfer Mode

SPI supports the DMA transfer mode. To enable this mode, you must set the thresholds of TX FIFO and RX FIFO respectively. Setting `spi_dma_tx_en` in the register SPI_FIFO_CONFIG_0 to 1 can enable the DMA sending mode. Setting `spi_dma_rx_en` in the register `SPI_FIFO_CONFIG_0` to 1 can enable the DMA receiving mode. When this mode is enabled, UART will check the TX/RX FIFO. Once the `tx_fifo_cnt/rx_fifo_cnt` in the register SPI_FIFO_CONFIG_1 is greater than `tx_fifo_th/rx_fifo_th`, a DMA request will be initiated, and DMA will transfer data into TX FIFO or remove data from RX FIFO as configured.

3.4.3.9 SPI Interrupt

SPI supports the following interrupt control modes:

- SPI end of transfer interrupt
  - In the master mode, the SPI end of transfer interrupt will be triggered when the transfer of each frame of data ends.
  - In the slave mode, the interrupt is triggered when the CS signal is released.
- TX FIFO request interrupt
  - The TX FIFO request interrupt will be triggered when the FIFO available count value is greater than the preset threshold and the interrupt flag will be cleared automatically when the condition is unmet.
- RX FIFO request interrupt
  - The RX FIFO request interrupt will be triggered when the FIFO available count value is greater than the preset threshold and the interrupt flag will be cleared automatically when the condition is unmet.
- Slave mode transfer timeout interrupt
  - The slave mode transfer timeout interrupt will be triggered when no clock signal is received in the slave mode after the threshold exceeds. If the TX/RX FIFO overflows or underflows, it will trigger the TX/RX FIFO overflow interrupt.
- Slave mode TX overload interrupt
  - Slave mode TX overload interrupt will trigger when the TX is not ready to transmit in slave mode.
- TX/RX FIFO overflow interrupt
  - TX/RX FIFO overflow interrupt is triggered if an overflow or underflow occurs in the TX/RX FIFO. When the `tx_fifo_clr/rx_fifo_clr` bit in the FIFO clear register `SPI_FIFO_CONFIG_0` is set to 1, the corresponding FIFO will be cleared, and the overflow interrupt flag will be cleared automatically.

You can query the interrupt status through the register `SPI_INT_STS` and write 1 to the corresponding bit to clear the interrupt.

3.5 I2C Module

3.5.1 I2C Overview

Inter-Integrated Circuit (I2C) is a serial communication bus, which uses a multi-slave and multi-master architecture and is connected to low-speed peripherals. Each device has a unique address identifier and can be used as a transmitter or receiver. The address of each device connected to the bus can be set by software through a unique address and the existing master or slave relation. The master can work as a master transmitter or a master receiver. If two or more masters are initialized at the same time, data can be prevented from being damaged through conflict detection and arbitration during transmission.

To facilitate communication with slaves with the FIFO of 2-word depth and interrupt function, it can be used with DMA to improve efficiency and supports flexible adjustment of clock frequency.

3.5.2 I2C Features

- Master mode
- Multi-master mode and arbitration function
- Flexible control of the level duration of the start, end, and data transmission phases in segments
- Supports 7-bit address mode and 10-bit address mode
- Supports DMA transfer mode
- Supports multiple interrupt mechanisms

3.5.3 I2C Functional Description

Table 11 lists the I2C pins.

Table 11 I2C Pin List

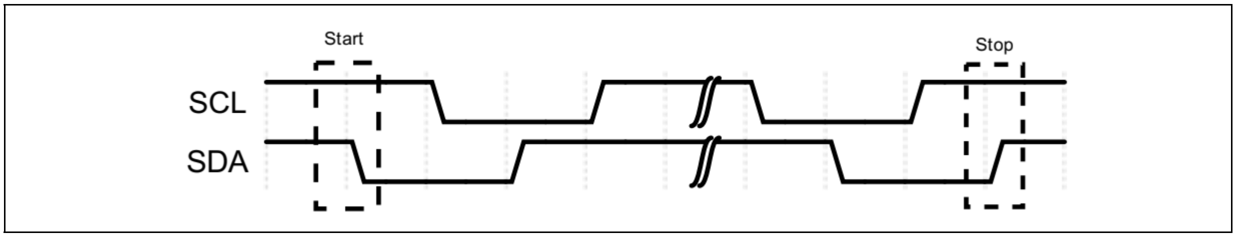
Name	Type	Description
I2Cx_SCL	Input/output	I2C serial clock signal
I2Cx_SDA	Input/output	I2C serial data signal

3.5.3.1 Start and Stop Conditions

All transmissions start with a START condition and end with a STOP condition. The START and STOP conditions are generally generated by the master. The bus is busy after the START condition and to be idle for a certain period after the STOP condition.

- START condition: SDA produces a high-to-low level transition when SCL is high
- STOP condition: SDA produces a low-to-high level transition when SCL is high
Figure 13 shows the timing diagram.

Figure 13 I2C Start and Stop Waveform Diagram



3.5.3.2 Data Transfer Format

3.5.3.2.1 7-bit Address Mode

The first 8-bits transferred are addressing bytes, including a 7-bit slave address and a 1-bit direction bit. Sending or receiving data by the master is controlled by the 8th bit in the first byte sent by the master. If it is 0, it means that the data is sent by the master, while "1" indicates that data is received by the master. After the direction bit is the answer bit (ACK), which is sent by the slave to answer (pull the signal low) and the host starts transmitting the specified length of data after receiving the answer. Once the data transfer is complete, the master sends out a STOP signal, with waveform shown as in Figure 14 to Figure 17.

Figure 14 Master Transmit and Slave Receive Data Formats

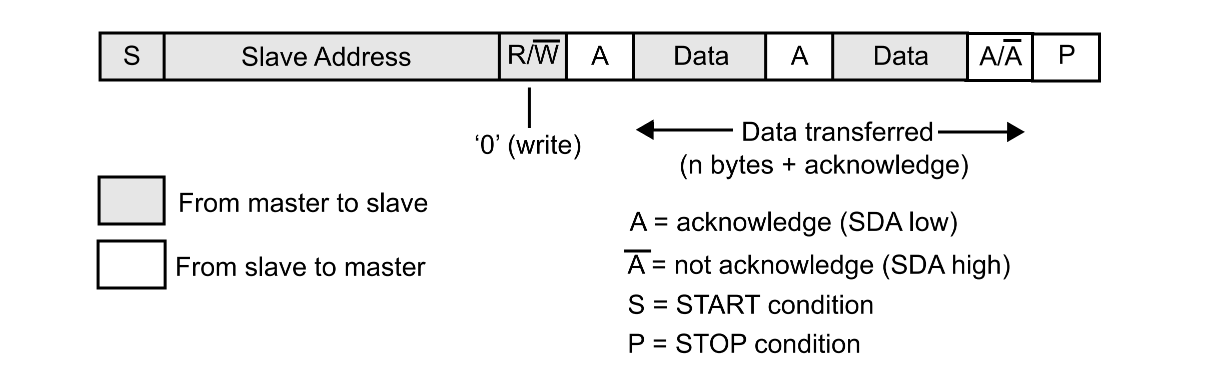


Figure 15 Master Receive and Slave Transmit Data Formats

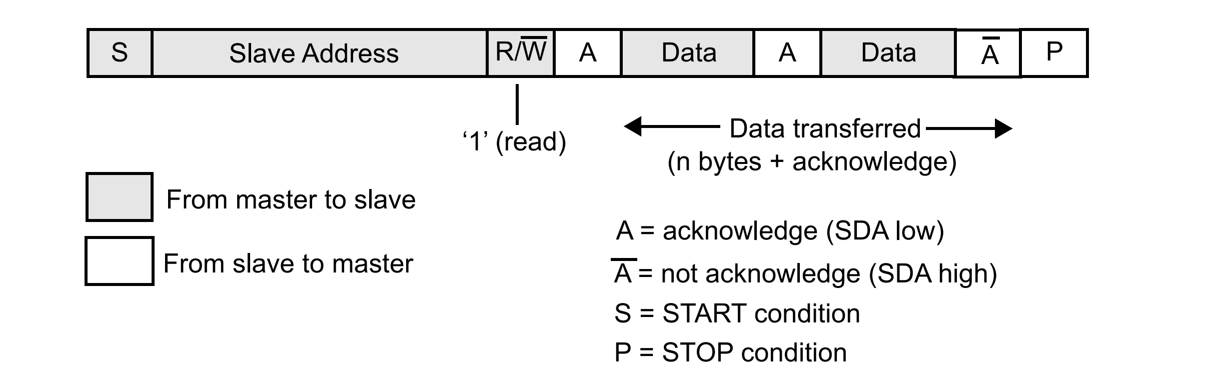


Figure 16 Timing of Master Transmitter and Slave Receiver

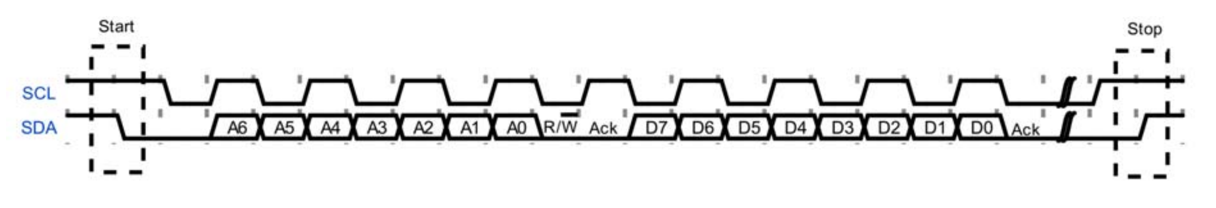
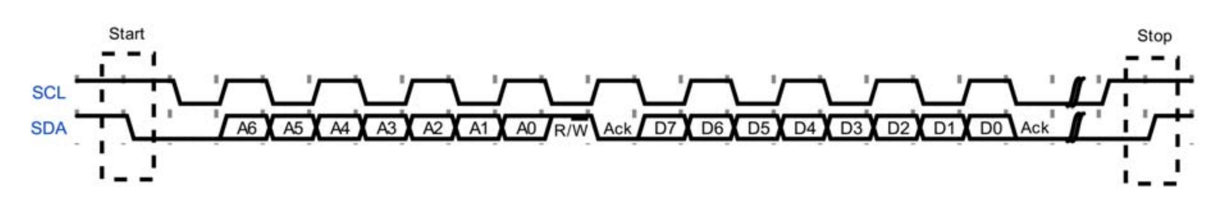


Figure 17 Timing of Master Receiver and Slave Transmitter

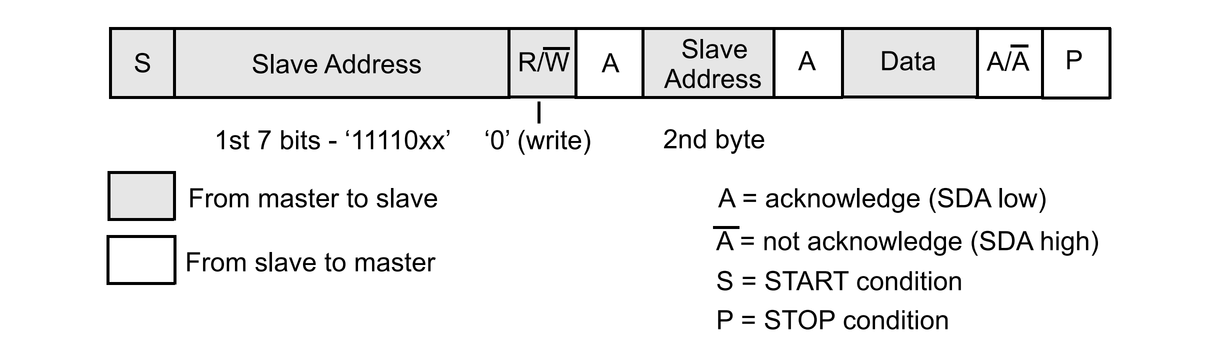


3.5.3.2.2 10-bit Address Mode

The `cr_i2c_10b_addr_en` in the register `I2C_CONFIG` must be set to 1 before use.

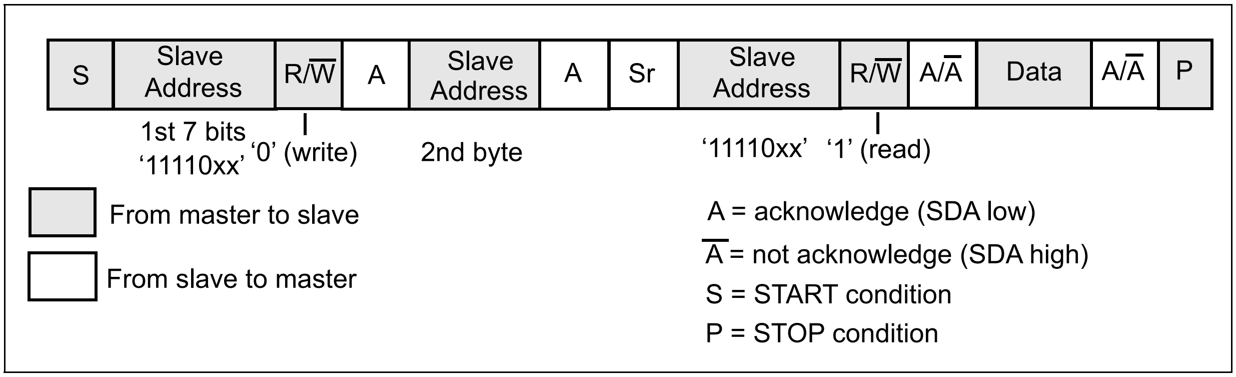
The 10-bit slave address consists of the two bytes after the START condition (S) or the repeated START condition (Sr). The first 7-bits of the first byte are 1111 0XX, where XX are the first two bits of MSB of the 10-bit address. The 8th bit of the first byte is the read/write bit that determines the transfer direction. The second byte is the remaining low 8-bits of the 10-bit address. The data transfer format is as shown in Figure 18.

Figure 18 Master Transmit and Slave Receive Data Format (10-bit slave address)



When receiving the 10-bit address following the START condition, the slave compares the first byte (1111 0XX) of the slave address with its own address and checks whether the eighth bit (read/write bit) is 0. If the value of XX in the first byte is the same as the top two bits of the slave's 10-bit address, the first byte match passes and the slave will give answer A. If there are multiple slave devices connected to the bus, more than one device may match and generate answer A. Next, all slaves start to match the second byte (XXXX XXXX), where only one slave will have the exact same lower eight bits of the 10-bit address as the second byte, and that slave will give answer A. The slave that is addressed by the master will remain addressed until it receives a termination condition, or a repeat start condition.

Figure 19 Master Receive and Slave Transmit Data Format (10-bit slave address)

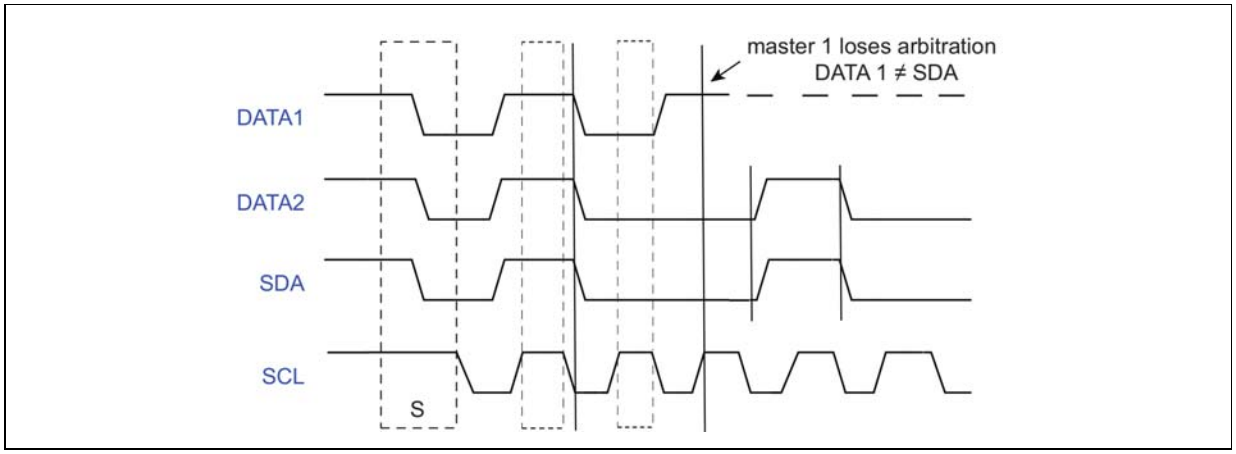


Before the second acknowledgment A, the process is same as that of the master-transmitter addressing the slave-receiver. After the repeated START condition (Sr), the matched slave will remain in the addressed state. This slave will check whether the first 7-bits of the first byte after Sr are 1111 0XX, and then test whether the 8th bit is 1 (read). If this also matches, the slave considers that it is addressed as a transmitter and generates an acknowledgment (A). The slave-transmitter will remain in the addressed state until it receives the STOP condition (P) or the repeated START condition (Sr) followed by a different slave address. Then, under Sr, all the slaves will compare their addresses with 11110XX and test the eighth bit (read/write bit). However, they will not be addressed, because for 10-bit devices, the read/write bit is 1, or for 7-bit devices, the slave addresses of 1111 0XX do not match

3.5.4 Arbitration

When there are multiple masters on I2C bus, it may happen that multiple masters start data transfer at the same time. At this time, the arbitration mechanism will decide which master has the right to transfer data, while other masters must give up control of the bus and wait until the bus is idle before transferring data again. During data transfer, all masters must check whether the SDA is consistent with the data they want to send when SCL stays high. When the SDA level is different from the expected one, it means that other masters are transferring data at the same time. The masters with different SDA levels will lose the arbitration and other masters will complete the data transfer. The waveform of two masters transferring data and initiating the arbitration mechanism at the same time is as shown in Figure 20.

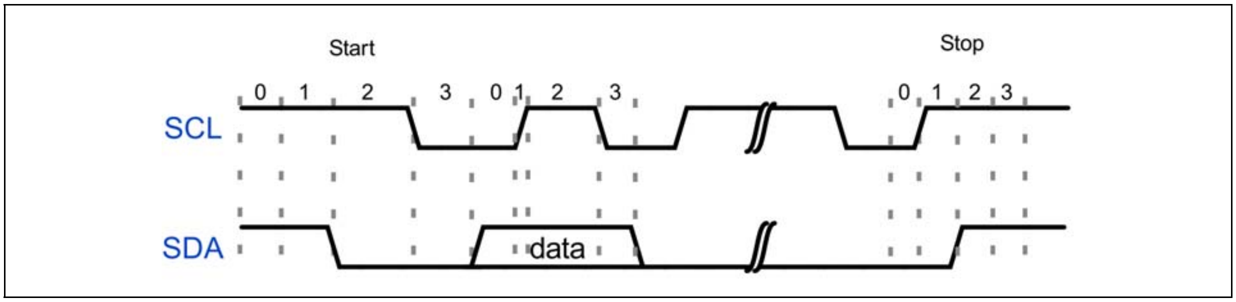
Figure 20 Waveform of Simultaneous Data Transfer



3.5.5 I2C Clock Setting

I2C clock can be derived from `bclk` (bus clock) and `xclk`, and frequency division can be done on this basis. The duration of the start condition, each bit of data and the end condition are set by registers `I2C_PRD_START, I2C_PRD_DATA` and `I2C_PRD_STOP` respectively. Each of these durations can be subdivided into four phases, and the number of samples in each phase is controlled by a separate byte in the register (the actual value is the register value plus 1). The four phase settings in the data section together determine the frequency division factor of the I2C clock. As shown in the Figure 21, suppose the I2C clock source is selected as 80M bclk and the register `I2C_PRD_DATA` is set to 0x09070b09, then the second 0 in the Figure 21 is 0x09 + 1 = 0x0a, the second 1 is 0x07 + 1 = 0x08, the second 2 is 0x0b + 1 = 0x0c, and the second 3 is 0x09 + 1 = 0x0a. Then the clock frequency of I2C is 80 MHz/(0x0a + 0x08 + 0x0c + 0x0a) = 2 MHz. Similarly, the first 0, 1, 2 and 3 are set by register `I2C_PRD_START`, which determines the duration of the start condition, and the third 0, 1, 2, and 3 are set by register `I2C_PRD_STOP`, which determines the duration of the end condition.

Figure 21 I2C Clock Setting



3.5.6 I2C Configuration Flow

Configuration Items:

- Read/write flag bit
- Slave address
- Slave register address
- Slave register address length
- Data (TX: configure the sent data; RX: store the received data)

- Data length
- Enable signal

3.5.6.1 Read / Write Flag Bit

I2C supports TX and RX working statuses. The `cr_i2c_pkt_dir` in the register `I2C_CONFIG` represents the TX/RX status, 0 for TX status and 1 for RX status.

3.5.6.2 Slave Address

Each slave connected to I2C will have a unique device address, which is usually 7-bits long. This address will be written into the cr_i2c_slv_addr in the register `I2C_CONFIG`. I2C will automatically shift to the left by 1 bit before sending the address, and the TX/RX direction bit will be added to the LSB.

3.5.6.3 Slave Register Address

The slave register address represents the register address where I2C must read and write a slave register. The slave register address is written to the register `I2C_SUB_ADDR` and the `cr_i2c_sub_addr_en` in the register `I2C_CONFIG` must be set to 1. If `cr_i2c_sub_addr_en` in the register `I2C_CONFIG` is set to 0, the I2C master will skip the slave register address field when sending.

3.5.6.4 Slave Register Address Length

The slave register address length is subtracted by 1 and then written to `cr_i2c_sub_addr_bc` in the register `I2C_CONFIG`.

3.5.6.5 Data

It refers to the data that must be sent to or received from the slave. When sending data, I2C must write the data (in word) into the register `I2C_FIFO_WDATA`. When receiving data, I2C must read out the data (in word) from the register `I2C_FIFO_RDATA`.

3.5.6.6 Data Length

The `cr_i2c_pkt_len` in the register `I2C_CONFIG` sets the send data length (the value written to the register + 1 is the send data length) and the maximum send length is 256-bytes.

3.5.6.7 Enable Signal

After the above items are configured, when cr_i2c_m_en in the enable signal register `I2C_CONFIG` is set to 1, the I2C sending process will be started automatically.

When the read/write flag bit is configured as 0, I2C sends data. For example, consider sending 2 bytes, the master's transmission flow is as follows:

1. Start bit
2. (The slave address shifts to the left by 1-bit + 0) + ACK
3. Slave register address + ACK
4. 1-byte data + ACK
5. 1-byte data + ACK
6. Stop bit

When the read/write flag bit is configured as 1, I2C receives data. For example, consider receiving 2 bytes, the master's transmission flow is as follows:

1. Start bit

2. (The slave address shifts to the left by 1-bit + 0) + ACK
3. Slave register address + ACK
4. Start bit
5. (The slave address shifts to the left by 1-bit + 1) + ACK
6. 1-byte data + ACK
7. 1-byte data + ACK
8. Stop bit

3.5.7 FIFO Management

I2C FIFO has a 2-word depth, and I2C includes RX FIFO and TX FIFO. The `rx_fifo_cnt` in the register `I2C_FIFO_CONFIG_1` represents how much data (in word) in RX FIFO needs to be read. The `tx_fifo_cnt` in the register `I2C_FIFO_CONFIG_1` represents how much free space (in word) in TX FIFO for writing.

I2C FIFO status:

- RX FIFO underflow: When the data in RX FIFO is completely read out or empty, if I2C continues to read data from RX FIFO, the `rx_fifo_underflow` in the register `I2C_FIFO_CONFIG_0` will be set to 1
- RX FIFO overflow: When I2C receives data until the two words of RX FIFO are filled without reading RX FIFO, if I2C receives data again, the `rx_fifo_overflow` in the register `I2C_FIFO_CONFIG_0` will be set to 1
- TX FIFO underflow: When the data size filled into TX FIFO does not meet the configured I2C data length cr_i2c_pkt_len in register I2C_CONFIG and no new data is filled into TX FIFO, the `tx_fifo_underflow` in the register `I2C_FIFO_CONFIG_0` will be set to 1
- TX FIFO overflow: After the two words of TX FIFO are filled before the data in TX FIFO is sent out, if data is filled into TX FIFO again, the `tx_fifo_overflow` in the register `I2C_FIFO_CONFIG_0` will be set to 1.

3.5.8 Use with DMA

I2C can send and receive data through DMA. Setting `i2c_dma_tx_en` in the register `I2C_FIFO_CONFIG_0` to 1 will enable the DMA TX mode. After the channel for I2C is allocated, DMA will transfer data from memory to the `I2C_FIFO_WDATA` register. Setting i2c_dma_rx_en in the register `I2C_FIFO_CONFIG_0` to 1 will enable the DMA RX mode. After the channel for I2C is allocated, DMA will transfer the data in the register I2C_FIFO_RDATA to memory. When I2C is used with DMA, DMA will automatically transfer data, so it is unnecessary for CPU to write data into I2C TX FIFO or read data from I2C RX FIFO.

3.5.8.1 DMA Sending Flow

1. Set read/write flag bit to 0
2. Set slave address
3. Set slave register address
4. Set slave register address length
5. Data length
6. Set enable signal register to 1
7. Configure DMA transfer size
8. Configure the transfer width of DMA source address

9. Configure the transfer width of DMA destination address (when I2C is used with DMA, the transfer width of destination address must be set to 32-bits, which is word-aligned)
10. Configure the DMA source address as the memory address for storing sent data
11. Configure the DMA destination address to I2C TX FIFO address, `i2c_fifo_wdata`
12. Enable DMA

3.5.8.2 DMA Receiving Flow

1. Set read/write flag bit to 1
2. Set slave address
3. Set slave register address
4. Set slave register address length
5. Data length
6. Set enable signal register to 1
7. Configure DMA transfer size
8. Configure the transfer width of DMA source address (when I2C is used with DMA, the transfer width of source address must be set to 32-bits, which is word-aligned)
9. Configure the transfer width of DMA destination address
10. Configure the DMA source address to I2C RX FIFO address, `i2c_fifo_rdata`
11. Configure the DMA destination address as the memory address for storing received data
12. Enable DMA

3.5.9 I2C Interrupt

I2C includes the following interrupts:

- `I2C_TRANS_END_INT:` I2C transfer end interrupt, which is generated when I2C completes a transfer.
- `I2C_TX_FIFO_READY_INT:` When `tx_fifo_cnt` in `I2C_FIFO_CONFIG_1` is greater than `tx_fifo_th`, a TX FIFO request interrupt will be generated, and the interrupt flag will be automatically cleared when the condition is not satisfied.
- `I2C_RX_FIFO_READY_INT:` When `rx_fifo_cnt` in `I2C_FIFO_CONFIG_1` is greater than `rx_fifo_th`, an RX FIFO request interrupt will be generated, and the interrupt flag will be automatically cleared when the condition is not satisfied.
- `I2C_NACK_RECV_INT:` When the I2C module detects a NACK state, a NACK interrupt is generated.
- `I2C_ARB_LOST_INT:` I2C arbitration lost interrupt.
- `I2C_FIFO_ERR_INT:` FIFO error interrupt is generated when TX/RX FIFO overflows or underflows.

3.6 TIMER Module

3.6.1 Timer Overview

The chip has two 32-bit counters, each of which can independently control and configure its parameters and clock frequency. There is a watchdog counter in the chip. Unpredictable software or hardware behavior may cause the application to malfunction. A watchdog timer can help the system recover from it. If the current time exceeds the predetermined time, but the dog is not fed or closed Timer, which can trigger interrupt or system reset according to the setting.

Figure 22 BL TIMER

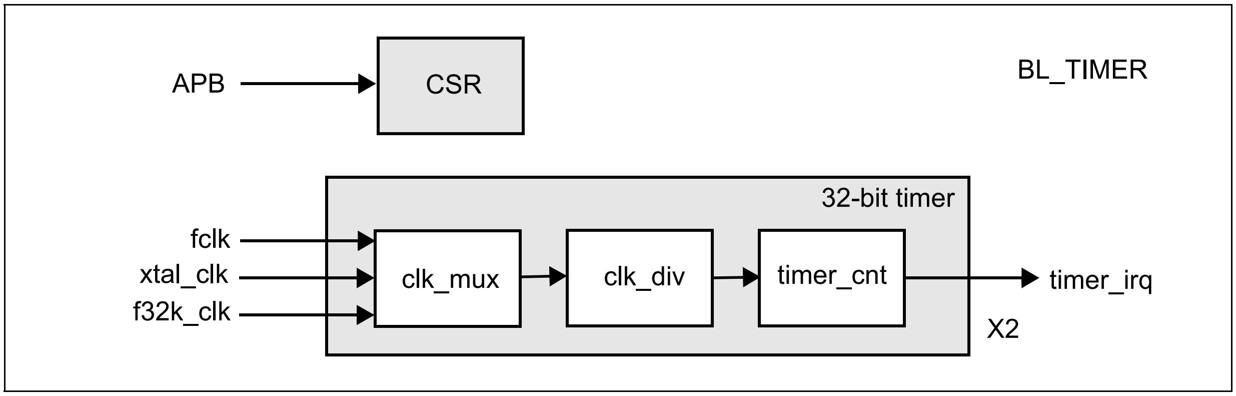
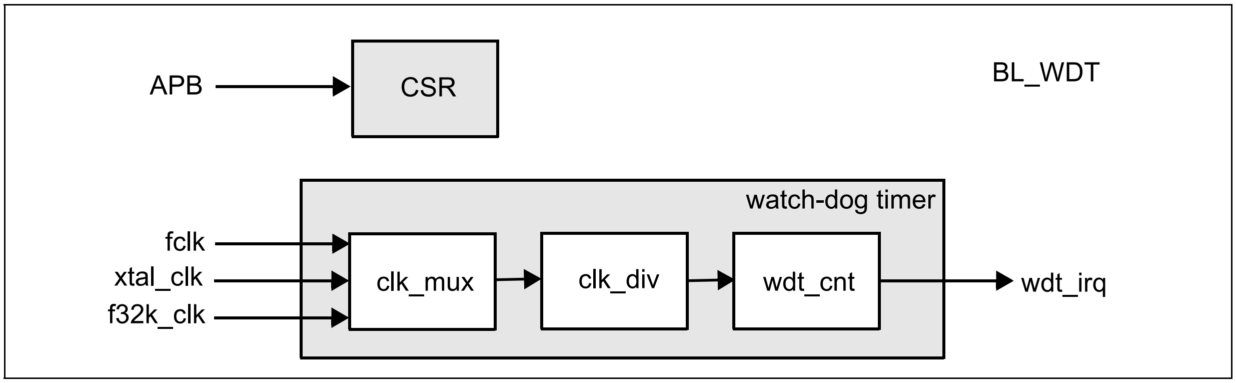


Figure 23 BL WDT



3.6.2 Timer Features

- Multiple clock source options
- 8-bit clock divider with a division factor of 1-256.
- Two 32-bit timers
- Each timer contains three alarm value settings, which can be set independently to alarm when each alarm value overflows
- Support FreeRun mode and PreLoad mode
- 16-bit watchdog timer
- Supports write password protection to prevent system abnormalities caused by incorrect settings
- Support two watchdog overflow methods: interrupt or reset

3.6.3 TIMER function description

8-bit Divider

There are three types of Watchdog timer clocks:

- Fclk--System master clock
- 32K--32K clock
- Xtal--External crystal

There are four timer clock sources:

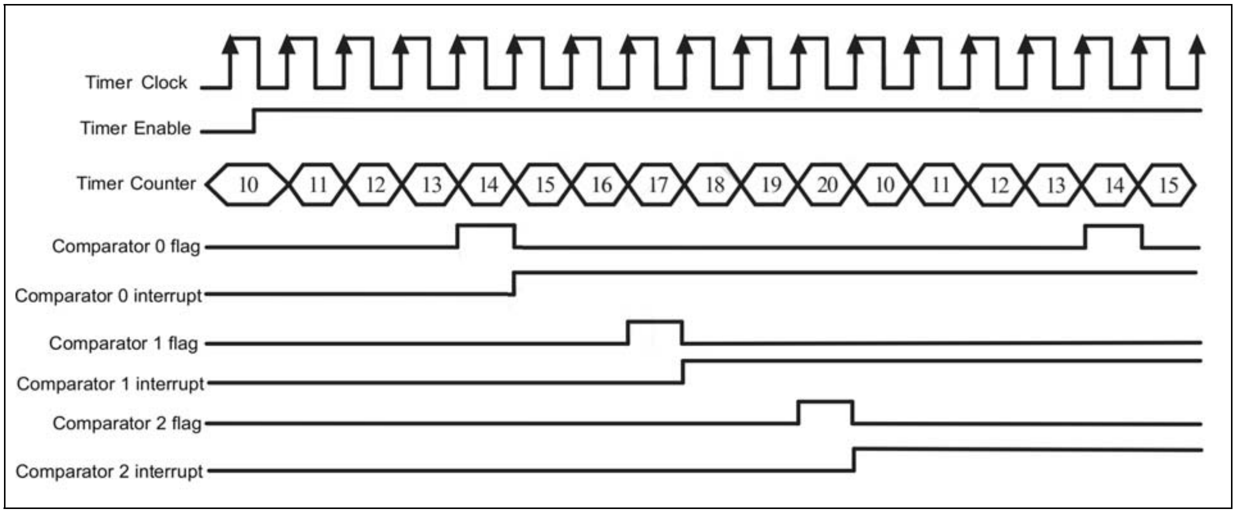
- Fclk--System master clock
- 32K--32K clock
- 1K--1K clock?32K frequency division?
- Xtal--External crystal

Each counter has its own 8-bit frequency divider. The selected clock can be divided by 1-256 through APB. Specifically, when it is set to 0, it means no frequency division, and when it is set to 1, it divides it by 2. The maximum frequency division coefficient is 256, the counter will use the divided clock as the unit of the counting cycle, each time a counting cycle is increased by one.

3.6.4 General timer operating mode

Each general-purpose timer includes three comparators, a counter and a preload register. When the clock source is set and the timer is started, the counter starts to count. When the counter value is equal to the comparator, the comparison is performed. When the flag is set, a compare interrupt is generated. The initial value of the counter depends on the timing mode. In FreeRun mode, the initial value of the counter is 0, and then counts. When it reaches the maximum value, it starts counting from 0 again.

Figure 24 PRELOAD



In PreLoad mode, the initial value of the counter is the value of the PreLoad register and then counts. When the PreLoad condition is met, the value of the counter is set to the value of the PreLoad register, and then the counter starts to count again. During the counting process, once the value of the counter matches one of the three comparators, the comparator's comparison flag will be set and a corresponding comparison interrupt can be generated.

If the value of the preload register is 10, the value of Comparator 0 is 13, the value of Comparator 1 is 16, and the value of Comparator 2 is 19, the working sequence of the timer in PreLoad mode is as follows:

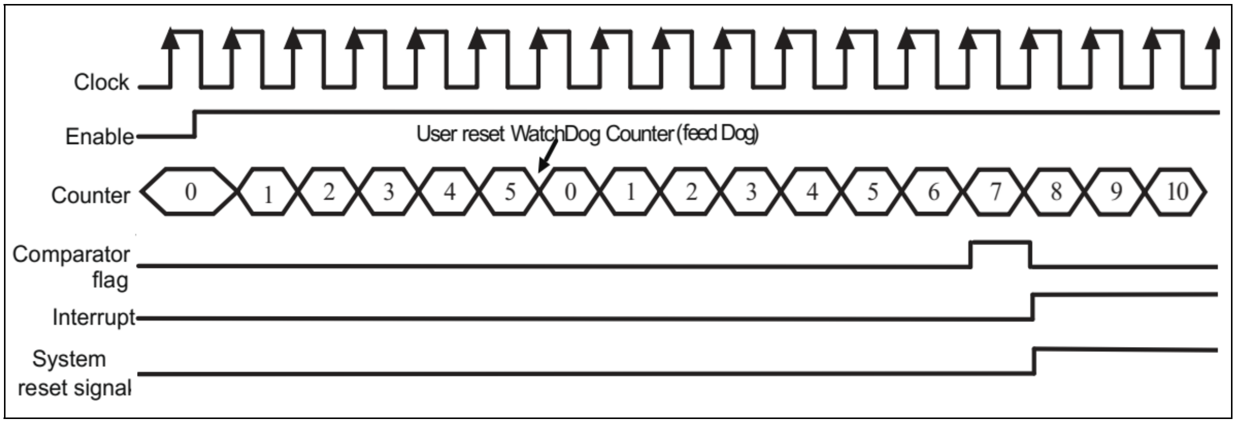
In FreeRun mode, the timer working sequence is basically the same as PreLoad, the difference is that the counter will start to accumulate from 0 to the maximum value. The mechanism of the generated compare flags and compare interrupts is the same as in PreLoad mode.

3.6.5 Watchdog timer operating mode

The watchdog timer includes a counter and a comparator. The counter counts from 0. If the counter is reset (feed the dog), it starts counting from 0 again. When the counter value is equal to the comparator, a comparison interrupt signal or a system reset signal will be generated, and the user can choose to use one of them as required.

The watchdog counter is incremented by one in each counting cycle unit. Software can reset the watchdog counter to zero at any point in time through the APB. If the value of the comparator is 6, the working sequence of Watchdog is shown in Figure 25.

Figure 25 WATCHDOG COUNTER



3.6.6 Alarm setting

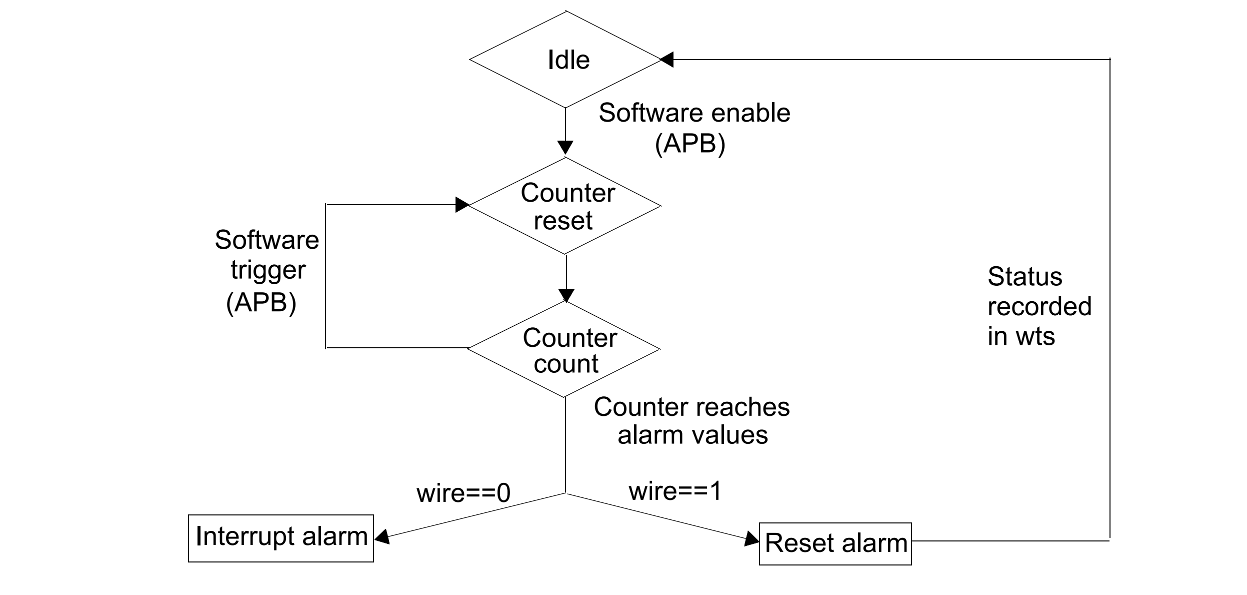
Each counter has three comparison values, and can set whether each comparison value triggers an alarm interrupt. When the counter matches the comparison value and the setting will alarm, the counter will notify the processor through the interrupt.

The software can read through the APB whether an alarm has occurred and which comparison value triggered the alarm interrupt. When the alarm interrupt is cleared, the alarm status is also cleared simultaneously.

3.6.7 Watchdog Alarm

A comparison value can be set for each counter. When the software fails to reset the watchdog counter to zero due to a system error, which causes the watchdog counter to exceed the comparison value, a watchdog alarm is triggered. There are two types of alarms. The first is to perform necessary actions through interrupt notification software. The second is to enter the system watchdog reset. When the watchdog reset is triggered, it will notify the system reset controller and prepare for system reset. When everything is ready, enter the system watchdog reset. It is worth noting that software can read the WSR register through APB to know if a watchdog system reset has occurred.

Figure 26 WATCHDOG ALARM



3.7 UART Module

3.7.1 UART Overview

The Universal Asynchronous Receiver/Transmitter (UART) provides a flexible way to exchange full-duplex data with external devices. µC subsystem is provided with 1 UARTs, which can be used together with DMA to achieve efficient data communication.

3.7.2 UART Features

- Full-duplex asynchronous communication
- Optional data bit length: 5/6/7/8-bit
- Optional stop bit length: 0.5/1/1.5/2-bit
- Supports odd/even/none check bit
- Error-detectable start bit
- Abundant interrupt control modes
- Hardware flow control (RTS/CTS)
- Convenient baud rate programming
- Configurable MSB/LSB transfer priority
- Automatic baud rate detection of ordinary/fixed characters
- 32-byte TX/RX FIFO
- Supports DMA transfer mode
- Supports baud rate of 10 Mbps and below
- Supports the LIN bus protocol
- Supports the RS485 mode
- Optional clock sources: 160M/BCLK/XCLk
- Support filter function

3.7.3 UART Functional Description

3.7.3.1 Data Formats

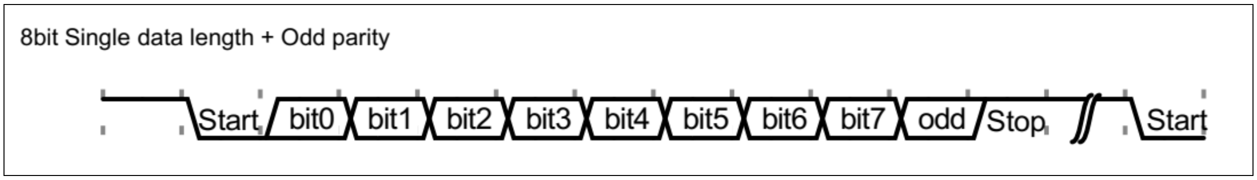
The normal UART communication data consists of start bits, data bits, parity check bits, and stop bits. The UART supports configurable data bits, parity check bits, and stop bits, which are set in registers `UTX_CONFIG` and `URX_CONFIG`. The waveform of a frame of data is as shown below in Figure 27 The start bit of the data frame occupies 1-bit, and the stop bit can be 0.5/1/1.5/2-bit wide by configuring the `cr_utx_bit_cnt_p` bit in the register `UTX_CONFIG`. The start bit is at a low level and the stop bit is at a high level. The data bit width can be set to 5/6/7/8-bit by the `cr_utx_bit_cnt_d` bit in the register `UTX_CONFIG`.

When the `cr_utx_prt_en` bit in the register `UTX_CONFIG` and the cr_urx_prt_en bit in the register `URX_CONFIG` is set, the data frame adds a parity check bit after the data. The `cr_utx_prt_sel` bit in the register `UTX_CONFIG` and the cr_urx_prt_sel bit in the register `URX_CONFIG` is used to select odd or even parity check. When the receiver detects the check bit error of the input data, it will generate the check error interrupt. However, the received data will still be stored into the FIFO.

Figure 27 UART Data Format

Calculation method of odd parity check: If there is an odd number of "1" in the current data bit, the odd parity check bit is set to 0. Otherwise, it is set to 1.

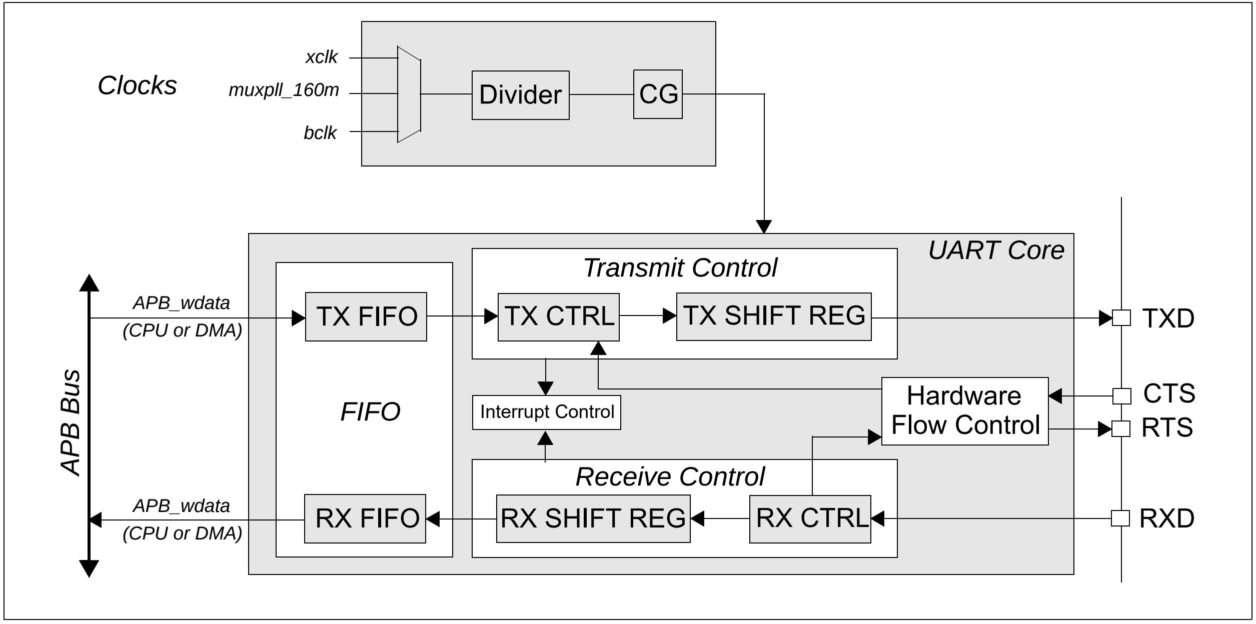
Calculation method of even parity check: If there is an odd number of "1" in the current data bit, the even parity check bit is set to 1. Otherwise, it is set to 0.



3.7.3.2 Basic Architecture

The UART has 3 clock sources: XCLK, 200 MHz CLK, and BCLK. The frequency divider in the clock is used to divide the frequency of the clock source and then generate the clock signal to drive the UART module. The UART controller is divided into two functional blocks: transmitter and receiver as shown in the Figure 28.

Figure 28 UART Data Architecture



3.7.3.3 Transmitter

The transmitter contains a 32-byte TX FIFO to store the data to be sent. When the transmission enable bit is set, the data stored in FIFO will be output from the TX pin. Software can transfer data into TX FIFO through DMA or APB bus. Software can check the status of transmitter by querying the remaining free space count value of `TX FIFO` through `tx_fifo_cnt` in the register `UART_FIFO_CONFIG_1`.

FreeRun mode of transmitter:

- If the FreeRun mode is disabled, transmission will be terminated, and an interrupt will be generated when the sent bytes reach the specified length. Before next transmission, you must re-disable and enable the `TxE` bit.
- If the FreeRun mode is enabled, the transmitter will send when there is data in the TX FIFO and will not stop working because the sent bytes reach the specified length.

3.7.3.4 Receiver

The receiver contains a 32-byte RX FIFO to store the received data. Software can check the status of receiver by querying the available data count value of RX FIFO through `rx_fifo_cnt` in the register `UART_FIFO_CONFIG_1`. The low 8-bits of the register `URX_RTO_TIMER` are used to set a receiving timeout threshold, which will trigger an interrupt when the receiver fails to receive data beyond the threshold. The `cr_urx_deg_en` and `cr_urx_deg_cnt` in the register `URX_CONFIG` is used to enable the deburring function and set the threshold, which controls the filtering part before sampling by UART. UART will filter out the glitches whose width is lower than the threshold in the waveform and then sends it to sampling.

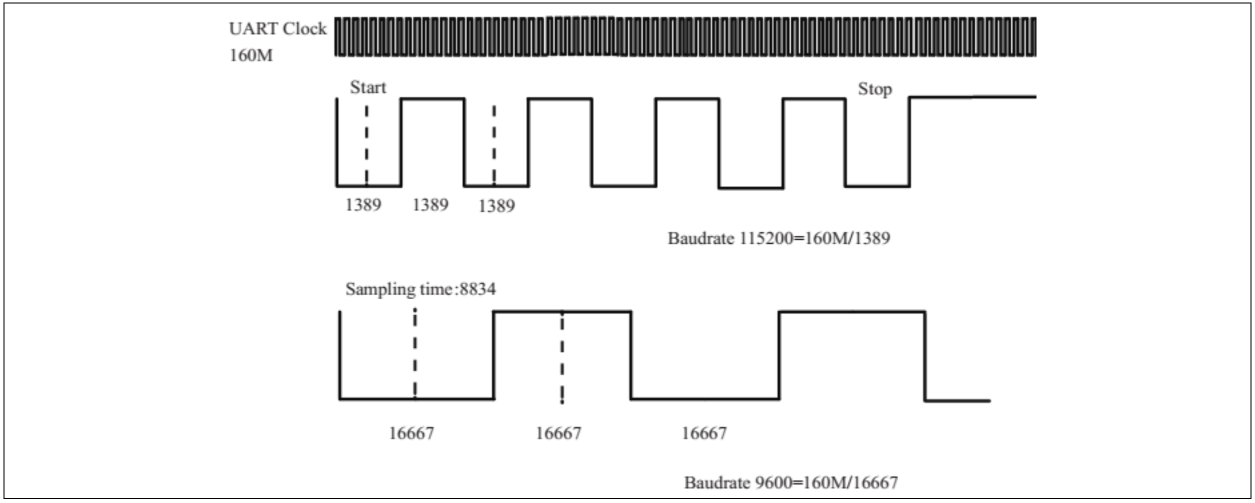
3.7.3.5 Baud Rate Setting

Baudrate = \frac{UART_clk}{uart_prd + 1}

User can set the baud rate of RX and TX separately. For example, consider TX the value of uart_prd is the value of the lower 16-bits `cr_utx_bit_prd` of the register `UART_BIT_PRD`. Since the maximum value of the 16-bit bit width coefficient is 65535, the minimum baud rate supported by UART is `UART_clk/65536`.

Before sampling the data, UART will filter the data to remove the glitches in the waveform. Then, the data will be sampled at the intermediate value of the 16-bit width factor, so that the sampling time can be adjusted based on baud rates to ensure that the intermediate value is always sampled providing much higher flexibility and accuracy. The sampling process is as shown in Figure 29.

Figure 29 UART Sampling Waveform



3.7.3.6 Filtering

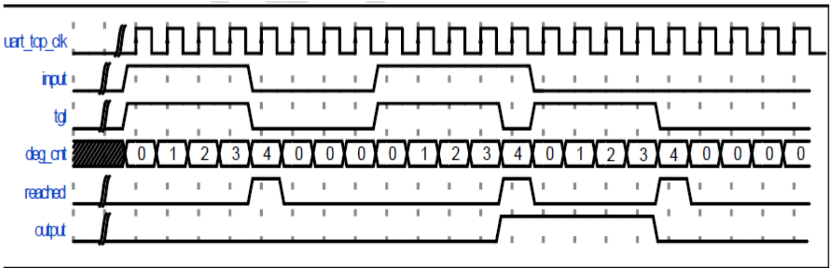
Figure 30 illustrates the UART filter waveform. When this function is enabled by configuring `cr_urx_deg_en` and the threshold is set by configuring `cr_urx_deg_cnt` in the register `URX_CONFIG`, UART will filter out the data that cannot meet the width threshold. As shown in Figure 30, when the data width is 4, setting `cr_urx_deg_cnt` to 4 can meet this condition. Input is the initial data and output is the filtered data.

Filtering logic process:

- `Tgl` is the exclusive XOR result of input and output.
- `Deg_cnt` counts from 0 and the counting condition is that `tgl` is at a high level and reached is at a low level.
- If the count value of deg_cnt reaches the value set by `cr_urx_deg_cnt`, reached is at a high level.
- When reached is at a high level, input is output to output.

Note: User-defined condition for `deg_cnt: tgl` is at a high level and reached is at a low level. In other cases, `deg_cnt` will be cleared to 0.

Figure 30 UART Filter Waveform



3.7.3.7 Automatic Baud Rate Detection

The UART module supports automatic baud rate detection, which is divided into two modes, a generic mode, and a fixed character (square wave) mode. The `cr_urx_abr_en` in the `URX_CONFIG` register enables auto baud rate detection, and when it is turned on, both detection modes are enabled.

3.7.3.7.1 Generic mode

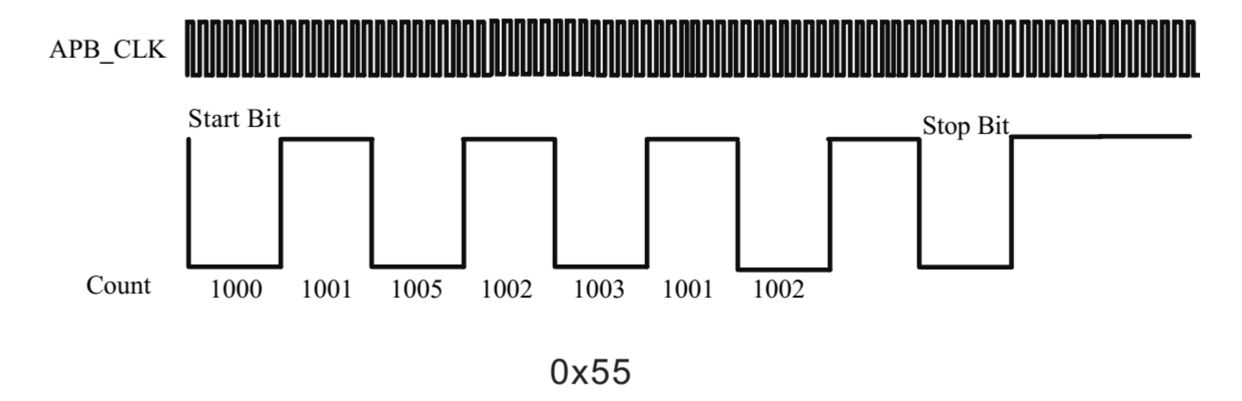
For any character data received, UART will count the number of clocks in the start bit width, which will then be written into the low 16-bits `sts_urx_abr_prd_start` in the register `STS_URX_ABR_PRD` and used to calculate the baud rate. So, the correct baud rate can be obtained when the first received data bit is 1, such as '0x01' under LSBFIRST.

3.7.3.7.2 Fixed character (square wave) mode

In this mode, after the UART module counts the number of clocks in the bit width, it will continue to count the number of clocks in the subsequent data bits and compare it with the start bit. The check is passed, otherwise the count value is discarded. The allowable error can be set by setting the `cr_urx_abr_pw_tol` bit in the register `URX_ABR_PW_TOL`, and the unit is the clock source of the UART.

Therefore, only when the fixed character '0x55'/'0xD5' under LSB-FIRST or '0xAA'/'0xAB' under MSB-FIRST is received, the UART module will write the clock count value in the starting bit width into the high 16-bit `sts_urx_abr_prd_0x55` of the register `STS_URX_ABR_PRD` as shown in Figure 31.

Figure 31 Waveform of UART in Fixed Character Mode



As shown above, assuming the maximum allowable error set is 4, for a received data with unknown baud rate, the UART uses UART_CLK to count the bit width of the starting bit as 1000, the bit width of the second bit as 1001, which is not more than 4 UART_CLK up or down from the previous bit width, then the UART will continue to count the third bit. The third bit is 1005, the difference with the starting bit is more than 4, the detection is not passed and the data is discarded. UART compares the first 6-bit widths of the data bits with the starting bit in turn.

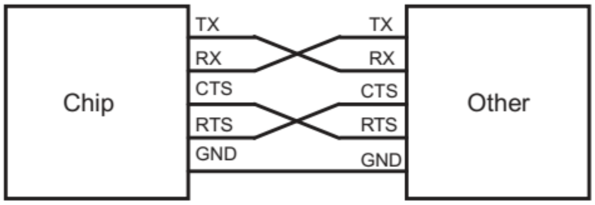
Formula for calculating the detected baud rate:

Baudrate = \frac{UART_clk}{Count + 1}

3.7.3.8 Hardware Flow Control

UART supports hardware flow control in CTS/RTS mode to prevent data in FIFO from being lost due to too late processing. Hardware flow control connection is as shown in Figure 32.

Figure 32 UART Hardware Flow Control



Require To Send (RTS) is an output signal, which indicates whether the chip is ready to receive data from the other side. This is valid at a low level denoting that the chip can receive data.

Clear To Send (CTS) is an input signal, which determines whether the chip can send data to the other side. This is valid at a low level denoting that the chip can send data to the other side.

When the hardware flow control function is enabled, the low level of chip's RTS indicates requesting the other side to send data and the high level of that indicates informing the other side to stop sending data. When the chip detects that CTS goes high, TX will stop sending data and continue sending until CTS goes low. If CTS goes high or low at any time during communication, it does not affect the continuity of data sent by TX and the other side also can receive continuous data.

Following are the two ways for hardware flow control of the transmitter.

- Hardware control (the `cr_urx_rts_sw_mode` in the register `UART_SW_MODE` is 0): RTS goes high when `cr_urx_en` in the register `URX_CONFIG` is not turned on or the RX FIFO is almost full (one byte left).
- Software control (the `cr_urx_rts_sw_mode` in the register `UART_SW_MODE` is 1): The level of RTS can be changed by configuring `cr_urx_rts_sw_val` in the register `UART_SW_MODE`.

3.7.3.9 DMA Transfer

UART supports DMA transfer. Using DMA transfer, the TX and RX FIFO thresholds must be set respectively by `tx_fifo_th` and `rx_fifo_th` in register `UART_FIFO_CONFIG_1`. When this mode is enabled, if `tx_fifo_cnt` in `uart_fifo_config_1` is greater than `tx_fifo_th`, a DMA TX request will be triggered. After the DMA is configured, when the DMA receives the request, it will move the data from the memory to the TX FIFO according to the settings. If the `rx_fifo_cnt` in `uart_fifo_config_1` is greater than `rx_fifo_th`, the DMA RX request will be triggered. After the DMA is configured, when the DMA receives the request, it will transfer the data of the RX FIFO to the memory according to the settings.

To ensure the correctness of the data transferred by the chip DMA TX channel, the following conditions need to be met in the channel configuration.

```
(transferWidth * burstSize) ≤ (tx_fifo_th + 1)
```

To ensure the integrity of the data transferred by the chip DMA RX channel, the following conditions need to be met in the channel configuration.

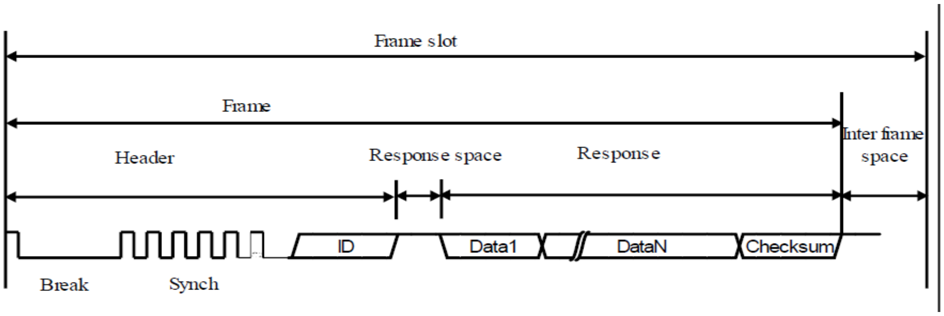
```
(transferWidth * burstSize) = (rx_fifo_th + 1)
```

3.7.3.10 Support for LIN Bus

The protocol for the Local Interconnect Network (LIN) is based on the Volcano-Lite technology developed by the Volvo spin-out company -Volcano Communications Technology (VCT). LIN is a complementary protocol to CAN and SAE J1850, suitable for applications that have low requirement for time or require no precise fault tolerance (as LIN is not as reliable as CAN). LIN aims to be easy to use as a low-cost alternative to CAN. The vehicle parts where LIN can be used include window regulator, rear view mirror, wiper, and rain sensor.

UART supports the LIN bus mode. The LIN bus is under the master-slave mode, and data is always initiated by the master node. The frame (header) sent by the master node contains synchronization interval field, synchronization byte field, and identifier field. A typical LIN data transfer is as shown in Figure 33.

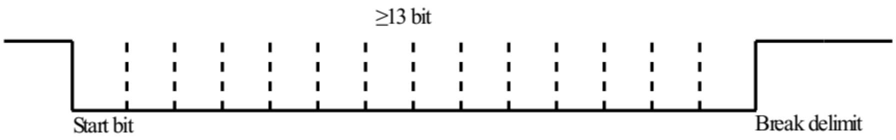
Figure 33 Typical LIN Frame



3.7.3.10.1 LIN break field

Figure 34 illustrates timing diagram for break field of LIN.

Figure 34 Break Field of LIN



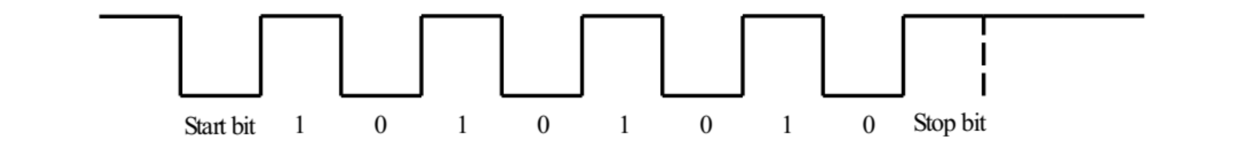
The synchronization interval field indicates the start of the message with at least 13 dominant bits (including the start bit). The synchronization interval ends with an “interval separator”, which contains at least one recessive bit.

The length of the break in the LIN frame can be set by `cr_utx_bit_cnt_b` in register `UTX_CONFIG`.

3.7.3.10.2 LIN Sync field

A synchronization byte field is sent to determine the time between two falling edges, to determine the transmission rate used by the master node. The bit pattern is 0x55 (01010101, maximum number of falling edges).

Figure 35 Sync Field of LIN

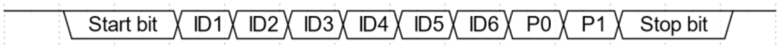


3.7.3.10.3 LIN ID field

The identifier field contains a 6-bit identifier and two parity check bits. The 6-bit identifier contains information about the sender and receiver, and the number of bytes required in the response. The parity check bit is calculated as follows:

The check bit P0 is the result of logical OR operation among ID0, ID1, ID2, and ID4. The check bit P1 is the result of inversion after logical OR operation among ID1, ID3, ID4, and ID5.

Figure 36 LIN ID Field



The slave node waits for the synchronization interval field, and then starts to synchronize between master and slave nodes through the synchronization byte field. Depending on the identifier sent by the master node, the slave node will receive, send, or do not respond. The slave node that should send data sends the number of bytes requested by the master node, and then ends the transmission with a checksum field.

UART supports the LIN transfer mode. To enable this mode, you must configure the `cr_utx_bit_cnt_b` by setting `cr_utx_lin_en` in the register `UTX_CONFIG` so that the synchronization interval field consists of at least 13-bit dominant level.

3.7.3.11 RS485 mode

UART supports the RS485 mode. After the `cr_utx_rs485_en` in the register `UTX_RS485_CFG` is set, UART can work in the RS485 mode. Then, UART can be connected to the RS485 bus through an external RS485 transceiver. In this mode, the RTS pin in the module performs the Dir function of transceiver. When UART has data to send, it will automatically control the RTS pin at a high level so that the transceiver can send data to the bus. Contrarily, when UART has no data to send, it will automatically control the RTS at a low level to keep the transceiver in the RX state.

UART supports the RS485 transfer mode. To enable this mode, you must set `cr_utx_rs485_pol` and `cr_utx_rs485_en` in the register `UTX_RS485_CFG`.

3.7.3.12 UART Interrup

UART supports the following interrupt control modes:

- TX end of transfer interrupt
- RX end interrupt
- TX FIFO request interrupt
- RX FIFO request interrupt
- RX timeout interrupt
- RX parity check error interrupt
- TX FIFO overflow interrupt
- RX FIFO overflow interrupt
- RX BCR interrupt
- LIN synchronization error interrupt

- Auto baud rate detection (universal mode) interrupt
- Auto baud rate detection (fixed characters mode) interrupt

3.7.3.12.1 TX/RX end of transfer interrupt

You can set a transfer length for TX and RX respectively by configuring the high 16-bits of the registers `UTX_CONFIG` and `URX_CONFIG`. When the number of transferred bytes reaches this value, the corresponding TX/RX end of transfer interrupt will be triggered. While this interrupt is generated, TX stops working. To continue to use TX, you must re-initialize this module. Then, RX resumes to work. If the preset transfer length of TX is less than the data volume sent by TX, the other side can only receive the data equal to the transfer length and the remaining data will be stored in TX FIFO. After this module is re-initialized, the data in TX FIFO can be sent out.

For TX, if the data is continuously filled into the TX FIFO is greater than the set transmission length value, only the data of the set length value will be transmitted on the TX pin and the excess data will be kept in the TX FIFO, and then the TX will be re-enabled. After the function, the remaining data in the TX FIFO will be sent out.

For example, set the TX transmission length value to 64, enable the TX function, first fill 63 bytes into the TX FIFO, these 63 bytes will be transmitted on the pins, but no TX transmission completion interrupt is generated, and then the TX FIFO is sent to the TX FIFO. Fill in 1 byte, at this time, the transmission length reaches the transmission length value set by TX, a transmission completion interrupt will be generated, and the TX function will stop. Continue to fill 1 byte into the TX FIFO, you will find that there is no data transmission on the pin, the byte is still retained in the TX FIFO and the TX function is turned off and re-enabled, and the byte is sent out on the pin.

For RX, if the data length sent by the other party exceeds the set transmission length, RX can continue to receive data after the RX transmission completion interrupt is generated.

For example, set the RX transmission length value to 16, the other party sends 32 bytes of data, RX will generate RX transmission completion interrupt when it receives 16 bytes of data and continue to receive the remaining 16 bytes of data, all saved in the RX FIFO.

3.7.3.12.2 TX/RX FIFO request interrupt

An TX FIFO request interrupt will be generated when `tx_fifo_cnt` in `uart_fifo_config_1` is greater than `tx_fifo_th`. When the condition is not met, the interrupt flag will be cleared automatically.

An RX FIFO request interrupt will be generated when `rx_fifo_cnt` in uart_fifo_config_1 is greater than `rx_fifo_th`. When the condition is not met, the interrupt flag will be cleared automatically.

TX/RX supports multiple rounds of transmission/receiving instead of reaching the value set by t`x_fifo_th/rx_fifo_th` at a time.

For example,

1. Set t`x_fifo_th/rx_fifo_th` in register `UART_FIFO_CONFIG_1` to 16.
2. Set `cr_utx_frm_en` in register `UTX_CONFIG` to enable free run mode.
3. Set `cr_utx_frdy_mask/cr_urx_frdy_mask` in register `UART_INT_MASK` to 0 and enable FIFO interrupt of TX/RX.
4. Set `cr_utx_en/cr_urx_en` in register `UTX_CONFIG/URX_CONFIG` to enable TX/RX.
5. TX FIFO interrupt: TX will always enter the FIFO interrupt, when the chip sends 128-bytes, set `cr_utx_frdy_mask` to 1 to shield the interrupt. If you want to enter the TX FIFO interrupt again, set `cr_utx_frdy_mask` to 0.
6. RX FIFO interrupt: The other party first sends 15-bytes, no interrupt is generated. At this time, the value of `rx_fifo_cnt` is 15 and an interrupt is generated when 1 byte is sent again to reach the value set by `rx_fifo_th`. After the transmission is interrupted, the other party sends the data again and the chip can receive the data.

3.7.3.12.3 RX timeout interrupt

The RX timeout interrupt generation condition. After receiving data last time, the receiver will start timing and the interrupt will be triggered when the timing value exceeds the timeout threshold, and the next data has not been received. The time-out threshold value is in the unit of communication bit. When the other party sends data to the chip, a timeout interrupt will be generated after the set timeout period is reached.

3.7.3.12.4 RX Parity Check Error Interrupt

The RX parity check error interrupt will be generated when a parity check error occurs. But it does not affect the RX, which still can correctly receive and analyze the data sent by the other side. When receiving data, RX takes the first 8-bits as data bits and ignores parity check bits ensuring data consistency.

For example, you can enable parity check by setting `cr_utx_prt_en/cr_urx_prt_en` in the register `UTX_CONFIG/URX_CONFIG` and select the parity check type by setting `cr_utx_prt_sel/cr_urx_prt_sel` in the register `UTX_CONFIG/URX_CONFIG`. When the other side sends data to the chip through odd/even parity check, the parity check of RX is disabled but RX can receive correct data.

3.7.3.12.5 TX/RX FIFO Overflow Interrupt

If the `TX/RX FIFO` overflows or underflows, it will trigger the corresponding overflow interrupt. When the `tx_fifo_clr` and `rx_fifo_clr` bits in the FIFO clear bit register `UART_FIFO_CONFIG_0` is set to 1, the corresponding FIFO will be cleared, and the overflow interrupt flag will be cleared automatically. You can query the interrupt status through the register `UART_INT_STS` and clear the interrupt by writing 1 to the corresponding bit in the register `UART_INT_CLR`.

3.7.3.12.6 RX BCR Interrupt

A BCR interrupt will be generated when the data received by RX reaches the value set by `cr_urx_bcr_value` in the register `URX_BCR_INT_CFG`.

The difference from RX END interrupt is that END interrupt is suitable for receiving data of known length, while BCR interrupt can be used to receive interrupts of unknown length. The trigger position of the END interrupt is controlled by `cr_urx_len` and the counter will be cleared to 0 when an interrupt is triggered. The trigger position of BCR interrupt is controlled by `cr_urx_bcr_value`. When the interrupt is triggered, the counter will accumulate instead of being cleared to 0, but it can be cleared by software (`cr_urx_bcr_clr`). When the BCR interrupt is used together with the chained DMA, check the “count” to find out how many data have been transferred by the DMA.

3.7.3.12.7 LIN Synchronization Error Interrupt

When `cr_utx_lin_en` in `UTX_CONFIG` is enabled, the LIN mode is enabled. Then, if the synchronization field of the LIN bus is not detected when data is received in this mode, the LIN synchronization error interrupt will be generated.

3.7.3.12.8 Auto Baud Rate Detection (universal/fixed characters mode) Interrupt

In the auto baud rate detection mode, when a baud rate is detected, the auto baud rate detection (universal/fixed characters mode) interrupt will be generated as configured.

3.8 GPIO Module

3.8.1 GPIO Overview

Users can connect General Purpose I/O Ports (GPIO) with external hardware devices to control these devices.

3.8.2 GPIO Features

- 20 GPIO pins
- Each I/O pin can be configured in pull-up, pull-down, or floating mode
- Each I/O pin can be configured as input, output or Hi-Z state mode
- The output mode of each I/O pin has 4 optional drive capabilities
- The input mode of each I/O pin can be set to enable/disable the Schmitt trigger

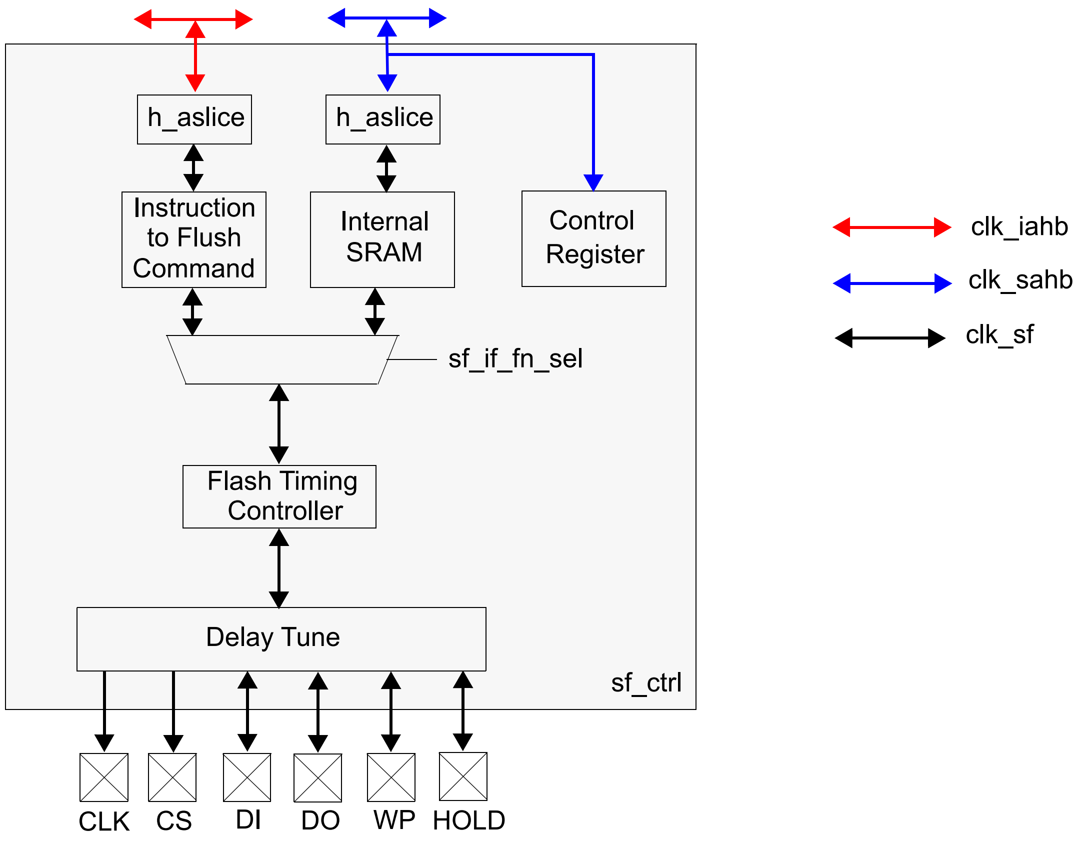
3.8.3 GPIO Pins

The IC’s GPIO pins are multiplexed with embedded peripheral functions. Please refer to section 2.2, *GPIO Pin Multiplex* and section 6.12, *GPIO Control Register*.

3.9 SF Control Module

- Execute-In-Place (XIP) for MCU application
- Configurable flash controller supports all brands of QSPI Flash
- Support Max 256MB Flash capacity
- Support Max 133MHz Flash speed (or faster according to timing constraint)
- The Flash Controller is a QSPI timing controller. It converts data to QSPI protocol in the following two modes:
  - — Indirect mode: read/erase/program flash from system bus. The Flash controller execute the sram data to/from QSPI interface.
  - — XIP mode: read flash data from instruction bus. Once the Flash controller receives the target address from instruction bus, it execute read command immediately and read back the data from QSPI flash

Figure 37 SF Control Architecture

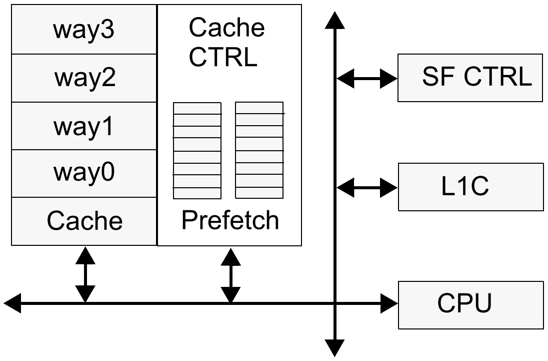


3.10 L1C Module

3.10.1 L1C Overview

L1 Cache Controller is a unit module located outside the processor, used to manage the code or data buffer on Flash and improve the speed of CPU access to Flash. The architecture is illustrated in the Figure 38. L1C is a high-speed unit integrated between the processor and Flash. Because the speed of the processor is very fast, when the processor needs to wait for a long time to access Flash, the less time wasted, the higher the efficiency. The L1C cache can be used as a lubricating role between the processor and the Flash to improve the efficiency of the processor.

Figure 38 L1C Architecture



3.10.2 Features

- 4-way Set-Associative mapping
- Support cache performance statistics

3.10.3 L1C function description

3.10.3.1 Mutual conversion between TCM and Cache RAM resources

To increase memory usage efficiency, it is supported to adjust all or part of the Cache's 16K RAM, so that users can adjust the memory usage and efficiency according to the actual situation. The maximum Cache can be set to 16K, divided into 4 ways, each way is 4K, and the unit of adjustment is 1 way, which is 4K. The default size is 16K. Set by WayDisable. The actual space size of Cache can be flexibly adjusted.

Table 12 Space size of Cache and ITCM

Way Disable	Cache	ITCM
None	16K	0K
One way	12K	4K
Two way	8K	8K
Three way	4K	12K
Four way	0K	16K

3.10.3.2 Cache

The unit of each line buffer is 32 bytes, and the 4-way associative mapping cache is used.

Each set of associative mapping caches contains two parts, the first is a tag, which contains the valid value and the address mapping relationship. The second part is data storage. When the processor accesses the cache, the cache processor compares the relationship between the address and the tag. When the address comparison is successful, the representative can directly get data from the cache. Conversely, the cache processor will capture related data through the AHB master and put the data into the cache and respond to the processor's data.

When most of the data can be successfully compared in the tag, the waiting time of the processor can be greatly reduced, and the use efficiency can be increased.

3.11 DMA Module

3.11.1 DMA Overview

DMA (Direct Memory Access) is a memory access technology that can independently read and write system memory directly without processor intervention. Under the same degree of processor load, DMA is a fast data transfer method. The DMA controller has 4 channels, which manage the data transfer between peripheral devices and memory to improve bus efficiency.

There are four main types of transfers: memory to memory, memory to peripheral, peripheral to peripheral and peripheral to memory. And support LLI link list function. Use the software to configure the transmission data size, data source address, and destination address.

3.11.2 DMA main Features

- 4 independently configurable channels (requests) on DMA
- Independent control of source and destination access width (single-byte, double-byte, four-byte)
- Each channel acts as a read-write cache independently
- Each channel can be triggered by independent peripheral hardware or software
- Support peripherals including UART, I2C and SPI
- 4 kinds of process control
  - — DMA flow control, source memory, target memory
  - — DMA flow control, source memory, target peripheral
  - — DMA flow control, source peripheral, target memory
  - — DMA flow control, source peripheral, target peripheral
- Support LLI linked list function to improve DMA efficiency

3.11.3 DMA Functional Description

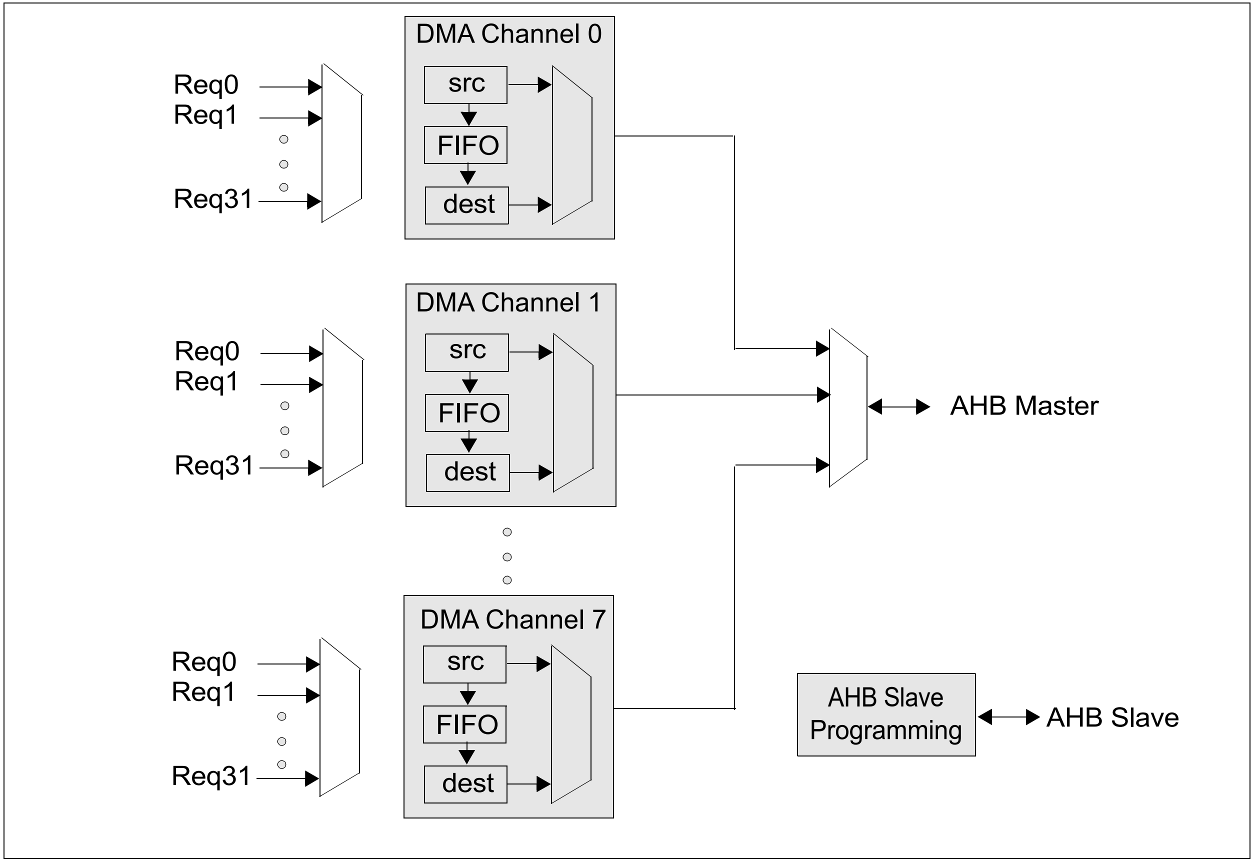
3.11.3.1 Operating principle

When a device attempts to transfer data (usually a large amount of data) directly to another device via the bus, it will first send a DMA request signal to the CPU. The peripheral device makes a bus request to the CPU to take over the bus control right through the DMA. After the CPU receives the signal, after the current bus cycle ends, it will respond to the DMA signal according to the priority of the DMA signal and the order of the DMA request.

When the CPU responds to a DMA request to a device interface, it will give up bus control.

Therefore, under the management of the DMA controller, the peripherals and the memory directly exchange data without CPU intervention. After the data transfer is complete, the device sends a DMA end signal to the CPU, returning the bus control.

Figure 39 DMA Architecture



3.11.3.2 DMA Channel Configuration

DMA supports 4 channels in total, each channel does not interfere with each other and can run at same time. The following lists the configuration process of DMA channel x.

1. Set 32-bit source address in `DMA_C[3:0]SRCADDR`.
2. Set 32-bit target address in `DMA_C[3:0]DSTADDR`.
3. Configure SI (source) and DI (destination) in `DMA_C[3:0]CONTROL` register to set whether to enable automatic address accumulation.
4. Set the transfer data width STW (source) and DTW (destination) in `DMA_C[3:0]CONTROL`. The options are single, double, and four byte.
5. Set burst type, SBS (source) and DBS (destination). The options are single, INCR4, INCR8, and INCR16.
6. A single burst cannot exceed 16-bytes.
7. Set data transmission length range: 0-4095

3.11.3.3 Peripheral Support

Peripheral ports are controlled by setting the value of register `DMA_C[3:0]CONTROL (0x2600_4[4:1]10), bit[10:6]=DSTPH` (Destination peripherals) and `bit[5:1]=SRCPH` (Source peripherals).

Table 13 Peripheral Port

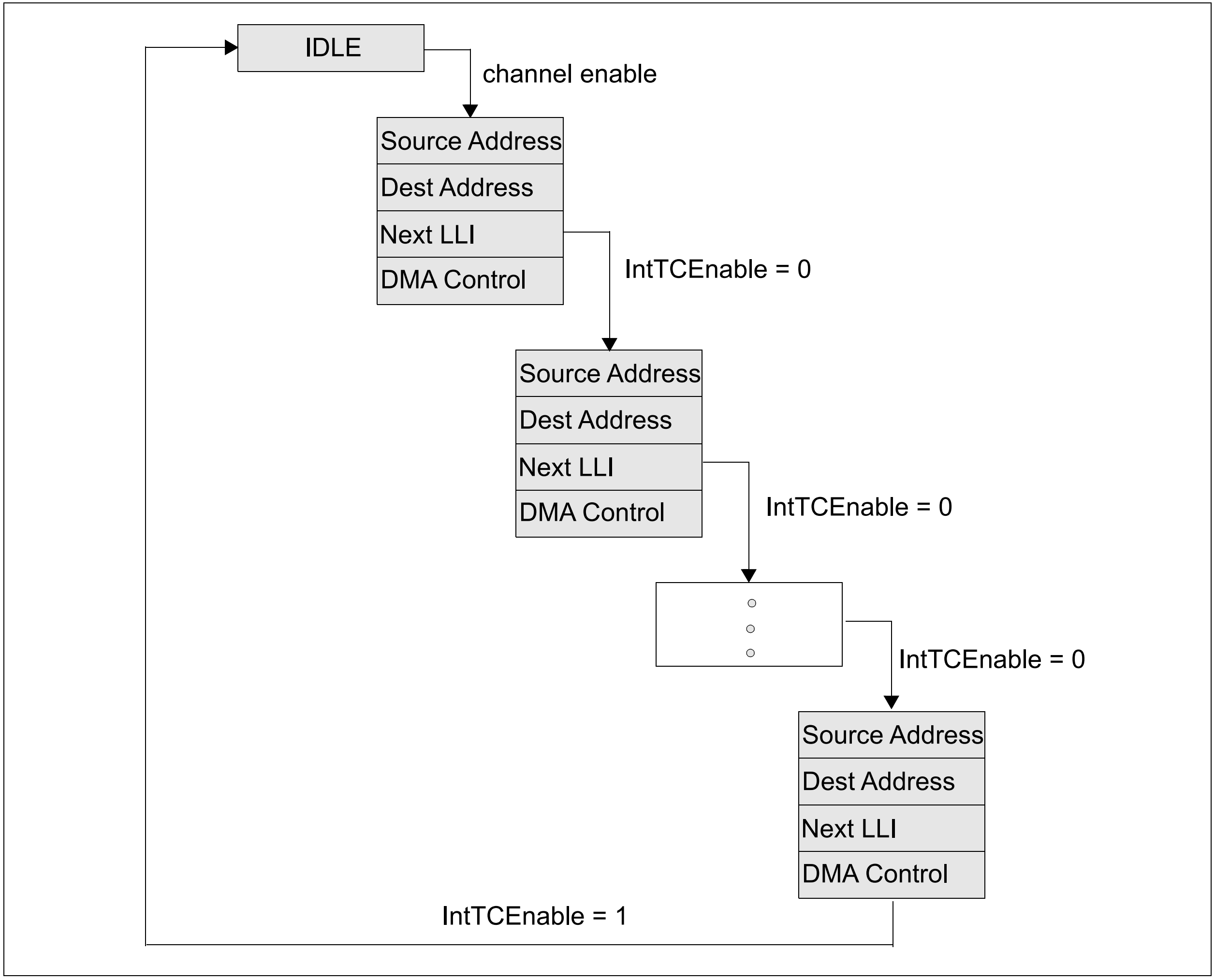
0	1	2	3	4	5	6	7
UART TX	UART RX	SPI0 TX	SPI0 RX	SPI1 TX	SPI1 RX	I2C TX	I2C RX

3.11.3.4 Linked List Mode

DMA supports linked list operation mode: When performing DMA read or write operation, you can fill data in next linked list. After completing the data transfer of current list, read the `DMA_C[3:0]LLI` register to obtain the start address of the next linked list, and directly transfer the data in the next linked list.

Ensure continuous and uninterrupted work during DMA transfer and improve the efficiency of CPU and DMA.

Figure 40 LLI Architecture



3.11.3.5 DMA interrupt

- DMA_INT_TCOMPLETED
  - — Data transfer complete interrupt: This interrupt will be generated when a data transfer is completed
- DMA_INT_ERR
  - — Data transfer error interrupt: This interrupt will be generated when an error occurs during data transfer

3.11.4 DMA Transmission Mode

3.11.4.1 Memory to memory

After this mode is started, the DMA will move the data from the source address to the destination address according to the set transfer size. After the transfer, the DMA controller will automatically return to the idle state and wait for the next transfer.

The specific configuration process is as follows:

1. Set the value of the register `DMA_C[3:0]SRCADDR` to the memory address of the source
2. Set the value of the register `DMA_C[3:0]DSTADDR` to the target memory address
3. Select the transmission mode and set the value of the `FLOWCTRL` bit in register `DMA_C[3:0]CONFIG` to 0, that is, select the memory-to-memory mode
4. Set the value of the corresponding bit in the `DMA_C[3:0]CONTROL` register: set the DI and SI bits to 1 to enable the automatic address accumulation mode, the `DTW` and `STW` bits set the transmission width of the source and destination, and the `DBS` and `SBS` bits set the burst type of the source and destination

3.11.4.2 Memory to Peripheral

In this working mode, the DMA will move data from the source to the internal cache according to the set transfer size (`TransferSize`). When the cache space is insufficient, the DMA will automatically suspend it. When there is sufficient cache space, continue to transfer until it reaches Set the moving quantity.

On the other hand, when the target peripheral request triggers, it will burst the target configuration to the target address until it reaches the set number of moves and automatically returns to the idle state, waiting for the next startup.

The specific configuration process is as follows:

1. Set the value of the register `DMA_C[3:0]SRCADDR` to the memory address of the source
2. Set the value of the register `DMA_C[3:0]DSTADDR` to the target peripheral address
3. Select the transmission mode and set the value of the FLOWCTRL bit in register `DMA_C[3:0]CONFIG` to 1, that is, select the memory-to-peripheral mode
4. Set the value of the corresponding bit in the `DMA_C[3:0]CONTROL` register: the SI bit is set to 1 to enable the address auto-accumulation mode, and the DI bit is set to 0 to disable the address auto-accumulation mode, the DTW and STW bits set the transmission width of the source and destination, and the DBS and SBS bits set the burst type of the source and destination
5. Select the appropriate channel, enable DMA, and complete the data transfer

3.11.4.3 Peripheral to Memory

In this working mode, when the source peripheral request is triggered, the source configuration is burst to the buffer until the set number of moves reaches the stop. On the other hand, when the internal cache is enough for the target burst number once, the DMA will automatically move the cached content to the target address until it reaches the set number of moves and automatically returns to the idle state, waiting for the next startup

The specific configuration process is as follows:

1. Set the value of the register `DMA_C[3:0]SRCADDR` to the source peripheral address
2. Set the value of the register `DMA_C[3:0]DSTADDR` to the target memory address
3. Select the transmission mode and set the value of the `FLOWCTRL` bit in register `DMA_C[3:0]CONFIG` to 2, that is, select the peripheral-to-memory mode
4. Set the value of the corresponding bit in the `DMA_C[3:0]CONTROL r`egister: the DI bit is set to 1 to enable the address auto-accumulation mode, and the SI bit is set to 0 to disable the address auto-accumulation mode , the DTW and STW bits set the transmission width of the source and destination respectively, and the DBS and SBS bits set the burst type of the source and destination respectively
5. Select the appropriate channel, enable DMA, and complete the data transfer

3.11.4.4 Peripheral to Peripheral

In this working mode, when the source peripheral requests a trigger, the source configuration burst will be stored in the buffer, and it will stop until the set number of moves is reached. On the other hand, when the internal cache is enough for the target burst number, DMA will automatically move the cached content to the target address until the set number of transfers is reached and automatically return to the idle state, waiting for the next start

The specific configuration process is as follows:

1. Set the value of the register `DMA_C[3:0]SRCADDR` to the peripheral address of the source
2. Set the value of the register `DMA_C[3:0]DSTADDR` to the target peripheral address
3. Select the transmission mode and set the value of the FLOWCTRL bit in register `DMA_C[3:0]CONFIG` to 3, that is, select the peripheral-to-peripheral mode
4. Set the value of the corresponding bit in the `DMA_C[3:0]CONTROL` register: DI and SI bits are set to 0, the address automatic accumulation mode is disabled, the STW and DTW bits respectively set the source and target transfer widths, and the SBS and DBS bits respectively set the source and target bursts type.
5. Select the appropriate channel, enable DMA, and complete the data transfer

3.12 SEC ENGINE Module

3.12.1 SEC Engine Overview

SEC ENG has a variety of built-in computing modules, including AES, SHA.

3.12.2 SEC Engine Features

- AES
  - — Supports 128-bit, 192-bit, and 256-bit key length
  - — Supports encryption and decryption of multiple link modes (ECB/CBC/CTR/XTS)
  - — Exclusive AES LINK function
- SHA
  - — Supports SHA1/SHA224/SHA256/SHA384/SHA512
  - — Exclusive SHA LINK function

3.12.3 SEC Engine Functional Description

3.12.3.1 AES Accelerator

3.12.3.1.1 Key

The key length required for encryption mode and decryption mode can be selected by configuring `se_aes_0_mode` and `se_aes_0_dec_en` in the register `SE_AES_0_CTRL`.

Table 14 AES operation mode diagram

amode	adec en	Operation mode
0	0	AES-128 encryption
1	0	AES-128 encryption
2	0	AES-128 encryption
0	1	AES-128 encryption
1	1	AES-128 encryption
2	1	AES-128 encryption

Select whether to enable the hardware key through `aes_0_hw_key_en` in the register `SE_AES_0_CTRL`. If you use a software key, you also need to configure the registers `SE_AES_0_KEY_0~SE_AES_0_KEY_7` to store the key and each register stores a 4-byte key.

3.12.3.1.2 Link mode

Different link modes can be selected through `se_aes_0_block_mode` in the register `SE_AES_0_CTRL`. Currently, ECB, CBC, CTR, and XTS modes are supported

3.12.3.1.3 Plaintext or Ciphertext

- Plaintext or ciphertext must be a multiple of 16
- The register `SE_AES_0_MSA` stores the plaintext address entered during encryption or the ciphertext address entered during decryption
- The register `SE_AES_0_MDA` stores the ciphertext address output during encryption or the plaintext address output during decryption

- se_aes_0_msg_len in the register SE_AES_0_CTRL is used to set the length of ciphertext or plaintext (in units of 16-bytes)

3.12.3.1.4 Initialization vector

The registers SE_AES_0_IV_0~SE_AES_0_IV_3 store the initialization vector (IV). You can choose whether to use a new IV by configuring aes_0_iv_sel in register SE_AES_0_CTRL. You must clear 0 when configuring iv for the first time and must set 1 if you continue to use this iv or automatically update iv.

3.12.3.1.5 Encryption and decryption configuration process

- Enable AES with se_aes_0_en in the configuration register SE_AES_0_CTRL.
- Configuration register SE_AES_0_ENDIAN, including se_aes_0_dout_endian, se_aes_0_din_endian, se_aes_0_key_endian, se_aes_0_iv_endian, and se_aes_0_twk_endian. If the value is 0, it means little-endian and if the value is 1, it means big-endian.
- The se_aes_0_block_mode in the configuration register SE_AES_0_CTRL selects the link mode.
- The se_aes_0_mode in the configuration register SE_AES_0_CTRL selects the key length
- To use software key, configure registers SE_AES_0_KEY_0~SE_AES_0_KEY_7 to store the key. To use a hardware key, set aes_0_hw_key_en in register SE_AES_0_CTRL.
- Configure registers SE_AES_0_IV_0~SE_AES_0_IV_3 to set IV, the filling order for MSB is se_aes_0_iv_0~se_aes_0_iv_3 and the filling order for LSB is se_aes_0_iv_3~se_aes_0_iv_0.
- The se_aes_0_dec_en in the configuration register SE_AES_0_CTRL selects the encryption or decryption mode.
- The configuration register SE_AES_0_MSA sets the source address of the data to be processed.
- The configuration register SE_AES_0_MDA sets the destination address where the processing result is stored.
- The se_aes_0_msg_len in the configuration register SE_AES_0_CTRL sets the length of the data to be processed in units of 16-bytes
- The se_aes_0_trig_1t in the configuration register SE_AES_0_CTRL triggers AES to run.
- The result is output to the destination address specified by register SE_AES_0_MDA.

3.12.3.2 SHA Accelerator

3.12.3.2.1 SHA mode

The se_sha_0_mode in the register SE_SHA_0_CTRL:

- 0: SHA-256;
- 1: SHA-224;
- 2: SHA-1;
- 3: SHA-1;
- 4: SHA-512;
- 5: SHA-384;
- 6: SHA-512/224;
- 7: SHA- 512/256;

3.12.3.2.2 Plaintext and Ciphertext

- The register SE_SHA_0_MSA stores the plain text address.
- The registers SE_SHA_0_HASH_L_0~SE_SHA_0_HASH_L_7 stores the cipher text.

3.12.3.2.3 Operation Flow

- Configure se_sha_0_mode in the register SE_SHA_0_CTRL to set the specific mode of SHA.
- Enable SHA by configuring se_sha_0_en in the register SE_SHA_0_CTRL.
- Configure se_sha_0_hash_sel in the register SE_SHA_0_CTRL. 0 means starting a new HASH calculation, and 1 means using the last result for HASH calculation.
- Configure the register SE_SHA_0_MSA to set the source address of the data to be processed.
- Configure se_sha_0_msg_len in the register SE_SHA_0_CTRL to set the length of the data to be processed (512-bits for SHA1, SHA224, and SHA256, while 1024-bits for SHA512, SHA384, SHA512/224, and SHA512/256)
- Configure se_sha_0_trig_1t in the register SE_SHA_0_CTRL to trigger SHA.
- The output result is stored in registers SE_SHA_0_HASH_L_0~SE_SHA_0_HASH_L_7,
  - — MSB: se_sha_0_hash_l_0~se_sha_0_hash_l_7
  - — LSB: se_sha_0_hash_l_7~se_sha_0_hash_l_0

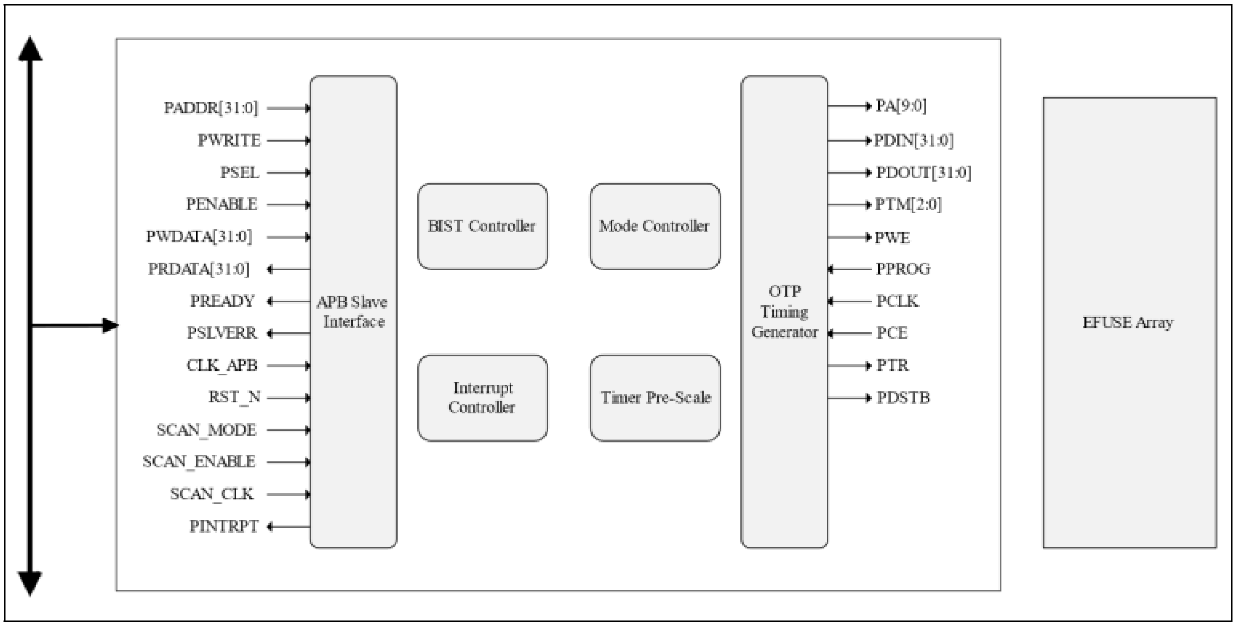
3.13 OTP Module

OTP is integrated into Mockingbird’s µC subsystem (base address: 0x2A08_0000, IRQ21).

- System bus APB is connected to µC
- “PINTRPT” is connected to E21 IRQ21
- EFUSE = 4KB (1Kx32)
- System APB bus runs at µC APB speed: 200MHz

The APB system bus module is described in Figure 41.

Figure 41 System APB Bus



The OTP controller addresses are listed in Table 15.

Table 15 OTP controller address

Offset	Size	Usage	Comment
0x8000 ~ 0x8ffc	32K	OTP	eFuse
0x0800 ~ 0x081c		PTR	Test row address pa[0]~pa[7]
0x3400 ~ 0x3508		CFG	CSR in register section

The OTP array assignment is listed in Table 16.

Table 16 OTP Array Assignment

Offset	Size	Usage
0~0x3f	512b	AES key
0x40~0x7f	512b	SHA key
0x80~0x8f	128b	BIOS Straps
0x90~0x3ff		Process data and misc use

3.14 I3C Module

3.14.1 I3C Overview

An I3C controller is implemented in each chiplet for inter-chiplet synchronization and message passing. µC in each chiplet access register bus of application layer to transmit and receive packets between chiplets.

The controller in I3C system supports dynamic address assignment (DAA). The controller is responsible for generating the clock, issuing commands, and controlling the data transfer. Each I3C device has a unique address assigned by DAA during power up or hot join.

The APB interface is connected to µC's (E21) bus matrix (BMX). I3C transferred through Command/Tx/Rx FIFO. Interrupt is serviced by µC interrupt controller. DMA interface is disabled.

- Application clock (pclk) = µC bus clock (200 MHz)
- Core clock (core_clk) = free run µC bus clock (200 MHz)

Table 17 shows an I3C example.

Table 17 I3C Example

Clock Domain	Min Freq	Typ Freq	Max Freq
core_clk (core clock)	125 MHz	125 MHz	700 MHz
pclk (application clock)	30 MHz	50 MHz	700 MHz
dma_clk (DMA clock)	30 MHz	50 MHz	700 MHz

I3C controller supports only subset of I2C features. Following are the limitations.

- 10-bit addressing is not supported
- High speed mode is not supported
- SCL is always driven in push-pull
- I2C controller in system is not supported
- Bus clear feature is not supported

3.14.2 I3C slave provisional ID register

These registers are used in slave mode of operation.

- Register: `SLV_MIPI_ID_VALUE` Address: `0x2A010070`

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	NA	NA	Reserved
15:1	SLV_MIPI_MFG_ID	15'h0	R/W	Specifies the MIPI manufacture ID. PID [47:33]. Reset value of this register field is considered from input port signal slv_pid[47:33]. Note: FLC's MIPI Manufacture ID = 15'h0528
0	SLV_PROV_ID_SEL	1'h0	R/W	Specifies the Provisional ID type selector PID[32]. Reset value of this register field is considered from input port signal slv_pid[32] 1'b1: Random Value 1'b0: Vendor Fixed Value

- Register: `SLV_PID_VALUE` Address: `0x2A010074`

Bits	Bit Name	Default	Type	Comment
31:16	SLV_PART_ID	16'h0	R/W	Specifies the part ID of DWC_mipi_i3c device PID [31:16]. Reset value of this register field is considered from input port signal slv_pid [31:12]. **Note:** FLC's part ID = 16'h0001
15:12	SLV_INST_ID	4'h0	R/W	This field is used to program the instance ID of the slave. Reset value of this register field is considered from input port signal slv_pid[15:12]. **Note:** FLC's instance ID = 4'h0
11:0	SLV_PID_DCR	12'h0	R/W	Specifies the additional 12-bit ID of DWC_mipi_i3c device PID [11:0]. Reset value of this register field is considered from input port signal slv_pid [11:0]. **Note:** FLC's additional ID = 12'h0C6

3.14.3 I3C Private read/write opcodes

Below is the format that has been followed for Goldfinch/Mockingbird to define the I3C private read/write commands.

Table 18 I3C Private read/write Commands

MSB	Octet 1	Octet 2	Octet 3	Octet 4
Word 1	Error Code	Data Length		Opcode
Word 2	Address			
Word 3	Data 1			
Word 4	Data 2			

Based on the Data size, the word size can be 4 or 64 words.

The below OPCODES are predefined by FLC and are used as part of the Firmware to define the I3C communication between MKB and GFH.

```
0x01 READ_REGISTER_OPCODE
0x02 WRITE_REGISTER_OPCODE
0x03 READ_REGISTER_RESPONSE_OPCODE
0x04 WRITE_REGISTER_RESPONSE_OPCODE
0x05 D2D_OPCODE
0x06 D2D_RESP_OPCODE
0x80 FW_DOWNLOAD_COMPLETE_OPCODE
0x81 FW_DOWNLOAD_COMPLETE_RESP_OPCODE
```

Also note that, based on opcode, data format might change. Below are the examples for the `READ_REGISTER_OPCODE` and `D2D_OPCODE`.

Table 19 I3C Read Register OPCODE

MSB	Octet 1	Octet 2	Octet 3	Octet 4
Word 1	Error Code	Data Length		READ_REGISTER_OPCODE
Word 2	Address			
Word 3	Data 1			
Word 4	Data 2			

Table 20 I3C D2D OPCODE

MSB	Octet 1	Octet 2	Octet 3	Octet 4
Word 1	Error Code	Data Length		D2D_OPCODE
Word 2	D2D SYNC STATUS			
Word 3	Data 1			
Word 4	Data 2			

3.15 AXI4 Traffic Generator Module

3.15.1 AXI4 Traffic Generator Overview

The AXI4 traffic generator is a fully synthesizable AXI4-compliant core with the following features.

- Configurable option to generate and accept data according to different traffic profiles.
- Supports dependent/independent transaction between read/write masgter port with configurable delays.
- Programmable repeat count for each transaction with constant/increment/random address.
- External start/stop to generate traffic without processor intervention.
- Generates IP specific traffic on AXI interface for pre-defined protocols.

3.15.2 AXI4 Traffic Generator Sequence

The AXI4 traffic generator sequence:

1. Set `axi4tg_resetn (0x2600_0418, axi4tg_ctl[0]) = 1'b1`.
2. Load `CMDRAM (R/W)`:

   `CMDRAM (RD) 0x2a028_000~0x2a028_fff` (256 commands 128-bit wide),

   `CMDRAM (WR) 0x2a029_000~0x2a029_fff` (256 commands 128-bit wide).

   Note: There should be at least one command with `valid_cmd` bit set to zero (i.e., one invalid command) for both reads and write.
3. Load `PARAM` (Write only):

   `PARAM (RD) 0x2a021_000~0x2a021_3ff` (256 commands 32-bit wide),

   `PARAM (WR) 0x2a021_400~0x2a021_7ff` (256 commands 32-bit wide).
4. Load `MSTRAM (R/W)`:

   `0x2a02c_000~0x2a02c_fff`. (128 data 256-bit wide),

   AXI data width = 256-bit (`mstram_index` valid values = 0x0, 0x20, 0x40,…,fe0).
5. Enable the desired interrupt/status bits:

   Set `MSTIRQEN (0x2a02_000c[31])` and other error enable bits,

   Set `MINTREN (0x2a02_0010[15])`.
6. Start AXI4TG:

   Set `MSTEN (0x2a02_0000[20])` or

   Set `axi4tg_start (0x2600_0418[1])`.
7. Check completion:

   Wait for interrupt `axi4tg_irq (0x2600_0418[16])` or

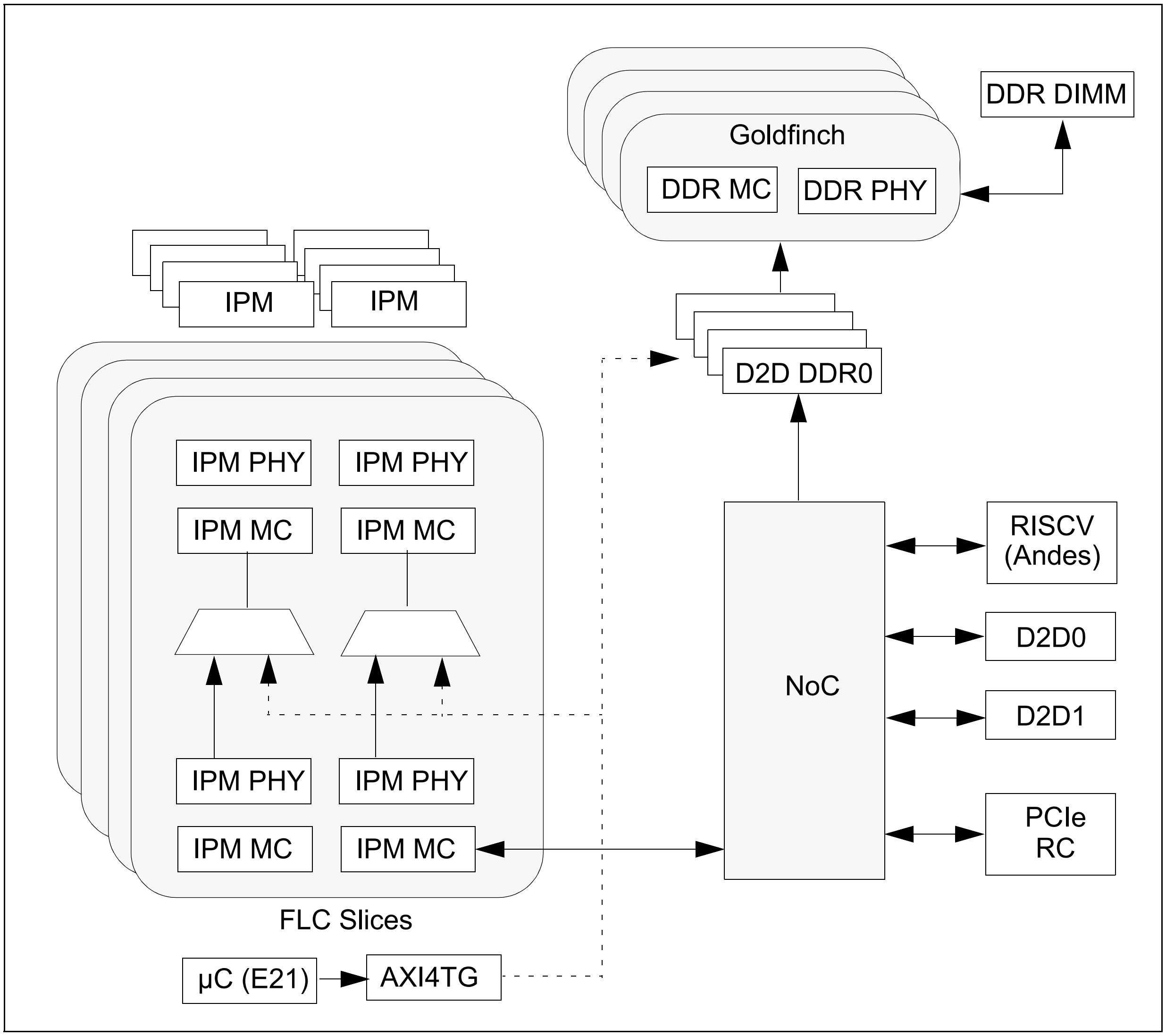
   Poll for `MSTDONE (0x2a02_0008[31]`, write 1'b1 to clear) and check other error status bits or

   Check `MSTEN (0x2a02_0000[20])`, this bit is automatically cleared to indicate to SW that the AXI4TG is done.
8. Compare R/W data in `MSTRAM`.

3.15.3 IPM AXI4TG

Figure 42 illustrates IPM slice.

Figure 42 IPM AXI4TG



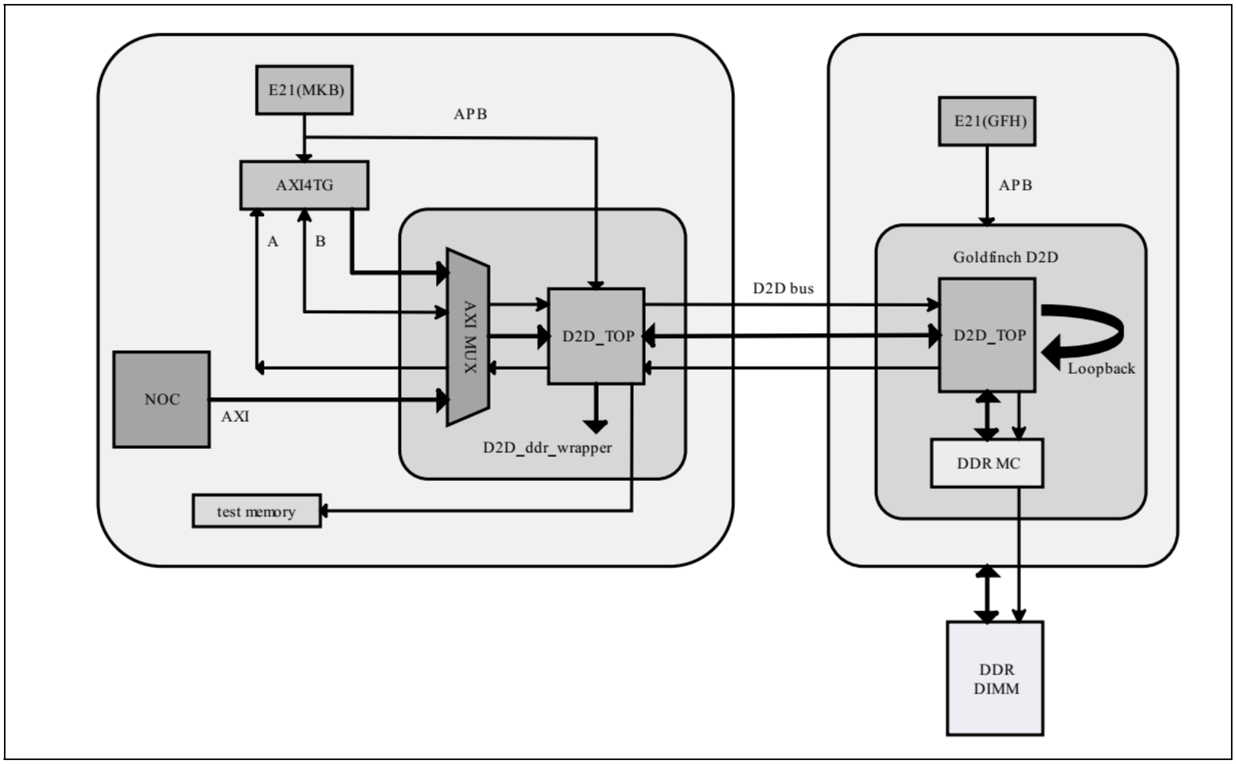
Build in AXI4 traffic generator to test IPM, bypass FLC1.

1. E21 init IPM PHY, MC.
2. E21 Programs AXI4 TG module.
3. E21 selects one of the IPM to test via CSR. There are total of 9 IPMs in Mockingbird.
4. E21 sets FLC1_MC_TOP AXI MUX to select path from AXI4 TG.
5. E21 set CSR bit to start generating traffic at IPM MC’s AXI interface. 6. After test is done, E21 access AXI4 TG to check result.

3.15.4 D2D AXI4TG

The Figure 43 illustrate the chiplet test mode and the AXI traffic generator modules.

Figure 43 D2D AXI4TG



Build in AXI4 traffic generator to test D2D, bypass NoC.

Test flow A with D2D loopback

1. E21 configures D2D.
2. E21 programs pattern in AXI4 TG.
3. AXI4 TG generates AXI4 write traffic through d2d_ddr_wrapper to GFH d2d (loopback), write to test memory.
4. AXI4 TG generates AXI4 read traffic through d2d_ddr_wrapper to GFH d2d (loopback), read from test memory.
5. E21 checks AXI4 TG read buffer.
6. Test Memory: AHB memory at E21 address space:0x40xx_xxxx.

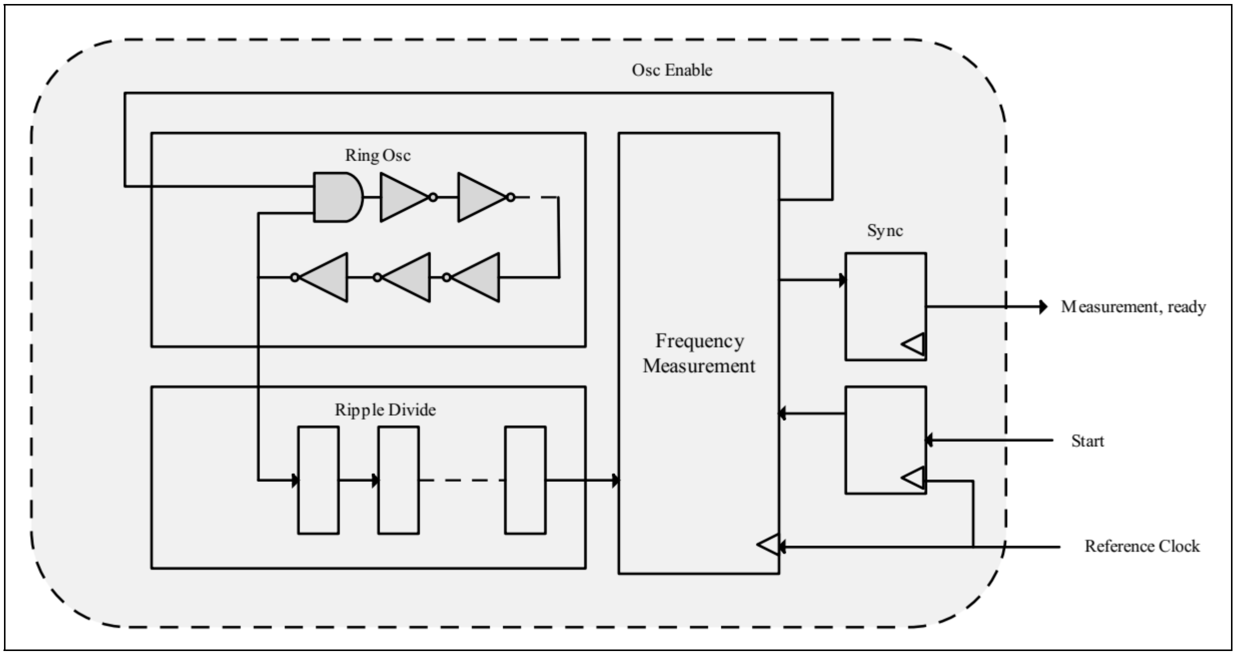
Test flow B without D2D loopback

1. E21 configures D2D and DDR MC.
2. E21 programs pattern in AXI4 TG.
3. AXI4 TG generates AXI4 write traffic through d2d_ddr_wrapper to GFH d2d, write to DDR.
4. AXI4 TG generates AXI4 write traffic through d2d_ddr_wrapper to GFH d2d, read from DDR.
5. E21 checks AXI4 TG read buffer.

3.16 DRO Module

Six DRO modules are included for process monitoring.

Figure 44 DRO for Process Monitor



Reference clock: 200MHz (μc_clk)

rosc chain: 83 stages, typical case, period= ~938ps

- Multiple Vt and/or combined cells. Harden at PAR stage.
- Ring stopped automatically after measurement complete (ready=1).

Multiple instances at different locations. Ripple: 8 stage, period = 938ps*256 = 240128ps

Period count by 200Mhz clock: measurement = 240128/5000 =~ 48

CSR bit

- Start (R/W):

  0x2600_0800[0] = dro0_start; 0x2600_0800[16] = dro1_start;

  0x2600_0804[0] = dro2_start; 0x2600_0804[16] = dro3_start;

  0x2600_0808[0] = dro4_start; 0x2600_0808[16] = dro5_start;

- Measurement Ready (RO):

  0x2600_0800[1] = dro0_ready; 0x2600_0800[17] = dro1_ready;

  0x2600_0804[1] = dro2_ready; 0x2600_0804[17] = dro3_ready;2024 FLC Technology Group Inc. Confidential 94

  0x2600_0808[1] = dro4_ready; 0x2600_0808[17] = dro5_ready;

- Measurement result (RO):

  0x2600_0800[15:8] = dro0_mea; 0x2600_0800[31:24] = dro1_mea;

  0x2600_0804[15:8] = dro2_mea; 0x2600_0804[31:24] = dro3_mea;

  0x2600_0808[15:8] = dro4_mea; 0x2600_0808[31:24] = dro5_mea;

3.17 EMAC Module

3.17.1 Introduction

The EMAC module is a 10/100Mbps Ethernet MAC (Ethernet Media Access Controller) compatible with IEEE 802.3. It includes status and control register group, transceiver module, transceiver buffer descriptor group, host interface, MDIO, physical layer chip (PHY) interface.

The status and control register group contains the status bits and control bits of the EMAC, which is the interface with the user program, and is responsible for controlling data receiving and sending and reporting the status.

The transceiver module is responsible for obtaining the data frame from the designated memory location according to the control word in the transceiver descriptor, adding the preamble, CRC, and expanding the short frame before sending it out through the PHY; Or receive data from the PHY, and put the data into the designated memory according to the transmit and receive buffer descriptor. Configure related event flags after sending and receiving. If the event interrupt is enabled, the interrupt request will be sent to the host for processing.

The MDIO and MII/RMII interfaces are responsible for communicating with the PHY, including reading and writing PHY registers and sending and receiving data packets.

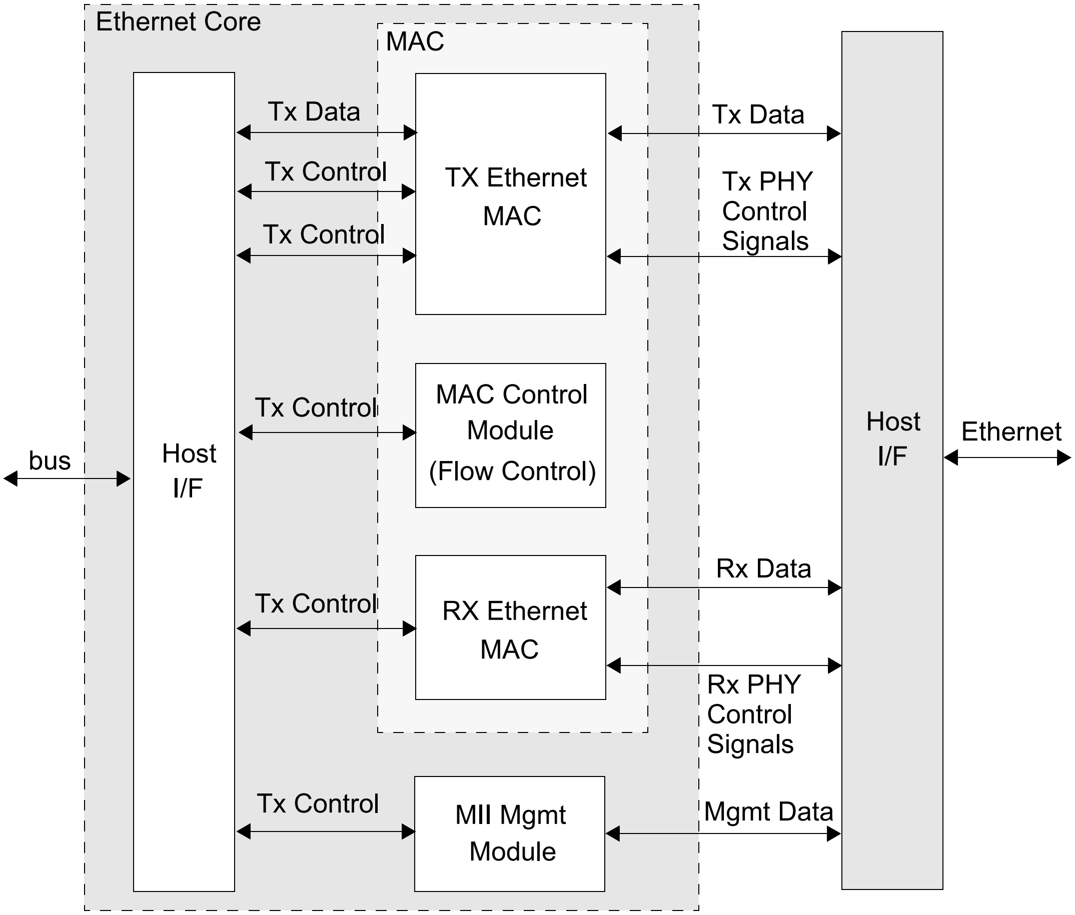
3.17.2 Main Features

- Compatible with the MAC layer functions defined by IEEE 802.3
- Support MII and RMII interface PHY defined by IEEE 802.3
- Interaction with PHY through MDIO
- Support 10Mbps and 100Mbps Ethernet
- Support half duplex and full duplex
- In full-duplex mode, support automatic flow control and control frame generation
- Support collision detection and retransmission in half-duplex mode
- Support CRC generation and verification
- Data frame preamble generation and removal
- When sending, automatically expand short data frames
- Detect too long or too short data frame (length limit)
- Can transmit long data frames (> standard Ethernet frame length)
- Automatically discard packets that exceed the number of retransmissions or the frame gap is too small
- Broadcast packet filtering
- Internal RAM for storing up to 128 BD (Buffer Descriptor)
- Various event flags sent/received
- Generate a corresponding interrupt when an event occurs

3.17.3 Function Description

The composition of the EMAC module is shown in the Figure 45.

Figure 45 EMAC Module



3.17.4 Architecture

The module's control register can read and write the PHY register through MDIO to realize configuration, select mode (half/full duplex), initiate negotiation and other operations.

The receiving module filters and checks the received data frame: whether there is a legal preamble, FCS, length, etc. And according to the descriptor, the data frame is stored in the designated buffer address.

The sending module obtains data from the memory according to the data buffer descriptor, adds preamble, FCS, pad, etc., and then sends the data according to the CSMA/CD protocol.

If CRS is detected, the retry will be delayed.

The send and receive buffer descriptor group is connected to the external RAM, which is used to store the Ethernet data frames sent and received. Each descriptor contains the corresponding control status word and the corresponding buffer memory address. There are 128 groups of descriptors, which can be flexibly allocated for sending or receiving.

3.17.4.1 EMAC Clock

The EMAC module needs a clock for synchronous transmission and reception (at 100Mbps, 25MHz (MII) or 50MHz (RMII); at 10Mbps, 2.5MHz). This clock must be synchronized between EMAC and PHY.

3.17.4.2 Send and Receive Buffer Descriptor (BD, Buffer Descriptor)

The transceiver buffer descriptor is used to provide the association between the EAMC and the data frame buffer address information, to control the transceiver data frame, and to provide the transceiver status prompt.

Each descriptor is composed of two consecutive words (32 bits). The low address word provides the length, control and status bits of the data frame contained in the buffer; the high address word is a memory pointer.

Specific BD description can refer to the register description chapter.

Note: For BD, you need to write in word.

The EMAC module supports 128 BDs, which are shared by the sending/receiving logic and can be freely combined. But the sending BD always occupies the previous continuous area (the number is specified by the `TXBDNUM` field in the `MAC_TX_BD_NUM` register).

EMAC processes the sending/receiving BD in the order of BD, until it encounters the BD marked as WR, it wraps around to send/receive the respective first BD.

3.17.4.3 PHY Interaction

The PHY interaction register group provides the commands and data communication methods needed for PHY interaction. EMAC controls the working mode of PHY through MDIO and ensures that the two match (rate, full/half duplex).

The data packet interacts between EMAC and PHY through the MII/RMII interface, and can be selected by `RMII_EN` in the EMAC mode register (`EMAC_MODE`). When this bit is 1, select RMII mode, otherwise it is MII mode.

Both MII and RMII modes support the 10Mbps and 100Mbps transmission rates specified in the IEEE 802.3u standard.. The transmission signal description of MII and RMII is shown in the Table 21.

Table 21 Transmission signal

Name	MII	RMII
EXTCK_EREFCK	ETXCK: send clock signal	EREFCK: reference clock
ECRS	ECRS: carrier detection	-
ECOL	ECOL: collision detection	-
ERXDV	ERXDV: data valid	ECRSDV: Carrier detect/data valid
ERX0-ERX3	ERX0-ERX3: 4-bit receive data	ERX0-ERX1: 2-bit receive data
ERXER	ERXER: Receive error indication	ERXER: Receive error indication
ERXCK	ERXCK: Receive clock signal	-
ETXEN	ETXEN: transmit enable	ETXEN: transmit enable
ETX0-ETX3	ETX0-ETX3: 4-bit transmit data	ETX0-ETX1: 2-bit transmit data
ETXER	ETXER: Send error indication	-
EMDC	MDIO Clock	MDIO Clock
EMDIO	MDIO Data Input Output	MDIO Data Input Output

The RMII interface has fewer pins, and a 2-bit data line is used for receiving and sending. At a rate of 100Mbps, a 50MHz reference clock is required.

3.17.4.4 Programming Process

The programming process is explained in the following sections.

3.17.4.4.1 PHY Initialization

- According to the PHY type, set the RMII_EN bit in the EMAC_MODE register to select the appropriate connection method
- Set the MAC address of EMAC to `EMAC_MAC_ADDR0` and `EMAC_MAC_ADDR1`
- Set the appropriate clock for the MDIO part by programming the field `CLKDIV` in the `EMAC_MIIMODE` register
- Set the corresponding PHY address to the FIAD field of the register `EMAC_MIIADDRESS`
- According to the PHY manual, send commands through the `EMAC_MIICOMMAND` and `EMAC_MIITX_DATA` registers
- The data read from the PHY will be stored in the `EMAC_MIIRX_DATA` register
- The status of interaction with PHY commands can be queried through the `EMAC_MIISTATUS` register

After the basic interaction is completed, the PHY should enter the auto-negotiation state. After the negotiation is completed, program the mode to the `FULLD` bit in the `EMAC_MODE` register according to the negotiation result.

3.17.4.4.2 Send Data Frame

- Configure bit fields such as data frame format and interval in the `EMAC_MODE` register
- By configuring the `TXBDNUM` field in the `EMAC_TX_BD_NUM` register to specify the number of BDs used for transmission, the remaining BDs are RX BDs
- Prepare the data frame to be sent in the memory
- Fill in the address of the data frame into the data pointer field (word 1) corresponding to the sent BD
- Clear the status flag in the control and status field (word 0) corresponding to the sent BD, and set the control field (CRC enable, PAD enable, interrupt enable, etc.)
- Write the length of the data frame and set the RD field to inform EMAC that this BD data needs to be sent; if necessary, set the IRQ bit to enable interrupts
- In particular, if it is the last BD sent, the WR bit needs to be set, and EMAC will “wrap around” to the first sent BD for processing after processing this BD
- If there are multiple BDs to be sent, repeat the steps of setting BD to fill all sending BDs
- If you need to enable the transmit interrupt, you also need to configure the TX related bits in the `EMAC_INT_MASK` register
- Configure the `TXEN` bit in the `EMAC_MODE` register to enable transmission
- If the interrupt is enabled, in the interrupt sent, the current BD can be obtained through the `TXBDNUM` field in the `EMAC_TX_BD_NUM` register
- Perform corresponding processing according to the current BD status word
- For the BD whose data has been sent, the RD bit in the control field will be cleared by hardware and will not be sent again; after filling in new data, set RD, and this BD can be used for sending again

3.17.4.4.3 Receive Data Frame

- Configure bit fields such as data frame format and interval in the `EMAC_MODE` register
- By configuring the `TXBDNUM` field in the `EMAC_TX_BD_NUM` register to specify the number of BDs used for transmission, the remaining BDs are RX BDs
- The area in the memory that is ready to receive data

- Fill in the address of the data frame into the data pointer field (word 1) corresponding to the received BD
- Clear the status flag in the control and status field (word 0) corresponding to the sending BD, and set the control field (interrupt enable, etc.)
- Write the receivable data frame length and set the E bit field to inform EMAC that the BD is idle and can be used for data reception; if necessary, set the IRQ bit to enable interrupts
- In particular, if it is the last valid receiving BD, the WR bit needs to be set, and EMAC will “wrap around” to the first receiving BD for processing after processing this BD
- If there are multiple BDs available to receive data, repeat the steps of setting BD to fill all BDs
- If you need to enable the receive interrupt, you also need to configure the RX related bits in the `EMAC_INT_MASK` register
- Configure the `RXEN` bit in the `EMAC_MODE` register to enable reception
- If the interrupt is enabled, in the received interrupt, the current BD can be obtained through the `RXBDNUM` field in the `EMAC_TX_BD_NUM` register
- Perform corresponding processing according to the current BD status word
- For the received BD, the E bit in the control field will be cleared by hardware and will not be used for receiving again; the data needs to be taken away, and E is set, this BD can be used for receiving again

3.18 CXL PCIe Module

3.18.1 PCIe PHY

Alphawave PCIe PHY CSR access: PHY selected by `pcie_phy_sel (0x2600_0038[3:0])`,

- Register: `INTERFACE_SEL`, Address: `0x26000038`

Bits	Bit Name	Default	Type	Comment
3:0	pcie_phy_sel	4’h0	R/W	Select CXL/PCIe interface for configuration. 0: RCx8[0] MAC [NVMe host/CXL RC] 1~4: host side slice pcie_ep[0:3] 5: RCx4[0] MAC 6: RC1_x8[1] MAC 7: RC1_x4[1] MAC 8~13: Reserved 14: Broadcast (RCx8[0:1]) 15: Broadcast all

RC interface is 4/8 lane, CXL/PCIe 5 root complex, connects to external switch or NVMe SSD or another DDR based CXL memory expander card. Host interface is 8 lane, CXL endpoint, connects to hosts

- PMA CSR: `0x2800_0000~0x283f_ffff.` Table 22 lists the PMA CSR address field.

Table 22 PMA CSR Address

Address[23:12]	Address[11:0]	Comment
0x000	CSR Offset	Common Register
0x010	CSR Offset	RX register
0x011	CSR Offset	TX register
0x012	CSR Offset	ETH register
0x013	CSR Offset	DFX register

PMA SRAM: `0x2840_0000~0x287f_ffff`. SRAM configuration = 4kx32 (per PHY).

PCS CSR: `0x2881_0000~0x28af_ffff`.

Address bit [21] = broadcast.

Address bit[20:16] = lane ID.

Table 23 lists the PCS CSR address field.

Table 23 PCS CSR Address

Address[23:12]	Address[11:0]	Comment
0x810	CSR Offset	Lane 1
0x820	CSR Offset	Lane 2
0x830	CSR Offset	Lane 3
0x840	CSR Offset	Lane 4
0x850	CSR Offset	Lane 5 (Host PHY only)
0x860	CSR Offset	Lane 6(Host PHY only)

Table 23 **PCS CSR Address (Continued)**

Address[23:12]	Address[11:0]	Comment
0x870	CSR Offset	Lane 7(Host PHY only)
0x880	CSR Offset	Lane 8(Host PHY only)
0x8a0	CSR Offset	Broadcast

Note: No PCS SRAM

PCIe MAC (EP0..EP3, RC) AXI slave port: MAC CSR and PCIe/CXL access (e.g., NVMe door bell, RC CXL.mem, RC CXL.io)

3.18.2 Synopsys PCIe/CXL Controller DBI

Table 24 **AXI DBI Address Bus Layout**

Access Type	31	30	29:22	21	20	19	18:16	15	14:13	12	11:2	1:0
CDM	0			0	CS2	0	Function	0			1K-DWORD Register Address	0
ELBI	0			1	0	0	Function	0			1K-DWORD Register Address	0
iATU	0			1	1	0	iATU Address					0
DMA	0			1	1	1	DMA Address					0
MSI-X Table	0	1	0	1	1	0	Function	0	8K-DWORD MSI-X Table Address			0
MSI-X PBA	0	1	0	1	1	0	Function	1	8K-DW MSI-X PBA Address			0

a. For more information on bits 18-2, see Table 24

b. Address zero corresponds to the first address of the MSI-X Table/MSI-X PBA PCIe MAC access:

Base address = `0x2900_0000 ~ 0x291f_ffff`

AXI DBI address A[31:30] = 0

AXI DBI address A[29:21] = 0

AXI DBI address A[20:0] = A[20:0]

Figure 46 EP Mode Controller Configuration Space Layout (CS=2)

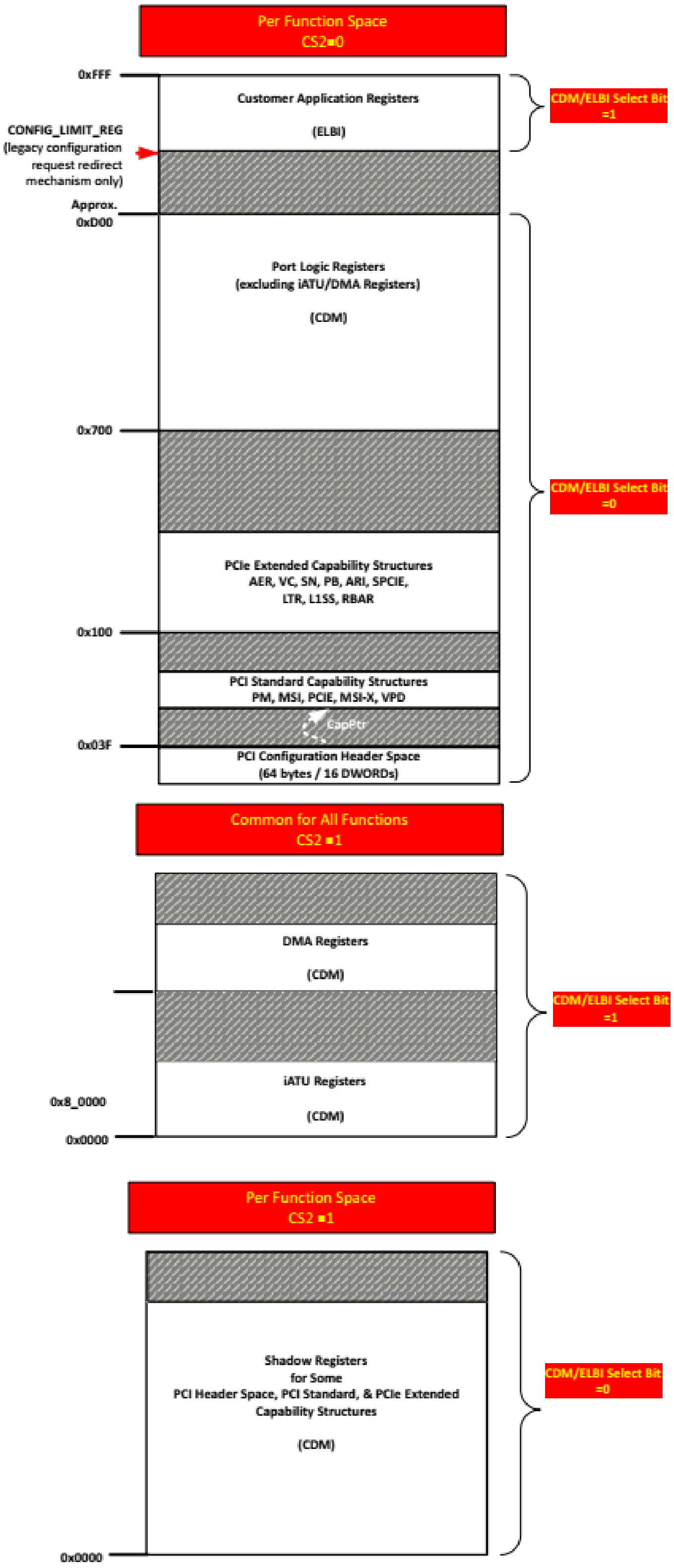


Table 25 lists the PCIe/CXL PCIe/CXL interfaces

Table 25 PCIe/CXL Interfaces

Interface	MAC's role	Protocol	Clock	Data Bus	Description
EP cxl.mem	Master	Native	Synchronous	512b	CXL-AXI bridge to FLC1
EP cxl.io	Master	AXI	Asynchronous	512b	AXI ID = 8b
EP cxl.mem	Slave	Native	Synchronous	512b	Not used
EP cxl.io	Slave	AXI	Asynchronous	512b	AXI ID = 0b
RC cxl.mem	Master	Native	Synchronous	512b	Not used
RC cxl.io	Master	AXI	Asynchronous	512b	AXI ID = 8b
RC cxl.mem	Slave	Native	Synchronous	512b	CXL-AXI bridge to NoC
RC cxl.io	Slave	AXI	Asynchronous	512b	AXI ID = 8b

Note: “synchronous” mode: PCIe/CXL MAC does not include CDC layer.

EP cxl.io master ID width from `pcie_iip_device_wrapper` is 8 also NoC it is 8 bits.2024 FLC Technology Group Inc. Confidential 104

EP cxl.io slave ID width of controller is 8 but at NoC AWID/ARID are not present, so IDs are tied off at controller.

EP cxl.mem master id with is 10, 9bits are to handle actual CXL transactions, 10th is for

demuxing(splitting transactions at `mkb_axi_cluster_demux`), as there are two `FLC1_mc_tops`.

10th bit is generated based on `flc_addr_mask` (offset: 0x510, 0x50c).

```
10th bit = ^({12'b0,cxl_mst_m2s_req.addr[$bits(cxl_mst_m2s_req.addr)-1:1],6'd0} &
cfg_flc_addr_mask);
```

3.18.3 ELBI Access

For type 3 CXL device below registers are to be implemented outside of the controller and

through ELBI interface.

1. HDM Capability is to be implemented. - BAR0
2. CXL Device Registers (mailbox) -BAR0
3. DOE Capability Registers – CFG space.
4. Intel DVSEC – CFG Space
5. PCIe DVSEC for Test Capability - CFG Space, (Address Registers are in mem space but, we are not reporting/implementing any test capability (so all 0s)).

ELBI access is required for CXL2.0.mem to implement CXL HDM decoder. MAC glue logic will route ELBI access to SRAM. Address is determined by: {slice#[1:0], ELBI_SRAM_OFFSET[10:0]}. Table 26 describes ELBI access.

Table 26 ELBI Access

Registers	SRAM Address Range	ELBI Address
Memory mapped (BAR0) CXL Capability Registers	0-15	0x1010-0x101f
Memory mapped (BAR0) HDM decoder Capability and other Capability can be inserted here.	16-511	0x1200- (1200+496 bytes)
CFG mapped, any DVSEC can be inserted here.	512-512+256 bytes	BAR=3'b111, address starts from 0xd00 of config address, (Intel NDR, Test DVSEC)
BAR1 mapped, for application registers (Here also BAR is configurable	768 ~ 768+256	BAR is configurable, size is configurable (16Bytes) granularity. Default BAR1 is decoded here, size is 256bytes.
BAR is configurable	1KB-2KB	64bytes of CXL Device Capabilities Registers and CXL Status Registers (size 8bytes) Mailbox (512+32bytes) CXL Memory Device Capabilities (8bytes) Total = 616bytes

Total SRAM area occupied = 2KB/EP, total 4 endpoints = 8KB. ELBI_BASE is defined in μC CSR “SYS_CTRL” 0x26000008, bit[26:24] with 8KB allocation

- Register: SYS_CTRL, Address: 0x26000008

Bits	Bit Name	Default	Type	Comment
31:28	elbi_acc_flag[3:0]	16’h0	R/W	Set to “1” when ELBI accessed. Write 1 to clear
27	Reserved	4’h0	R/W	Reserved
26:24	elbi_sram_base	12’h0	R/W	ELBI access base address in SRAM with 8KB allocation

Table 27 lists the SRAM address.

Table 27 SRAM Address

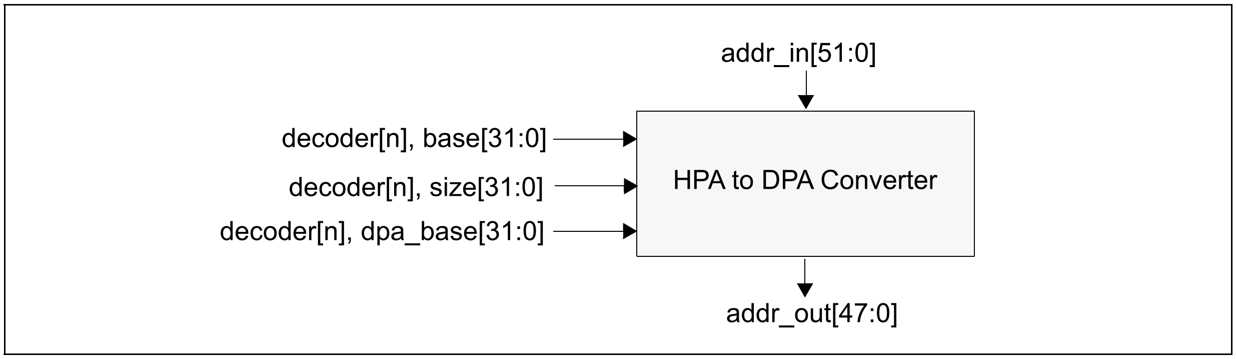
A[31:16]	A[15:13	A[12:11	A[10:0]
0x4000	elbi_sram_base	Slice#	offset

When there is an ELBI access, elbi_acc_flag[EP] is set to “1” and μC can be notified via interrupt. μC can write “1” to corresponding elbi_acc_flag bit to clear the latched flag.

3.18.4 HDM Decoder Format

Figure 46 illustrates the HDM decoder format.

Figure 47 HDM Decoder Format



3.18.5 Reference from MAC Specification

Table 29 lists the CXL 2.0 Component Specific Register Space.

Table 28 CXL 2.0 Component Specific Register Space

Range	Size	Destination	Implementation
0000_0000 to 0000_0FFFh	4K	CXL.io registers	External (ELBI)
0000_1000h to 0000_100Fh	4K	CXL.cache and CXL.mem registers	Internal
0000_1010h to 0000_101Fh			External (ELBI)
0000_1020h to 0000_11FFh			Internal
0000_1200h to 0000_1FFFh			External (ELBI)
0000_2000h to 0000_DFFFh	48K	Implementation specific registers	External (ELBI)
0000_E000h to 0000_E3FFh	1K	CXL ARB/ MUX Registers	Internal
0000_E400h to 0000_FFFFh	7K	Reserved	External (ELBI)

a. The output signal `lbc_ext_cxl_mbar0_access` indicates access to ELBI mapped regions of CXL 2.0 Port Specific Component Registers.This signal can be used along with lbc_ext_addr[15:0] to decode access to ELBI mapped regions of CXL 2.0 Port Specific Component Registers.

b. This address range is implemented on ELBI for `CXL_HDM_Capability_Header`, `CXL_Extended_Security_Capability_Header`, `CXL_IDE_Capability_Header` and `CXL_Snoop_Filter_Capability` registers.

c. This address range is implemented on ELBI for `CXL_HDM_Capability`, `CXL_Extended_Security_Capability`, `CXL_IDE_Capability` and `CXL_Snoop_Filter_Capability` registers

Figure 48 ELBI Transaction: 32-bit Write Access to External Registers

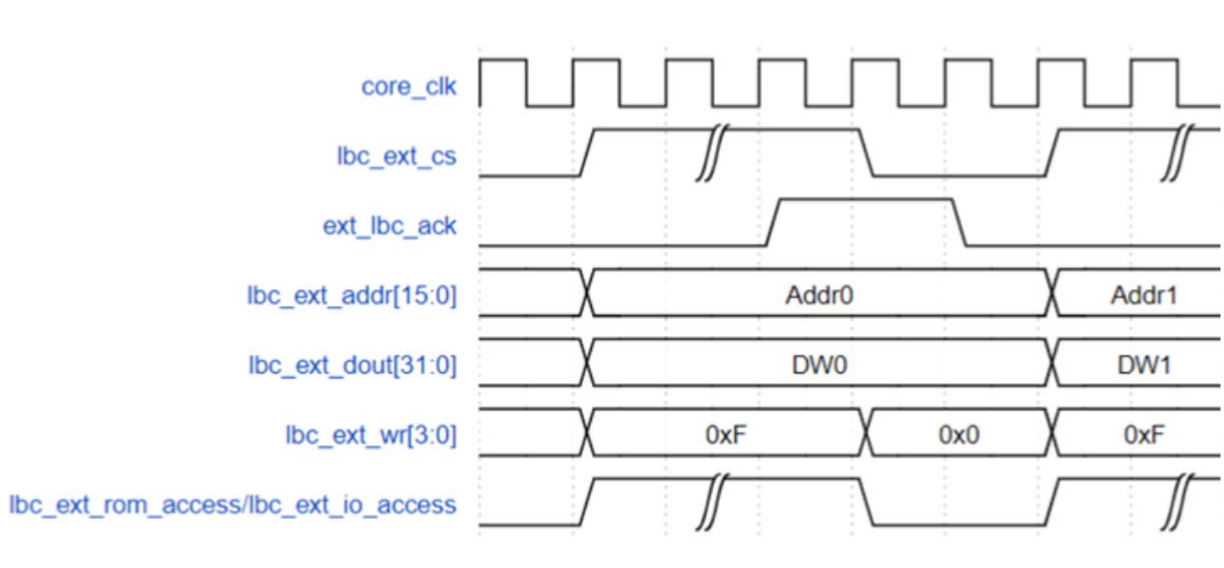
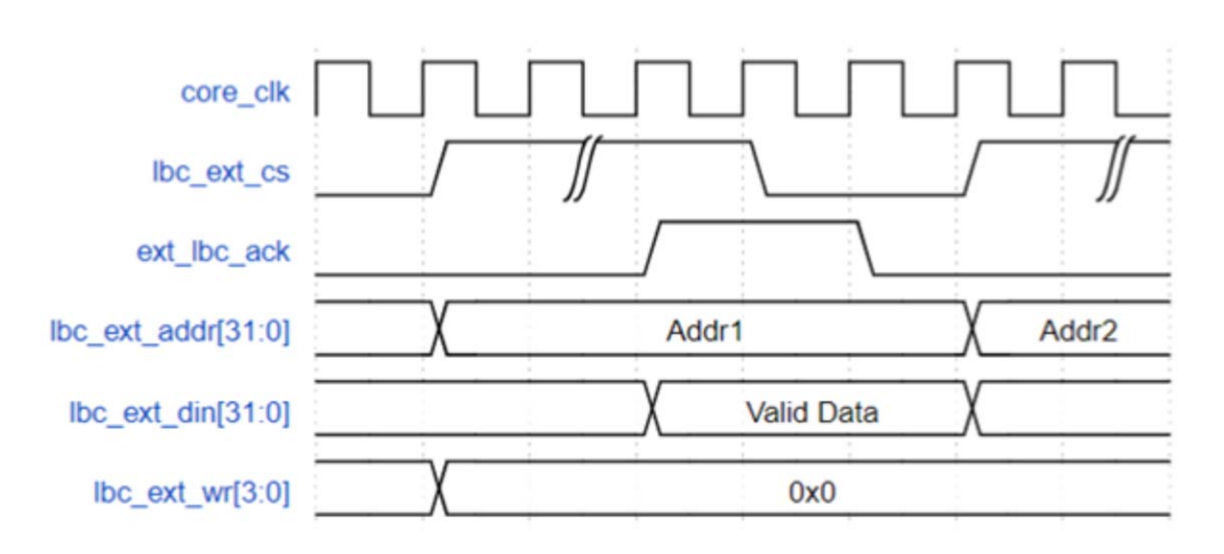


Figure 49 ELBI Transaction: 32-bit Read Access to External Registers



3.18.6 Clocks info figures

Figure 50 PCIe_IIP Device Clocking

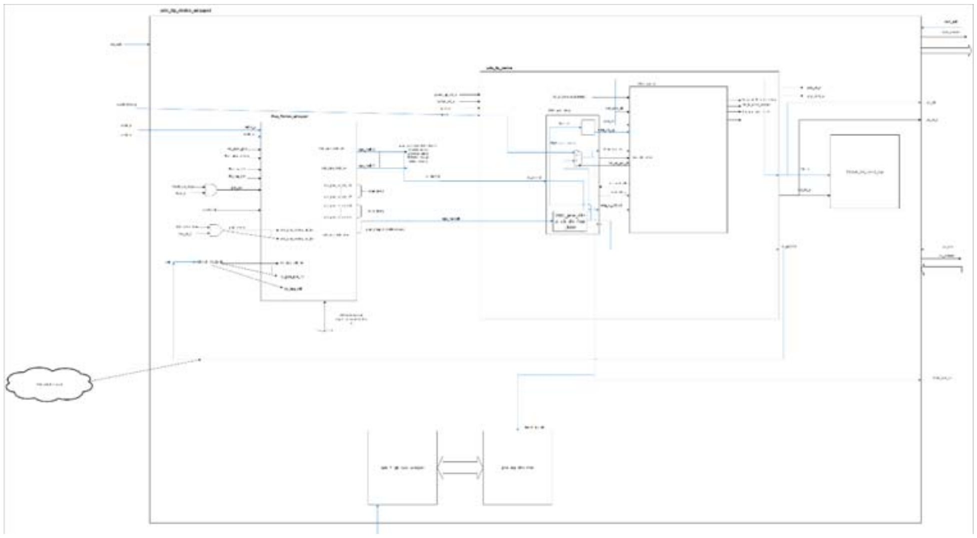
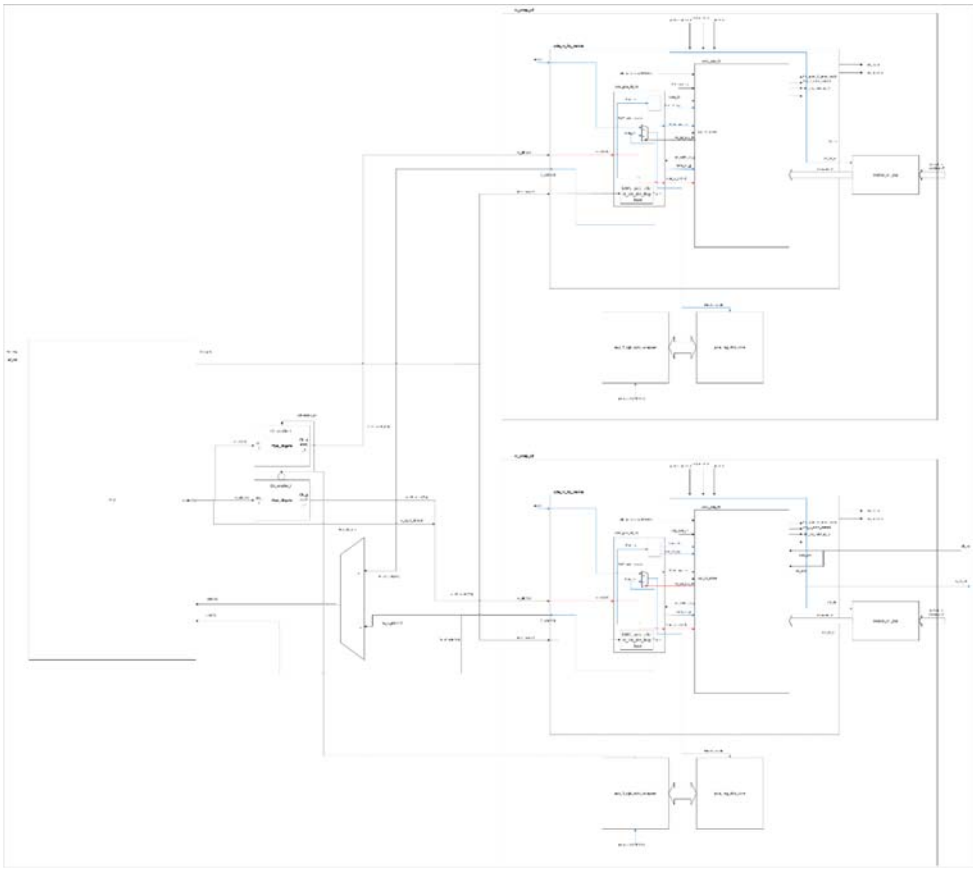


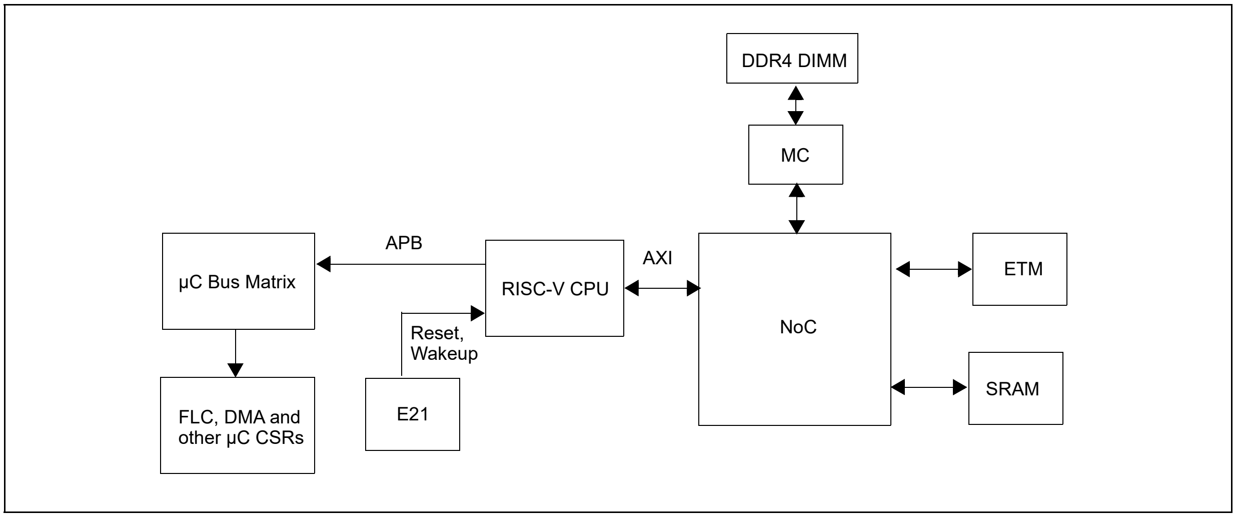
Figure 51 RC_WRAP Clocking



3.19 RISC-V AE350 Subsystem

Figure 52 illustrates the RISC-V CPU.

Figure 52 RISC-V CPU



RISC-V OS image in DDR4 DIMM. Bare-metal test image can be in SRAM or DDR4. Address space: `0xF3000000~0xFFFFFFFF` mapped to FLC CSR. Based on E21 address map. Table 30 lists the address space.

Table 29 Address Space

RISC-V	E12	Description
0xF3xx_xxxx	0x21xx_xxxx	QM
0xF4xx_xxxx ~ 0xF7xx_xxxx	0x22xx_xxxx ~ 0x25xx_xxxx	FLC1, FLC2, IPM, DMA
0xF8xx_xxxx	0x26xx_xxxx	μC sysctrl
0xFAxx_xxxx	0x28xx_xxxx	PCIe PHY
0xFBxx_xxxx	0x29xx_xxxx	PCIe MAC
0xFCxx_xxxx	0x2axx_xxxx	ETM

Table 30 lists the AE350 Memory Map.

Table 30 AE350 Memory Map

Address Range		Description
0x0000_0000	0x7FFF_FFFF	RAM Bridge
0x8000_0000	0x87FF_FFFF	SPI1/AHB Memory/Reset Vector
0xA000_0000	0xA01F_FFFF	Hart 0 Local Memory Slave Port: ILM
0xA020_0000	0xA03F_FFFF	Hart 0 Local Memory Slave Port: DLM
0xA040_0000	0xA05F_FFFF	Hart 1 Local Memory Slave Port: ILM
0xA060_0000	0xA07F_FFFF	Hart 1 Local Memory Slave Port: DLM
0xA080_0000	0xA09F_FFFF	Hart 2 Local Memory Slave Port: ILM
0xA0A0_0000	0xA0BF_FFFF	Hart 2 Local Memory Slave Port: DLM
0xA0C0_0000	0xA0DF_FFFF	Hart 3 Local Memory Slave Port: ILM

Table 30 AE350 Memory Map (Continued)

Address Range		Description
0xA0E0_0000	0xA0FF_FFFF	Hart 3 Local Memory Slave Port: DLM
0xC000_0000	0xC00F_FFFF	BMC
0xC020_0000	0xC02F_FFFF	DFS
0xE000_0000	0xE00F_FFFF	AHB Decoder
0xE050_0000	0xE05F_FFFF	L2C
0xE400_0000	0xE43F_FFFF	PLIC
0xE600_0000	0xE60F_FFFF	Machine Timer
0xE640_0000	0xE67F_FFFF	PLIC-SWINT
0xE680_0000	0xE68F_FFFF	Debug Module / Debug Vector
0xF000_0000	0xF00F_FFFF	APBBRG
0xF010_0000	0xF01F_FFFF	SMU
0xF020_0000	0xF02F_FFFF	UART1
0xF030_0000	0xF03F_FFFF	UART2
0xF040_0000	0xF04F_FFFF	PIT
0xF050_0000	0xF05F_FFFF	WDT
0xF060_0000	0xF06F_FFFF	RTC
0xF070_0000	0xF07F_FFFF	GPIO
0xF0A0_0000	0xF0AF_FFFF	I2C
0xF0B0_0000	0xF0BF_FFFF	SPI1
0xF0C0_0000	0xF0CF_FFFF	DMAC
0xF0F0_0000	0xF0FF_FFFF	SPI2
0xF200_0000	0xF20F_FFFF	DTROM
0x10000_0000	0x1FFFF_FFFF	• Uncacheable alias to region 0x0000_0000 to 0xFFFF_FFFF. • The data in cacheable regions will be cached into L1 caches of the processor. • This region is an uncacheable alias to the cacheable region. Accesses to this region will bypass the L1 caches. • This is useful when the processor and the external bus masters need to share the data in the memory space. • This region is present only when BIU_ADDR_WIDTH is 33 bits

Note:

1. The RAM bridge space indicates the size of the RAM behind this bridge. It can be different from the size of the address space allocated to the bridge on the bus.
2. The default setting allocates 2GiB space (0x0000_0000 to 0xFFFF_FFFF) to the bridge on the bus.
3. When the bridge sees a transaction for addresses outside the addressable RAM, it will return an error response.
4. In addition to the bus view described here, ILM/DLM are accessible by the processor through private address spaces visible only to the processor:
   - The address ranges of these private address spaces are controlled by milmb.IBPA and mdlmb.DBPA.
   - The private address spaces have higher priority than the bus address spaces in the processor. Accesses will be directed to go through the local memory interfaces and bypass the bus address spaces if they hit the private address spaces.
   - If overlapping of address spaces is not desired, milmb, mdlmb control registers could be programmed to avoid overlapping.
   - ILM is visible to the processor at 0x0000_0000 in the default setting.
   - DLM is visible to the processor at 0x0020_0000 in the default setting.
   - Debug module region will be mapped to the default slave when macro PLATFORM_TO_DEBUG_SUPPORT is defined.

0x10000000 = DLM base (256KB). Table 31 lists the Andes core interrupts.

Table 31 Andes Core Interrupts

Sl. No.	Source	Description
1	RTC_PERIOD	
2	RTC_ALARM	
3	PIT	Programmable Interval Timer
4	SPI1	NA
5	SPI2	NA
6	I2C	NA
7	GPIO	
8	UART1	
9	UART2	NA
10	DMA	
15-11	0	
16	L2C_ERR	
17	SSP	NA
28	SDC	NA
29	MAC	NA
20	LCD	NA
21	FLC1	
22	FLC2	
23	IPM_MC	
24	IPM_PHY	
25	AXI_DMA	
26	PCIe PHY	
27	PCIe MAC	
28	ETM	
29	E21	
30	EMAC	
31	0	

3.19.1 RISC-V CPU boot up procedure

1. Download uBoot and OS image from host (via PCIe) to DDR or on-die SRAM
2. As an alternative, E21 can download RISC-V code from QSPI SF (lower performance).
3. E21 setup reset vector in CSR: 0x26000014~0x2600001c.
4. E21 release RISC-V reset.

Sample code: RISC-V core 0 boot from SRAM, address 0x84

```
W 26000018, 0x000c0000 //reset vec base = 0xc0_00000000 (SRAM)

W 26000014, 0x00085001 //reset vec offset = 0x84, release core 0 reset
```

3.19.2 Reset vector calculation

- CSR: `sysctrl_riscv = 0x26000014.`
- CSR: `sysctrl_riscv_rstvec = 0x26000018`
- CSR: `sysctrl_riscv_rstvec1=0x2600001c`
  - — When `sysctrl_riscv[12]=0, "core0_reset_vector" = 0`, else `"core0_reset_vector" = {sysctrl_riscv_rstvec[11:0], sysctrl_riscv[31:24],2'b0}`
  - — When `sysctrl_riscv[13]=0, "core1_reset vector" = 0`, else `"core1_reset_vector" = {sysctrl_riscv_rstvec1[31:2],2'b0}`

To match NoC decoder, high bits of reset vector added to both cores before sending transactions to NoC.

3.19.3 Final address to NoC

• Core0 access = `{sysctrl_riscv_rstvec[26:12],32'h0} + "core0_reset_vector".`

• Core1 access = `{sysctrl_riscv_rstvec[26:12],32'h0} + "core1_reset_vector"`

3.19.4 RISC-V ETM interface

There are 2 interfaces on the ETM module:

1. APB CSR read/write through bus matrix: E21 (0x2a03_[3:0]xxx). Which is also acceptable from RISC-V via "`0xfc03_[3:0]xxx`"
2. RISCV AXI (target) read only through NoC: It uses programmable window in E21 CSR `0x26000010, bit [31:16]`. RISCV issues memory read through this window to download ETM FIFO.

- Register: `RESET_INFO`, Address: `0x26000010`

Bits	Bit Name	Default	Type	Comment
31	riscv_etm_enable	1'h0	R/W	RISCV AXI (target) read only through NoC 1=Enable riscv_etm mapping
27	sysctrl_riscv_etm[30:16]	15'h0	R/W	RISCV-ETM AXI access address [30:16]

For example, set 0x26000010 = 0x80280000, address [15:14] selects ETM#

- RISCV `0x0028_0xxx` access ETM0 (NoC address `0xa0_0000_0xxx`)
- RISCV `0x0028_4xxx` access ETM1 (NoC address `0xa1_0000_0xxx`)
- RISCV `0x0028_8xxx` access ETM2 (NoC address `0xa2_0000_0xxx`)
- RISCV `0x0028_cxxx` access ETM3 (NoC address `0xa3_0000_0xxx`)

3.20 QM SRAM Access

QM’s submission queue (SQ) and completion queue (CQ) are placed in a SRAM (size: 8KB) within QM module. This SRAM is mapped into QM’s address space and QM’s queue handler SM will access this SRAM via APB. We need to provide a path from PCIe RC to access this SRAM (read/write queue contents), the path is:

- PCIe RC (AXI master) ->
- NoC ->
- µC bus matrix ->
- QM’s APB slave port ->
- QM SRAM

Note: The same PCIe RC’s AXI master interface also carries data transfer, which is:

- PCIe RC (AXI master) ->
- NoC à DDR4 MC

(NoC will route traffic based on RC’s address to either MC or µC…)

Note: QM SRAM address is `0xc0_21fx_xxxx` (1MB).

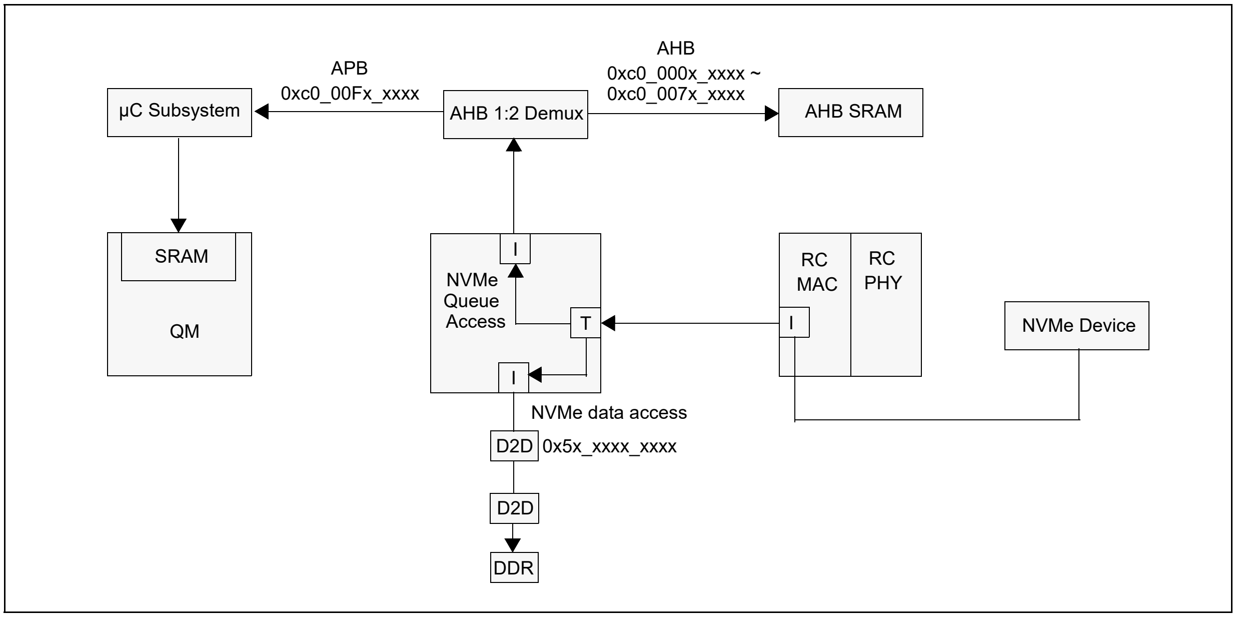
`QM0=0xc0_21f0_xxxx` (64KB)

`QM1=0xc0_21f1_xxxx` (64KB)

`QM2=0xc0_21f2_xxxx` (64KB)

`QM3=0xc0_21f3_xxxx` (64KB)

Figure 53 Mockingbird NVME Access Scheme



Mockingbird NVME access scheme 2:

- Reuse NoC’s AHB SRAM port
- Add AHB de-mux and attach to µC system’s APB target port

3.21 AHB SRAM

AHB SRAM connected to NoC; size16Kx32 (total=64KB), running at NoC clock (=1Ghz). It is shared by various units and served as system storage. E21 FW manages SRAM usage via CSR.

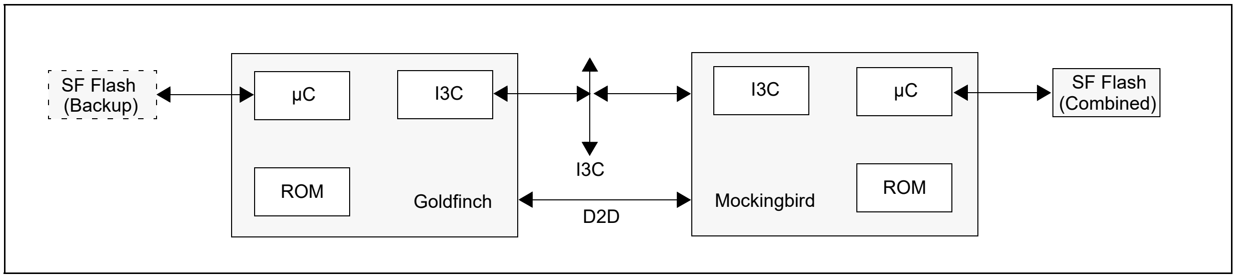
Table 32 AHB SRAM

Offset	Size	Usage	CSR	Comment
0xc000~0xffff	16K	Reserved		For Andes boot code
0xa000~0xbfff	8K	Reserved		For RC use
0x8000~9fff	8K	PCIe EP config / MMIO registers	(SYS_CTRL) 0x26000008 ELB_SRAM_BASE	PCIe MAC AXI interface, 2K per endpoint for CXL.io. Slice# appended to A[12:11]
0x4000~0x7fff	16K			For management queue use. Data and response queues are handled by QM
0x80~0x3fff	~16K	Reserved		
0~0x7f	128K			16B/message per device.8x16B=128B message buffer for EP (4x) or RC (4x)

3.22 Chiplet Flash Configuration

Figure 54 illustrates the chiplet flash configuration.

Figure 54 Chiplet Flash Configuration



- Internal ROM for I3C and D2D init.
- Combined SF flash for all chiplet's firmware.
- Goldfinch µC read chiplet_specific firmware through D2D or I3C.
- Keep SF interface for chiplets other than Mockingbird as backup option.

Table 33 FW Init Sequence

Step	Size	Source	Description
1	Power up / reset	ROM and SF Flash	Each chiplet's µC fetch D2D and I3C initialization code from internal ROM.
2	D2D init		µC programs D2D and I3C. I3C is ready after this step.
3	D2D Sync	I3C	µC trains inter-D2D timing with I3C. D2D is ready after this step
4	Chiplet init	SF Flash	For Mockingbird: µC fetch init code from SF Flash and perform FLC, PCIe, IPM init. For other chiplets: Mockingbird µC executes chiplet system init code through D2D. I3C used for interrupt messages.

QSPI connected to Mockingbird and stores PHY FW, FSBL and APP code for Mockingbird system. Table 34 is based on 32MB QSPI.

Table 34 QSPI Flash Table

Offset	Size	Usage	Description
0x0		I3C Driver	
		D2D Driver	
	128K	...	Reserved for other drivers
0x20000	256K	AW PHY FW	
0x60000	256K	IPM PHY FW	
0xa0000	256K	DDR PHY FW	
0xe000	128K		Reserved
0x1_00000	1MB	Free RTOS	E21
0x2_00000	29MB	Linux/Bare metal	Andes
0x1f_00000	1MB	Memory repair	

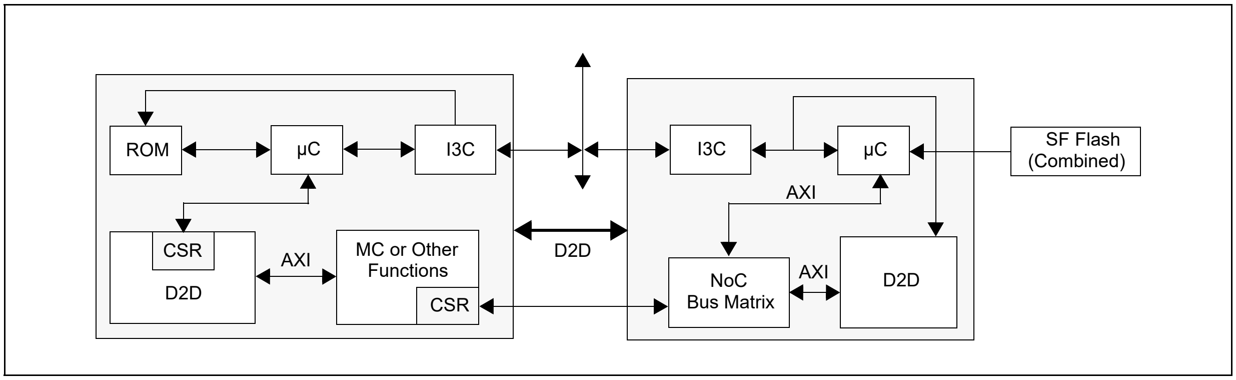
QSPI connected to Mockingbird and stores PHY FW, FSBL and APP code for mockingbird system. Table 35 based on 256MB QSPI.

Table 35 256MB QSPI

Offset	Size	Usage	Description
0x0		I3C Driver	
		D2D Driver	
	128K	...	Reserved for other drivers
0x20000	256K	AW PHY FW	
0x60000	256K	IPM PHY FW	
0xa0000	256K	DDR PHY FW	
0xe000	128K		Reserved
0x1_00000	1MB	Free RTOS	E21
0x2_00000	238MB	Linux/Bare metal	Andes
0x1f_00000	16MB	Memory repair	

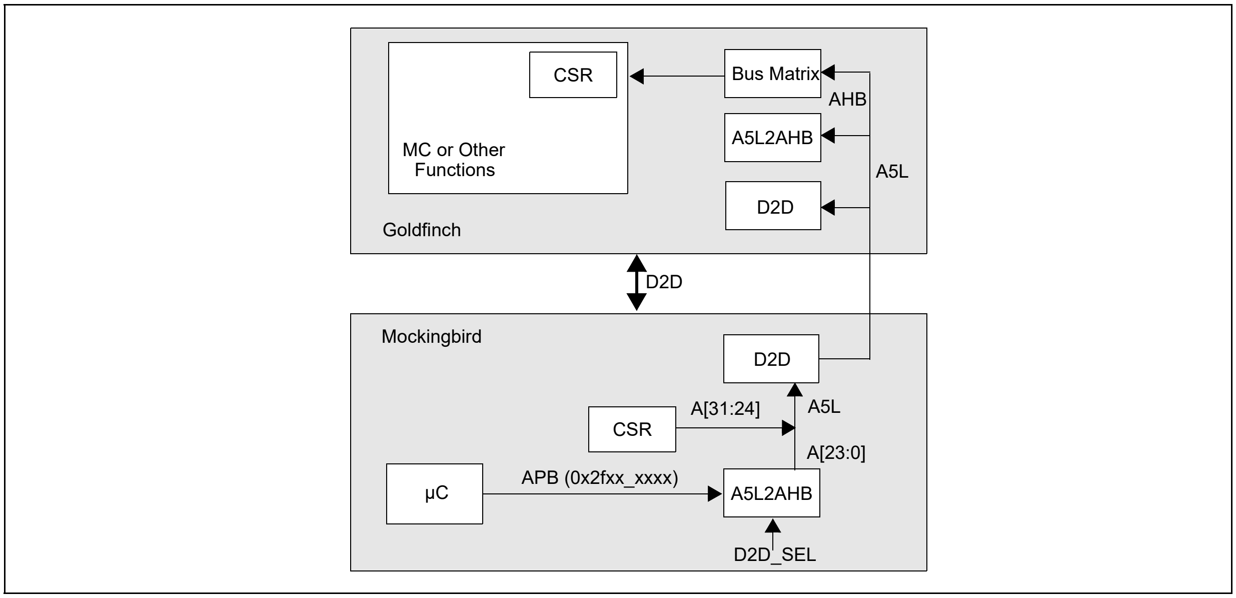
.Figure 55 and Figure 56 illustrate the transaction flow from MKB to other chiplets

Figure 55 Transaction Flow



A5L2Ahb: convert AXI5 Lite protocol to AHB,CDC (1GHz ->200 Mhz)

Figure 56 Detailed Block Diagram



Apb2A5L: convert apb protocol to AXI5 Lite, CDC (200 Mhz ->1GHz)

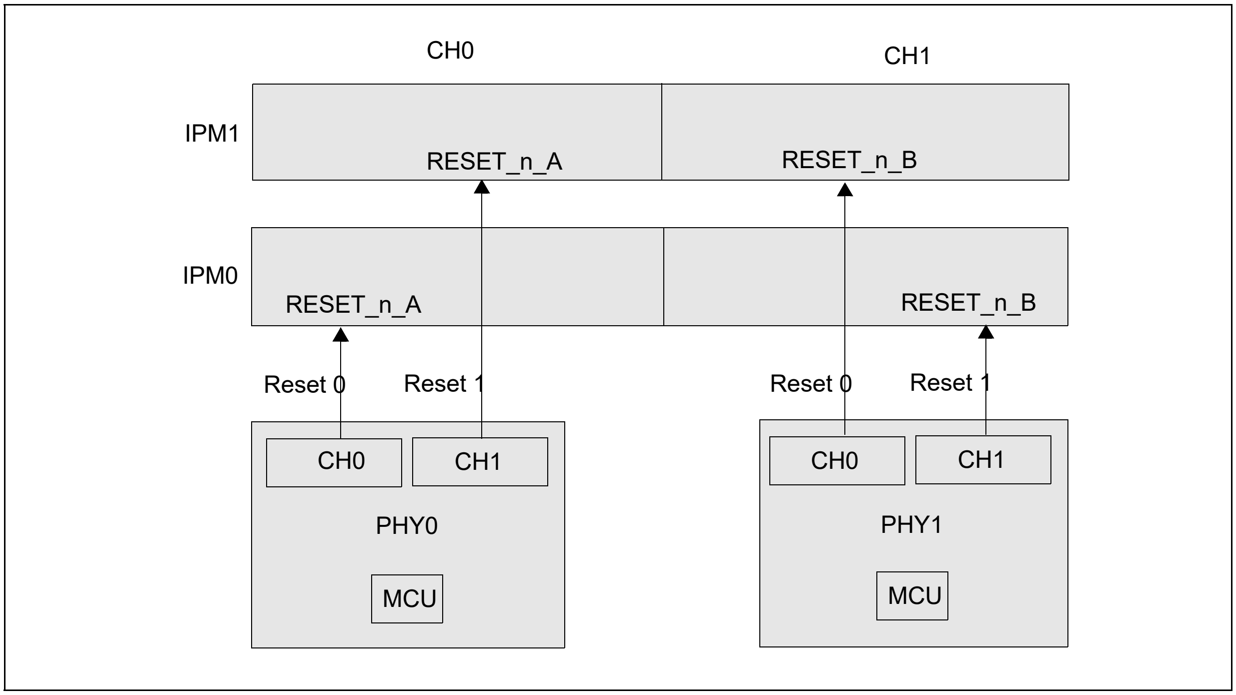
D2D_SEL:0x26000038[11:8]. Select one D2D interface to access

CSR: 0x26000038[19:12] (D2D_IOA31x24). Replace upper address bits[31:24] to allow full range access

3.23 Interleaved IPM

IPM PHY and memory connections are interleaved for balanced RDL routing. When IPM1 is not installed, only ch0 of each PHY is initialized. Figure 57 illustrates the Interleaved IPM module.

Figure 57 Interleaved IPM

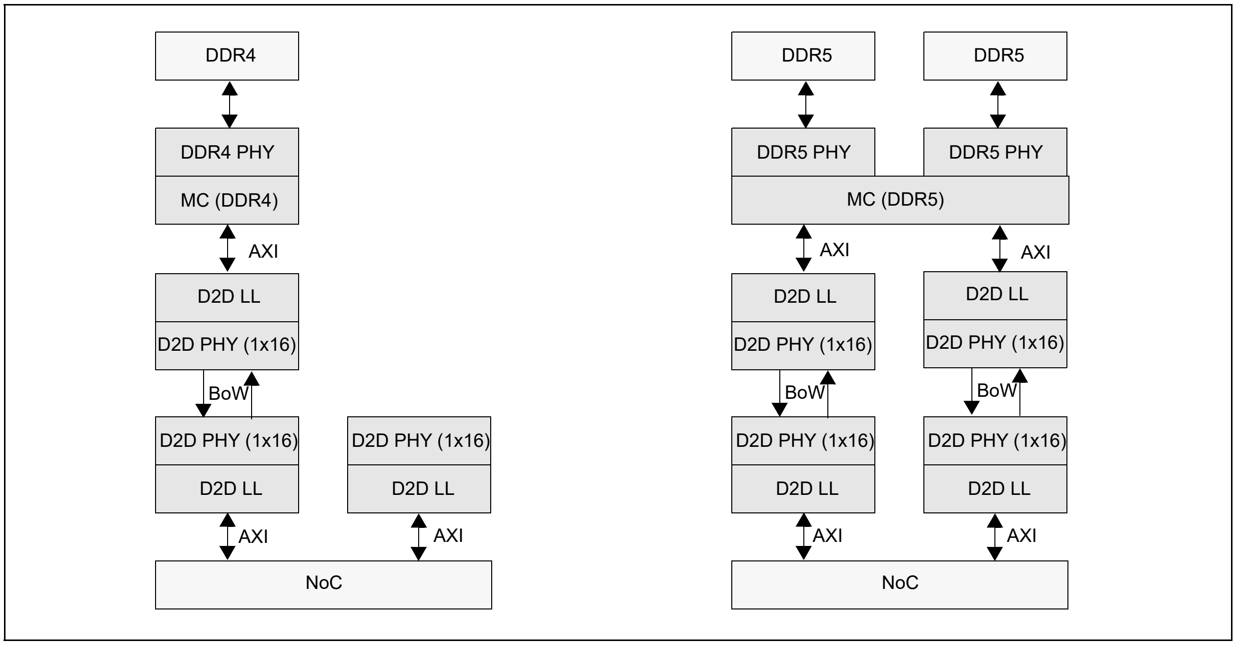


IPM init sequence with PHY MCU

1. VDD=1, PHY.boot_i=1, PHY.mcu_core_rstn=0, MC.clk_resetn=0
2. E21 load FW into PHY (via APB bus).
3. PHY.mcu_core_rstn=1 (E21 programs CSR 0x26000064, bit 0 = 1)
4. Wait 0.5ms
5. PHY.boot_i=0, MC.clk_resetn=1(E21 programs CSR 0x26000064, bit 5 = 0)
6. MC csr init (by E21)

Figure 58 illustrates the D2D interface between MKB and GFH.

Figure 58 MKB D2D Interface to GFH



256 b AXI @ 1Ghz, peak BW = 256b/s

1x16 D2D peak BW = 256b/s

To support 2x16 D2D, 2x AXI bus to e used for NoC connection.

E21 FW controls enable/disable of 2nd D2D LL/PHY

DDR5 MC supports 2 AXI interfaces, with airbitration among DDR5 channels.

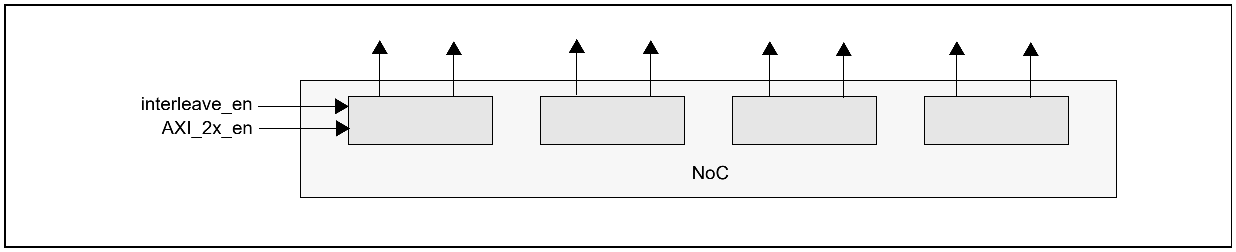
Client chiplet (LPDDR5)CMN option: single 256b AXI @ 2GHz

noc_cfg: E21 CSR: 0x26000008 bit [23:15] controls NoC interleave and other options.

- `[23:20] = MemIntlvWays`, default 4'b0100
- `[19] = GFH3En`, default 1'b1
- `[18] = GFH2En`, default 1'b1
- `[17] = GFH1En`, default 1'b1
- `[16] = GFH0En`, default 1'b1
- `[15] = DualMemChEn`, default 1'b0

Figure 59 illustrates the NoC module for MKB-GFH D2D interleave.

Figure 59 NoC for MKB-GFH Interleave



Interleave Enable	AXI_2x_en
00: Interleave disabled	0: Single AXI link
01: Interleave 2 targets	1: Dual AXI link
10: Interleave 4 targets	

4.0 Address Space

This chapter describes the Address Space utilized by the various programming modules present within the Memory Expander.

4.1 NoC Address Map

- 52 bit address
- Fixed map at NoC
- Programmable system address region via external CSR

The Table 36 lists the NoC address map.

Table 36 NoC Address Map

Device	Size	Start	End	Comment
		0	0x9f_fffffff	Reserved
ETM0	16K	0xa0_00000000	0xa0_00003fff	
ETM1	16K	0xa1_00000000	0xa1_00003fff	
ETM2	16K	0xa2_00000000	0xa2_00003fff	
ETM3	16K	0xa3_00000000	0xa3_00003fff	
SRAM	16M	0xc0_00000000	0xc0_00ffffff	Actual SRAM size may be smaller
FLC2	1T	0x0_0100_00000000	0x0_01ff_ffffffff	FLC2 slice 0 slave address
FLC2	1T	0x0_0200_00000000	0x0_02ff_ffffffff	FLC2 slice 1 slave address
FLC2	1T	0x0_0300_00000000	0x0_03ff_ffffffff	FLC2 slice 2 slave address
FLC2	1T	0x0_0400_00000000	0x0_04ff_ffffffff	FLC2 slice 3 slave address
		0x0_0500_00000000	0x0_05ff_ffffffff	Reserved
D2D_0	256T	0x1_0000_00000000	0x1_ffff_ffffffff	Inter-chiplet
D2D_1	256T	0x2_0000_00000000	0x2_ffff_ffffffff	Inter-chiplet
CXL_PCIe_EP0	1T	0x0_4000_00000000	0x0_40ff_ffffffff	Host0 CXL_PCIe AXI slave address
CXL_PCIe_EP1	1T	0x0_4100_00000000	0x0_41ff_ffffffff	Host1 CXL_PCIe AXI slave address
CXL_PCIe_EP2	1T	0x0_4200_00000000	0x0_42ff_ffffffff	Host2 CXL_PCIe AXI slave address
CXL_PCIe_EP3	1T	0x0_4300_00000000	0x0_43ff_ffffffff	Host3 CXL_PCIe AXI slave address
		0x0_4400_00000000	0x0_4fff_ffffffff	Reserved
D2D_DDR0	1T	0x0_5000_00000000	0x0_50ff_ffffffff	Goldfinch0
D2D_DDR1	1T	0x0_5100_00000000	0x0_51ff_ffffffff	Goldfinch1

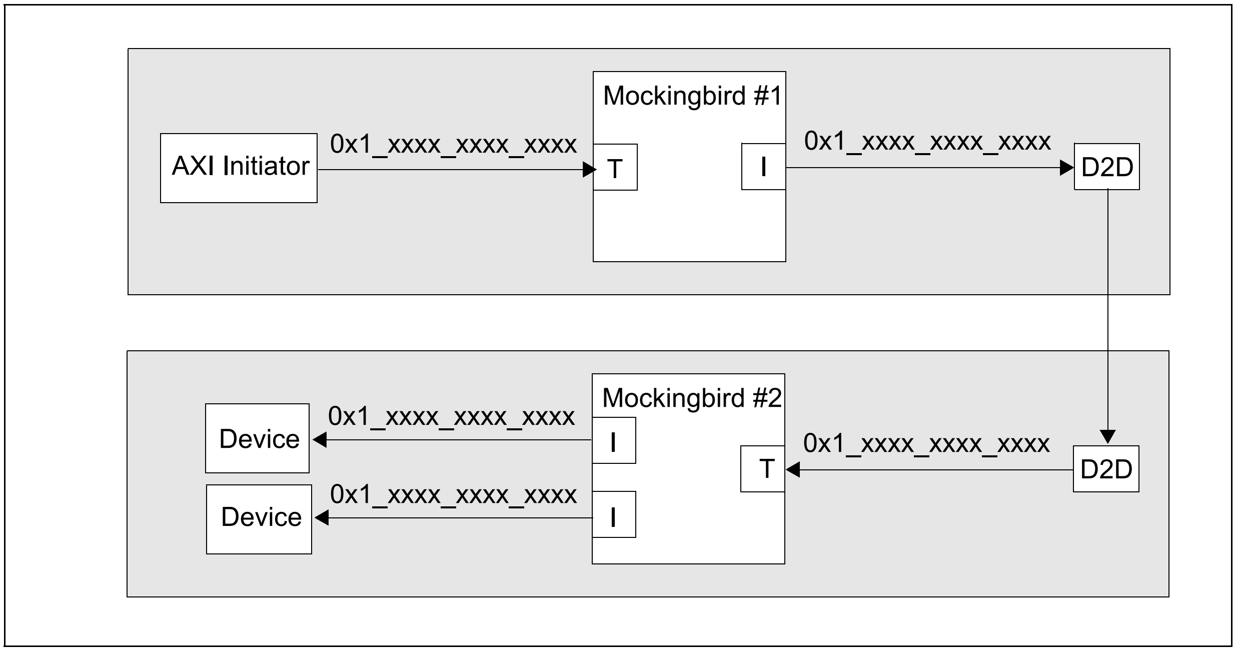
Table 36 NoC Address Map (Continued)

Device	Size	Start	End	Comment
D2D_DDR1	1T	`0x0_5100_00000000`	`0x0_51ff_ffffffff`	Goldfinch1
D2D_DDR2	1T	`0x0_5200_00000000`	`0x0_52ff_ffffffff`	Goldfinch2
D2D_DDR3	1T	`0x0_5300_00000000`	`0x0_53ff_ffffffff`	Goldfinch3
	256T	`0x8_0000_00000000`	`0x8_ffff_ffffffff`	Reserved
	256T	`0x9_0000_00000000`	`0x9_ffff_ffffffff`	Reserved
	256T	`0xa_0000_00000000`	`0xa_ffff_ffffffff`	Reserved
	256T	`0xb_0000_00000000`	`0xb_ffff_ffffffff`	Reserved
CXL_PCIe_RC1_x8	8T	`0x0_c000_00000000`	`0x0_c7ff_ffffffff`	Downstream CXL.mem (RC1_x8)
CXL_PCIe_RC1_x8	8T	`0x0_c800_00000000`	`0x0_cffff_ffffffff`	Downstream CXLio/PCIe (RC1_x8)
CXL_PCIe_RC1_x4	8T	`0x0_d000_00000000`	`0x0_d7ff_ffffffff`	Downstream CXL.mem (RC1_x4)
CXL_PCIe_RC1_x4	8T	`0x0_d800_00000000`	`0x0_dffff_ffffffff`	Downstream CXLio/PCIe (RC1_x4)
CXL_PCIe_RC0_x8	8T	`0x0_e000_00000000`	`0x0_e7ff_ffffffff`	Downstream CXL.mem (RC0_x8)
CXL_PCIe_RC0_x8	8T	`0x0_e800_00000000`	`0x0_effff_ffffffff`	Downstream CXLio/PCIe (RC0_x8)
CXL_PCIe_RC0_x4	8T	`0x0_f000_00000000`	`0x0_f7ff_ffffffff`	Downstream CXL.mem (RC0_x4)
CXL_PCIe_RC0_x4	8T	`0x0_f800_00000000`	`0x0_fffff_ffffffff`	Downstream CXLio/PCIe (RC0_x4)

4.1.1 Inter-Chiplet Transaction Mapping

NoC D2D0/D2D1 (inter-chiplet) AXI master interface is assigned address space as `0x1_xxxx_xxxx_xxxx/0x2_xxxx_xxxx_xxxx`. When transactions passing through D2D to reach targeting chiplet, it'll be decoded as `0x_0_xxxx_xxxx_xxxx` for devices within. Figure 60 illustrates the inter chiplet transaction mapping.

Figure 60 Inter Chiplet Transaction Mapping

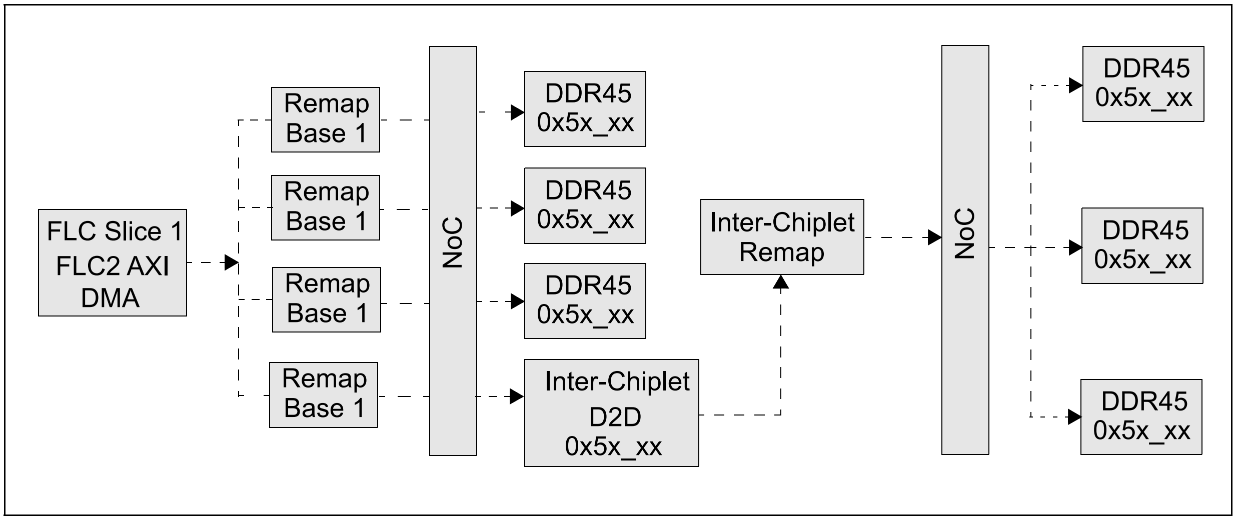


Address remap from AXI/AHB master to NoC slave port

- Modify request address from masters to fit (fixed) NoC map.
- Run time programmable CSR for memory pooling.
- Logical address from host vs chiplet memory partition.
- Expand address bit: µC (32) -> NoC (52).
- Applied on master interfaces: FLC2 (AXI), RC master and µC master (AHB).
- CSR fields
  - — Master BAR (base, size): to compare address field with incoming request.
  - — Slave BAR (base): new base address for outgoing request.
  - — 4 sets of slave BAR per FLC AXI master to access shared devices: DDR45, inter-chiplet, 3DNAND and CXL_PCIe RC.

Figure 61 illustrates the address remap from AXI/AHB master to NoC slave port.

Figure 61 AXI / AHB master to NoC slave port Mapping



There are 4 sets of mappers for each FLC slice. Defined in register FLC_ADDRESS_REMAP (0x26000080~0x260000ff).

Table 37 FLC Map Allocation

µC Address	FLC Slice	Mapper
0x26000084~80	0	0
0x2600008C~88	0	1
0x26000094~90	0	2
0x2600009C~98	0	3
0x260000A4~A0	1	0
0x260000AC~A8	1	1
0x260000B4~B0	1	2
0x260000BC~B8	1	3
0x260000C4~C0	2	0
0x260000CC~C8	2	1
0x260000D4~D0	2	2
0x260000DC~D8	2	3
0x260000E4~E0	3	0
0x260000EC~E8	3	1
0x260000F4~F0	3	2
0x260000FC~F8	3	3

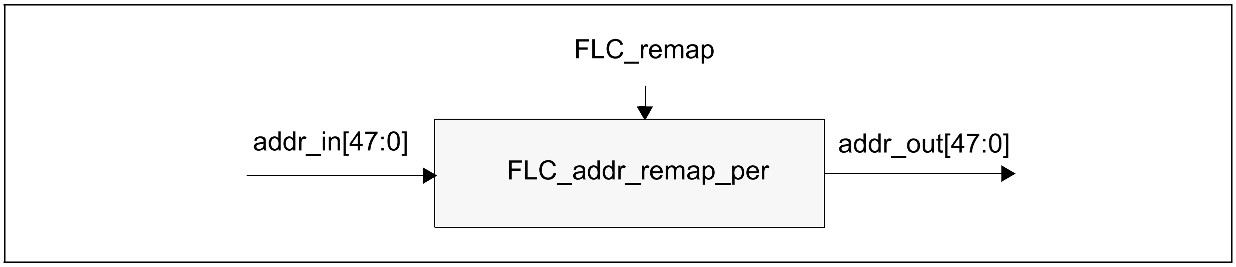
Each mapper register is 64b, remaps memory block of 16MB to a new location.

Table 38 Address Mapper Register

Bits	Bit Name	Default	Type	Comment
63:40	New_base_address	24’b0	R/W	Outgoing (remapped) address A[47:24]
39:36	Reserved	4’b0	R/W	Reserved
35:32	New_base_address_h	4’b0	R/W	Outgoing (remapped) address A[51:48]
31:8	Base_address	24’b0	R/W	Incoming address (to be mapped) A[47:24]
7:6	Reserved	2’b0	R/W	Reserved
5:1	length	5’b0	R/W	Length of mapped region (2’s power of 16MB). 0 = 16MB, 1 = 32MB, 2 = 64MB, …
0	valid	1’b0	R/W	1 = mapper valid

Figure 62 illustrates the HDM decoder.

Figure 62 HDM Decoder



Note: Present flc_addr_remapper which is converting 48-bit address to 52-bit address

```
addr_in_masked = addr_in[47:24] && ({24{1’b1}}<<addr_area_length);
addr_in_offset = addr_in[47:24] & (~addr_mask_base);
addr_out=new_addr_base + addr_in_offset;
```

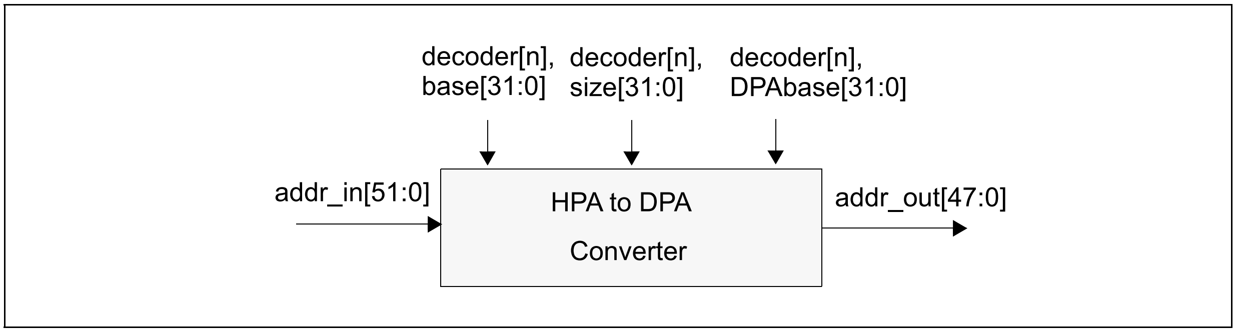
Here granularity is 16 MB, so offset is calculated after ignoring 24 lower bits

Table 39 Address Map Format

Address Map Format	
63:40	New base address
39:36	Reserved
35:32	New base
31:8	Base Address
7:6	Reserved
5:1	Address map area length
0	Address map valid
[51:48]	Address

Figure 63 illustrates the HPA to DPA converter.

Figure 63 HPA to DPA Converter



decoder[n].base[31:0] only 24 bits used out of 32 bits

decoder[n].size[31:0] only 24 bits used out of 32 bits

decoder[n].DPAbase[31:0] only 20 bits used out of 32 as output addr_out is 48 bits. Lower 28 bits are calculated from incoming address

hdm1_base_addr[51:0]={Decoder[n].base[23:0],28’b0}

hdm1_size[51:0]={Decoder[n].size[23:0],28’b0}

hpa1_offset=addr_in[51:0] - hdm1_base_addr[51:0];

```
for(i=0;i<10;i++) begin

    rang1_hit = hdm1_base_addr[51:0] <= addr_in[51:0]<( hdm1_base_addr[51:0]+ hdm1_size[51:0])

    hpa1_offset = addr_in[51:0]-hdm1_base_addr[51:0];

    if (range1_hit)

        dpa1 = {Decoder[n].DPAbase[19:0],28’b0}+hp1_offset;

end
```

4.2 µC to NoC Map

µC address space is 4G (32b). All µC access to NoC goes through fixed mapping window.

Table 40 µC to NoC Map Address

µC Address window	Address to NoC	Region	Size
0x4000_0000 ~ 0x40ff_ffff	0x00c0_0000_0000 ~ 0x00c0_00ff_ffff	SRAM	16M
0x5000_0000 ~ 0x50ff_ffff	CSR: dbl_rc0_remap	(RCx8) PCIe/NVMe doorbell	MMIO Location within RC’s PCIe space
0x5100_0000 ~ 0x51ff_ffff	CSR: µc_rc0_remap	(RCx8) Downstream CXL/PCIe	16T
0x5200_0000 ~ 0x52ff_ffff	CSR: µc_ep0_remap	upstream CXL/PCIe ep 0	1T
0x5300_0000 ~ 0x53ff_ffff	CSR: µc_ep1_remap	upstream CXL/PCIe ep 1	1T
0x5400_0000 ~ 0x54ff_ffff	CSR: µc_ep2_remap	upstream CXL/PCIe ep 2	1T
0x5500_0000 ~ 0x55ff_ffff	CSR: µc_ep3_remap	upstream CXL/PCIe ep 3	1T
0x5600_0000 ~ 0x56ff_ffff	Reserved		
0x5700_0000 ~ 0x57ff_ffff	Reserved		

Table 40 µC to NoC Map Address (Continued)

µC Address window	Address to NoC	Region	Size
0x5800_0000 ~ 0x58ff_ffff	CSR: dbl_rc0_x4_remap	(RCx4) PCIe/NVMe doorbell	16T
0x5900_0000 ~ 0x59ff_ffff	CSR: µc_rc0_x4_remap	(RCx4) CXL/PCIe	16T
0x5a00_0000 ~ 0x5aff_ffff	CSR: dbl_rc1_x8_remap	(RC1x8) PCIe/NVMe doorbell	16T
0x5b00_0000 ~ 0x5bff_ffff	CSR: µc_rc1_x8_remap	(RC1x8) CXL/PCIe	16T
0x5c00_0000 ~ 0x5cff_ffff	CSR: dbl_rc1_x4_remap	(RC1x4) PCIe/NVMe doorbell	16T
0x5d00_0000 ~ 0x5dff_ffff	CSR: µc_rc1_x4_remap	(RC1x4) CXL/PCIe	16T
0x6000_0000 ~ 0x6fff_ffff	CSR: µc_nic_remap	µC_mapper (DDR45 D2D)	256T (default to 0x5000_0000_0000)
0x7000_0000 ~ 0x7fff_ffff	Reserved		

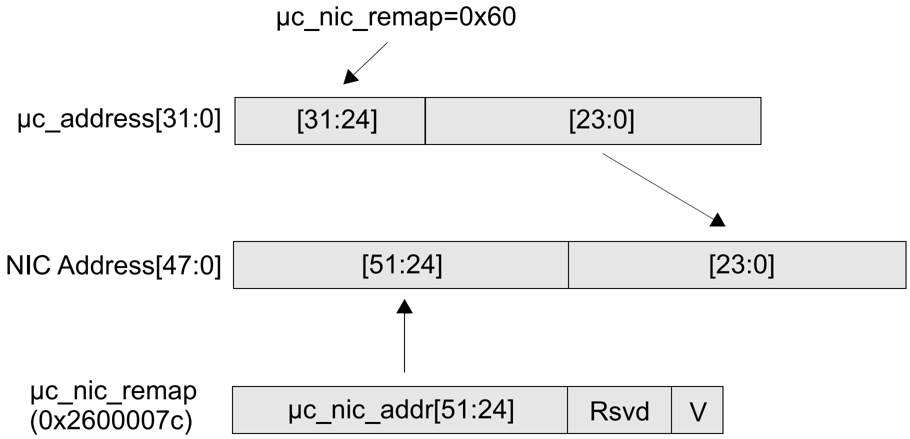
1. A CSR `µc_nic_remap` is added for software to remap µC to NoC access at run time. By default this register is used to map µC access to DDR45 of 1st D2D interface.

Table 41 µc_nic_remap

Address	Name	Bit	Field	Default	Type	Comment
0x2600007c	µc_nic_remap	0	valid	0	RW	1 = remap is valid
		3:1	Reserved	0	RW	Reserved
		27:4	µc_nic_addr_47x24	0x500000	RW	A[47:24] of µc to NoC access
		31:28	µC_nic_addr_51x48	0	RW	A[51:48] of µc to NoC access

The remapping method is shown in the Figure 64.

Figure 64 Remapping



2. CSR: `dbl_rc_remap` is added for software to remap FLC2/QM to RC doorbell access at run time.

   FLC2/QM access window = `0x50xx_xxxx`

System address after remap = `0xf_<dbl_rc_a_43_24>_xx_xxxx`

Table 42 Remap register: dbl_rc_remap

Address	Name	Bit	Field	Default	Type	Comment
0x26000068	dbl_rc_remap	19:0	dbl_rc_a_43_24	0	RW	
		31:20	Reserved	0	-	

3. CSR: `µc_rc_remap` is added for software to remap µC to RC AXI slave port access at run time.

   µC access window = `0x51xx_xxxx`

   System address after remap = `0xf_<µc_rc_a_43_24>_xx_xxxx`

Table 43 Reamp register: µc_rc_remap

Address	Name	Bit	Field	Default	Type	Comment
0x2600006c	µc_rc_remap	19:0	µc_rc_a_43_24	0	RW	
		31:20	Reserved	0	-	

4. CSR: `µc_ep01_remap` is added for software to remap µC to EP0/EP1 AXI slave port access at run time.

   µC access window = `0x52xx_xxxx`

   System address after remap = `0x40_<µc_ep0_a_39_24>_xx_xxxx`

   µC access window = `0x53xx_xxxx`

   System address after remap = `0x41_<µc_ep1_a_39_24>_xx_xxxx`

Table 44 Remap register: µc_ep01_reamp

Address	Name	Bit	Field	Default	Type	Comment
0x26000070	µc_ep01_remap	15:0	µc_ep0_a_39_24	0	RW	
		31:16	µc_ep1_a_39_24	0	RW	

5. CSR: `µc_ep23_remap` is added for software to remap µC to EP2/EP3 AXI slave port access at run time.

   µC access window = `0x54xx_xxxx`

   System address after remap = `0x42_<µc_ep2_a_39_24>_xx_xxxx`

   µC access window = `0x55xx_xxxx`

   System address after remap = `0x43_<µc_ep3_a_39_24>_xx_xxxx`

Table 45 Remap register: µc_ep23_remap

Address	Name	Bit	Field	Default	Type	Comment
0x26000074	µc_ep23_remap	15:0	µc_ep2_a_39_24	0	RW	
		31:16	µc_ep3_a_39_24	0	RW	

4.3 PHY JTAG Connection

The MKB has 3 PHY JTAG chains (IPM, PCIe, D2D). These are shared between `TMS, TCK, nTRST. TDI` and `TDO` are selected via `PAD_GPIO_[8:4]`.

Table 46 TDI / TDO Selection via GPIO

GPIO[8:4]	TDI / TDO	GPIO[8:4]	TDI / TDO
0	Slice 0, IPM0	14	PCIe EP3 PCS
1	Slice 0, IPM1	15	PCIe EP3 PMA
2	Slice 1, IPM0	16	PCIe RC0 PCS
3	Slice 1, IPM1	17	PCIe RC0 PMA
4	Slice 2, IPM0	18	PCIe RC1 PCS
5	Slice 2, IPM1	19	PCIe RC1 PMA
6	Slice 3, IPM0	20	D2D_DDR0 (GFH0)
7	Slice 3, IPM1	21	D2D_DDR1 (GFH1)
8	PCIe EP0 PCS	22	D2D_DDR1 (GFH2)
9	PCIe EP0 PMA	23	D2D_DDR2 (GFH3)
10	PCIe EP1 PCS	24	MKB D2D0
11	PCIe EP1 PMA	25	MKB D2D1
12	PCIe EP2 PCS	26	TAG_APB_bridge
13	PCIe EP2 PMA		

D2D PHY JTAG loopback test: Apply `GPIO[15:17]` to enable D2D LL and wrapper during JTAG test sequence.

Conditions in D2D loopback test mode (`CORERSTn==0`):

- `GPIO[15] -> EXT_PHY_RST_EN`
- `GPIO[16] -> EXT_TX_PHY_RSTN`
- `GPIO[17] -> EXT_RX_PHY_RSTN`

Table 47 JTAG APB TDR Register Bit Field

Bits	Type	Description
IR	Data reg =1 (write) Data reg =1 (write)	
DR	Data and status reg format	
[71]	reset_n	Reset JTAG bridge
[70]	start	Set high to trigger APB write/read command, Set low before start new APB transaction.
[69:67]	pprot	APB used
[66]	pwrite	APB used
[65:34]	paddr	APB used
[33:2]	pwdata/prdata	APB used
[1]	pslverr	APB used
[0]	busy	Status of APB write/read command

Steps:

1. Toggle `ntrst.` `TMS=0`. `start` in TDR reset to “0”.
2. Write TDR: {`address, pprot, pwrite..`}, set `start=1` to start APB cycle.
3. Read TDR, polling for `busy=0;`
4. `busy=0` indicates APB cycle completed, {`prdata` and `pslverr`} valid.
5. `WrBack` to step 2 for more APB transactions.ite `TDR` with `start=0`.
6. Back to step 2 for more APB transactions.

4.4 E21 Address Map

Table 48 E21 Address Map

Start Address	End Address	Size	Description	Comment
0x0000_0000	0x0000_0fff	4K	Debug controller control	
0x0000_3000	0x0000_3fff	4K	Error-device	
0x0000_4000	0x0000_4fff	4K	Test status	
0x0170_0000	0x0170_0fff	4K	Bus error unit	
0x0200_0000	0x02ff_ffff	16M	Interrupt controller	
0x1000_0000	0x1000_0fff	4K	Trace encoder	Only in trial version
0x2000_0000	0x2000_0fff	4K	SPI0	APB0. Refer to 6.8, *SPI Control Register*
0x2000_1000	0x2000_1fff	4k	SPI1	APB0. Refer to 6.8, *SPI Control Register*
0x2000_2000	0x2000_2fff	4K	I2C	APB0. Refer to 6.9, *I2C Control Register*
0x2000_3000	0x2000_3fff	4K	LTMR	APB0. Refer to 6.10, *Local Timer Register*
0x2000_4000	0x2000_4fff	4K	UART	APB0. Refer to 6.11, *UART Control Register*
0x2000_5000	0x2000_5fff	4K	GPIO	APB0. Refer to 6.12, *GPIO Control Register*
0x2000_6000	0x2000_ffff	40K	Reserved	
0x2001_0000	0x2001_ffff		Reserved	Simulation only (sim_term_sel)
0x2010_0000	0x211f_ffff		Reserved	

0x2120_0000	0x21ff_ffff		QM	APBX32
				QM0 = 0x2120_0000 ~ 0x2120_ffff or
				0x2124_0000 ~ 0x21ff_ffff
				QM1 = 0x2121_0000 ~ 0x2121_ffff
				QM2 = 0x2122_0000 ~ 0x2122_ffff
				QM3 = 0x2123_0000 ~ 0x2123_ffff

0x2200_0000	0x2200_ffff	64k	FLC Slice0, IPM_MC0	APBX0
0x2201_0000	0x2201_ffff	64k	FLC Slice0, FLC1_0	APBX8
0x2202_0000	0x2202_ffff	64k	FLC Slice0, FLC2	APBX16
0x2203_0000	0x2203_ffff	64k	FLC Slice0, AXI_MASTER	APBX20
0x2204_0000	0x2204_ffff	64k	FLC Slice0, IPM_MC1	APBX1
0x2205_0000	0x2205_ffff	64k	FLC Slice0, FLC1_1	APBX9
0x2206_0000	0x2207_ffff	128k	Reserved	
0x2208_0000	0x2208_ffff	64k	IPM_MC broadcast	APBX33
0x2209_0000	0x2209_ffff	64k	FLC1 broadcast	APBX34
0x220a_0000	0x220a_ffff	64k	FLC2 broadcast	APBX35
0x220b_0000	0x220b_ffff	64k	AXI_Master broadcast	APBX36
0x220c_0000	0x220f_ffff	256k	Reserved	
0x2210_0000	0x221f_ffff	1M	FLC Slice0, IPM_PHY0	APBX24
0x2220_0000	0x222f_ffff	1M	FLC Slice0, IPM_PHY1	APBX25

Table 48 E21 Address Map (Continued)

0x2230_0000	0x22ef_ffff	12M	Reserved	
0x22f0_0000	0x22ff_ffff	1M	IPM_PHY broadcast	APBX37
0x2300_0000	0x2300_ffff	64k	FLC Slice1, IPM_MC0	APBX2
0x2301_0000	0x2301_ffff	64k	FLC Slice1, FLC1_0	APBX10
0x2302_0000	0x2302_ffff	64k	FLC Slice1, FLC2	APBX17
0x2303_0000	0x2303_ffff	64k	FLC Slice1, AXI_MASTER	APBX21
0x2304_0000	0x2304_ffff	64k	FLC Slice1, IPM_MC1	APBX3
0x2305_0000	0x2305_ffff	64k	FLC Slice1, FLC1_1	APBX11
0x2306_0000	0x230f_ffff	640k	Reserved	
0x2310_0000	0x231f_ffff	1M	FLC Slice1, IPM_PHY0	APBX26
0x2320_0000	0x232f_ffff	1M	FLC Slice1, IPM_PHY1	APBX27
0x2330_0000	0x23ff_ffff	13M	Reserved	
0x2400_0000	0x2400_ffff	64k	FLC Slice2, IPM_MC0	APBX4
0x2401_0000	0x2401_ffff	64k	FLC Slice2, FLC1_0	APBX12
0x2402_0000	0x2402_ffff	64k	FLC Slice2, FLC2	APBX18
0x2403_0000	0x2403_ffff	64k	FLC Slice2, AXI_MASTER	APBX22
0x2404_0000	0x2404_ffff	64k	FLC Slice2, IPM_MC1	APBX5
0x2405_0000	0x2405_ffff	64k	FLC Slice2, FLC1_1	APBX13
0x2406_0000	0x240f_ffff	640k	Reserved	
0x2410_0000	0x241f_ffff	1M	FLC Slice2, IPM_PHY0	APBX28
0x2420_0000	0x242f_ffff	1M	FLC Slice2, IPM_PHY1	APBX29
0x2430_0000	0x24ff_ffff	13M	Reserved	
0x2500_0000	0x2500_ffff	64k	FLC Slice3, IPM_MC0	APBX6
0x2501_0000	0x2501_ffff	64k	FLC Slice3, FLC1_0	APBX14
0x2502_0000	0x2502_ffff	64k	FLC Slice3, FLC2	APBX19
0x2503_0000	0x2503_ffff	64k	FLC Slice3, AXI_MASTER	APBX23
0x2504_0000	0x2504_ffff	64k	FLC Slice3, IPM_MC1	APBX7
0x2505_0000	0x2505_ffff	64k	FLC Slice3, FLC1_1	APBX15
0x2506_0000	0x250f_ffff	640k	Reserved	
0x2510_0000	0x251f_ffff	1M	FLC Slice3, IPM_PHY0	APBX30
0x2520_0000	0x252f_ffff	1M	FLC Slice3, IPM_PHY1	APBX31
0x2530_0000	0x25ff_ffff	13M	Reserved	
0x2600_0000	0x2600_0fff	4k	System Control register	Refer to 6.1, *System Control Registers (0X2600_0000 ~ 0X2600_0FFF)*
0x2600_1000	0x2600_1fff	4k	SF_CTRL	Refer to6.2, *Serial Flash Control Register* SRAM buffer offset: 0x300
0x2600_2000	0x2600_2fff	4k	L1C	Refer to 6.3, *L1C Control Register*
0x2600_3000	0x2600_3fff	4k	Reserved	
0x2600_4000	0x2600_4fff	4k	DMA	Refer to 6.4, *DMA Control Register*
0x2600_5000	0x2600_5fff	4k	SEC	Refer to 6.5, *Security Engine Control Register*

Table 48 E21 Address Map (Continued)

0x2600_6000	0x2600_ffff	40k	Reserved	
0x2601_0000	0x2601_0fff	4k	μC_PCIe	APBX47. Refer to 6.6, *μC-PCIe Register*
0x2601_1000	0x2607_ffff		Reserved	
0x2608_0000	0x2608_ffff	64K	NEO_EFUSE	APBX48. Refer to 6.7, *OTP Register*
0x2609_0000	0x26ff_ffff		Reserved	
0x2700_0000	0x27ff_ffff	16M	D2D_PHY	APBX38
0x2800_0000	0x287f_ffff	8M	PCIe_PHY (PMA)	APBX39. PMA
0x2880_0000	0x2880_ffff		Reserved	
0x2881_0000	0x28af_ffff	8M	PCIe_PHY (PCS)	APBX39. PCS
0x28b0_0000	0x28ff_ffff		Reserved	
0x2900_0000	0x29ff_ffff	16M	PCIe MAC	APBX40. CXL_PCIe_EP, CXL_PCIe_RC config registers.
0x2a00_0000	0x2a00_ffff	64K	I3C Master	Refer to 6.14, *I3C Register*
0x2a01_0000	0x2a01_ffff	64K	I3C Slave	Refer to 6.14, *I3C Register*
0x2a02_0000	0x2a02_ffff	64k	AXI4TG	APBX42. Refer to 6.15, *AXI4TG Register*

0x2a03_0000	0x2a03_ffff	64K	ETM	APBX43
				ETM0 = 0x2a030000 ~ 0x2a030fff
				ETM1 = 0x2a031000 ~ 0x2a031fff
				ETM2 = 0x2a032000 ~ 0x2a032fff
				ETM3 = 0x2a033000 ~ 0x2a033fff

0x2a04_0000	0x2a04_ffff	64K	Reserved	APBX44
0x2a05_0000	0x2a0f_ffff		Reserved	
0x2a10_0000	0x2a1f_ffff	1M	Reserved	APBX45
0x2a20_0000	0x2aff_ffff		Reserved	APBx
0x2b00_0000	0x2b00_ffff	64K	EMAC	AHBX0. Refer to 6.13, *EMAC Register*
0x2b01_0000	0x2bff_ffff		Reserved	AHBx
0x2c00_0000	0x2c0f_ffff	1M	D2D_PHY	APBX46. Alternate window for D2D PHY, GFH0 D2D = 0x2c00_0000 ~ 0x2c00_ffff GFH1 D2D = 0x2c01_0000 ~ 0x2c01_ffff GFH2 D2D = 0x2c02_0000 ~ 0x2c02_ffff GFH3 D2D = 0x2c03_0000 ~ 0x2c03_ffff MKB0 D2D = 0x2c04_0000 ~ 0x2c04_ffff MKB1 D2D = 0x2c05_0000 ~ 0x2c05_ffff GFH0 D2D = 0x2c06_0000 ~ 0x2c0e_ffff Broadcast = 0x2c0f_0000 ~ 0x2c0f_ffff
0x2c10_0000	0x2dff_ffff		Reserved	
0x2e00_0000	0x2e00_ffff		ETU	APBX49
0x2e01_0000	0x2eff_ffff		Reserved	
0x2f00_0000	0x2fxx_xxxx	16M	D2D_IO	APBX41. D2D A5L access window for remote chiplet CSR

Table 48 E21 Address Map (Continued)

0x3000_0000	0x300f_ffff	1M	Boot ROM	Boot ROM: 32-256KB
0x3010_0000	0x3fff_ffff	255M	XIP NOR Flash	SPI NOR Flash (XIP access)
0x4000_0000	0x40ff_ffff	16M	SRAM	Chiplet SRAM
0x4100_0000	0x4fff_ffff	240M	Reserved	
0x5000_0000	0x50ff_ffff	24M	RC0_x8_DBL	CXL_PCIe RC0_x8 door bell
0x5100_0000	0x51ff_ffff	24M	RC0_x8_AXIS	CXL_PCIe RC0_x8 AXI slave port
0x5200_0000	0x52ff_ffff	24M	EP0_AXIS	CXL_PCIe slice 0 AXI slave port
0x5300_0000	0x53ff_ffff	24M	EP1_AXIS	CXL_PCIe slice 1 AXI slave port
0x5400_0000	0x54ff_ffff	24M	EP2_AXIS	CXL_PCIe slice 2 AXI slave port
0x5500_0000	0x55ff_ffff	24M	EP3_AXIS	CXL_PCIe slice 3 AXI slave port
0x5600_0000	0x56ff_ffff	24M	Reserved	
0x5700_0000	0x57ff_ffff	24M	Reserved	
0x5800_0000	0x58ff_ffff	24M	RC0_x4_DBL	CXL_PCIe RC0_x4 door bell
0x5900_0000	0x59ff_ffff	24M	RC0_x4_AXIS	CXL_PCIe RC0_x4 AXI slave port
0x5a00_0000	0x5aff_ffff	24M	RC1_x8_DBL	CXL_PCIe RC1_x8 door bell
0x5b00_0000	0x5bff_ffff	24M	RC1_x8_AXIS	CXL_PCIe RC1_x8 AXI slave port
0x5c00_0000	0x5cff_ffff	24M	RC1_x4_DBL	CXL_PCIe RC1_x4 door bell
0x5d00_0000	0x5dff_ffff	24M	RC1_x4_AXIS	CXL_PCIe RC1_x4 AXI slave port
0x5e00_0000	0x5eff_ffff		Reserved	
0x5f00_0000	0x5fff_ffff	24M	DBI	CXL_PCIe DBI interface selected by “pcie_phy_sel”.
0x6000_0000	0x6fff_ffff	256M	µc_nic_remap	Default access to DDR45, controlled by register “µc_nic_remap” (0x2600007c)
0x7000_0000	0x7fff_ffff	256M	Reserved	
0x8000_0000 ~ 0xffff_ffff: E21 core targets				
0x8000_0000	0x8001_ffff	128K	TIM0	
0x8002_0000	0x8003_ffff	128K	TIM1	

5.0 E21 IRQ Map

Table 49 shows the IRQ map within the micro controller.

Table 49 E21 IRQ Map

IRQ	Device	Comment
0	UART	
1	SPI0	
2	SPI1	
3	I2C	I2C controller
4	Reserved	Reserved for Local timer 1
5	TMR2	Local timer 2
6	TMR3	Local timer 3
7	GPIO	
8	DMA	
9	I3C Master	I3C Master
10	I3C Slave	I3C Slave
11	ELBI Access	
12	PCIe MAC	
13	RC_msi_ctrl_int	
14	EP0_PERST	
15	EP1_PERST	
16	EP2_PERST	
17	SEC-SHA	
18	SEC-AES	
19	SEC-TRNG	
20	SF_CTRL	XIP/SPI controller
21	OTP	
22	L1C	
23	BMX_Timeout	
24	BMX_Error	
25	DMA_interr	
26	DMA_inttc	
27	EP3_PERST	
28	Reserved	
29	Reserved	
30	ETM IRQ	ETM
31	RISCV_IRQ	From RISCV GPIO0
32	FLC1	FLC1
33	FLC2	FLC2
34	IPM MC	IPM MC

Table 49 E21 IRQ Map (Continued)

35	IPM PHY	IPM PHY
36	Reserved	
37	Reserved	
38	FLC_AXIM	FLC AXI Master
39	QM	Queue Manager
40	D2D_0_0	DDR chiplet #1, slice#0
41	D2D_1_0	DDR chiplet #2, slice#0
42	D2D_2_0	DDR chiplet #3, slice#0
43	D2D_3_0	DDR chiplet #4, slice#0
44	MKB_D2D0	Inter MKB D2D #1
45	MKB_D2D1	Inter MKB D2D #2
46	Reserved	
47	Reserved	
48	Radm_pm_turnoff	
49	~mac_mstr_aw_fifo_empty	
50	~m2sreq_memInv_NT_fifo_empty	
51	~m2sreq_specrd_fifo_empty	
52	m2sreq_memInv_NT_rsp_fifo_full	
53	Hot_reset_req	RCx8, RCx4, EP[3:0]
54	RC_radm_pme_or_err	RCx8 or RCx4
55	EMAC	
56	~Vdm_empty	Ep[1:4] vdm_0 or vdm_1 fifo not empty
57	TS_IRQ	Temperature Sensor
58	VS_IRQ	Voltage Sensor
59	Reserved	
60-55	Reserved	
61	Reserved	
62	Watchdog Timer	Local timer WDT
127-63	Reserved	
NMI	Watchdog Timer	Local timer WDT

5.1 Inter-chiplet Notification

Table 50 list the companion chiplets. Companion chiplets –> Mockingbird, E21 IRQ

Table 50 Companion Chiplets

IRQ	Device	Comment
40	D2D_0_0	DDR chiplet #1, slice 0
41	D2D_1_0	DDR chiplet #2, slice 0
42	D2D_2_0	DDR chiplet #3, slice 0

Table 50 Companion Chiplets (Continued)

IRQ	Device	Comment
43	D2D_3_0	DDR chiplet #4, slice 0
44	MKB_D2D0	Inter MKB D2D #1
45	MKB_D2D1	Inter MKB D2D #2

- Register: D2D_IRQ, Address: 0x26000060

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:10	d2d_irq_i[9:0]	10'b0	R	D2D Chiplet IRQ input status. Corresponding to E21 IRQ[47:40, 37:36] [9:6]: Reserved [5:4]: MKB D2D[1:0] [3:0]: goldfinch[0:3]
9:0	d2d_irq_o[9:0]	10'b0	R/W	D2D Chiplet IRQ output. [9:6]: Reserved [5:4]: MKB D2D[1:0] [3:0]: goldfinch[0:3]

- Virtual Wires: D2D port `NMI_in` and `NMI_out` are used, e.g. Mockingbird –> Goldfinch:
- (Mockingbird) E21 CSR `(d2d_irq) ->`

  `NMI_in (u_d2d_top_wrapper) ->`

  `… D2D … ->`

  `NMI_out (u_d2d_top_wrapper) ->`

  E21 IRQ/CSR (Goldfinch)
- Goldfinch –>

  Mockingbird: (Goldfinch) E21 CSR –>

  `NMI_in (u_d2d_top_wrapper) ->`

  `… D2D … ->`

  `NMI_out (u_d2d_top_wrapper) ->`

  E21 IRQ/CSR (Mockingbird)

6.0 Registers

This chapter summarizes all the registers for each functional module present in the FLC Memory Expander. CSR registers are µC controlled and selected by µC address map. See Table 48, *E21 Address Map* for complete address space assignment.

6.1 System Control Registers (0X2600_0000 ~ 0X2600_0FFF)

6.1.1 GPIO_POS_REG, 0x26000000

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	gpio_pos_reg	N/A	R	Sample value of GPIO [19:0] at chiplet reset.

6.1.2 PMU_CTRL, 0x26000004

Bits	Bit Name	Default	Type	Comment
31:28	FLC_SLICE_RST[3:0]	4'b0	R/W	1 = FLC pcie slice [3:0] hardware reset. Note: IPM_PHY and CSR do not reset by this bit.
27	Reserved	1'b0	R/W	Reserved
26	sel_riscv_uart	1'b0	R/W	RISCV's UART1 map to GPIO[1:0]
25	config_spi_en	1'b0	R/W	config_spi map to gpio [16:14]
24	spi1_en	1'b0	R/W	SPI1 map to gpio [13:10] 0 = SPI1 in reset state 1 = SPI1 enabled
23	spi0_en	1'b0	R/W	SPI0 map to gpio [7:4] (function 2) 0 = SPI0 in reset state 1 = SPI0 enabled
22	I2c_en	1'b0	R/W	I2C map to gpio [9:8] (function 3) 0 = I2C in reset state 1 = I2C enabled
21	force_clk_flc2_en	1'b0	R/W	1 = disable flc2 clock gated
20	force_clk_flc1_en	1'b0	R/W	1 = disable flc1 clock gated
19	force_clk_ipm_en	1'b0	R/W	1 = disable ipm clock gated
18	Reserved	1'b0	R/W	Reserved
17:16	mkb_d2d_resetn[1:0]	2'b0	R/W	0 = MKB_D2D interface in reset state 1 = MKB_D2D interface is enabled [17]: MKB_D2D #1 [16]: MKB_D2D #0

Bits	Bit Name	Default	Type	Comment
15:12	sf_io_pullup_en	4’b0	R/W	[15]: sf_io [3] pull up_en [14]: sf_io [2] pull up_en [13]: sf_io [1] pull up_en [12]: sf_io [0] pull up_en
11:8	d2d_ddr_resetn[3:0]	4’b0	R/W	0 = D2D_DDR (goldfinch D2D interface) in reset state 1 = D2D_DDR interface is enabled [11]: goldfinch #3 [10]: goldfinch #2 [9]: goldfinch #1 [8]: goldfinch #0
7	boot_tim	1’b0	R/W	Used together with bit 5 of this register (tim0_sel) and boot_xip pin, write this bit will automatically generate an internal reset pulse to reset the E21 core and select e21_reset_vec 0 and boot_xip = 0: e21_reset_vec = 32'h3000_0000 (ROM) 0 and boot_xip = 1: e21_reset_vec = 32'h3010_0060 (XIP NOR) 1 and tim0_sel = 0: e21_reset_vec = 32'h8002_0000 (TIM1) 1 and tim0_sel = 1: e21_reset_vec = 32'h8000_0000 (TIM0)
6	flc1_axi_master_sync_mode	1’b1	R/W	1 = flc1’s axi master interface runs in synchronous mode
5	tim0_sel	1’b0	R/W	See bit 7 of this register (boot_tim)
4	flc1_axi_slave_sync_mode	1’b1	R/W	1 = flc1's axi slave interface runs in synchronous mode
3	ipm_mc_axi_sync_mode	1’b0	R/W	1 = ipm_mc's axi interface runs in synchronous mode
2	flc2_slave_p0_sync_mode	1’b0	R/W	1 = flc2 slave port 0 run in sync mode
1	flc2_slave_p1_sync_mode	1’b1	R/W	1 = flc2 slave port 1 run in sync mode
0	µC_hclk_halt_en	1’b0	R/W	1 = stop µC hclk when WFI

6.1.3 SYS_CTRL, 0x26000008

Bits	Bit Name	Default	Type	Comment
31:28	elbi_acc_flag [3:0]	4'b0	R/W1c	Set to “1” when ELBI accessed. Write 1 to clear
27	Reserved	1'b1	R/W	Reserved
26:24	elbi_sram_base	3'b111	R/W	ELBI access base address in SRAM with 8KB allocation.
23:15	noc_cfg	9'b010011110	R/W	NoC configuration: [23:20] = MemIntlvWays, default 4'b0100 [19] = GFH3En, default 1'b1 [18] = GFH2En, default 1'b1 [17] = GFH1En, default 1'b1 [16] = GFH0En, default 1'b1 [15] = DualMemChEn, default 1'b0
14	spi1_master_mode	1'b0	R/W	1 = set SPI1 IO pads as master mode
13:4	AHB_NS_ACCESS_DIS	10'b0	R/W	1 = Disable AHB0 device access in non-secure mode. [13] = XIP [12] = Boot ROM [11] = Security engine [10] = DMA [9] = SYSCTRL [8] = AHB1 (flc1/2, axim, D2D, AXI4TG, I3C…) [7] = OTP [6] = L1C [5] = SFCTRL [4] = AHB0 (SPI, I2C, LTMR, UART, GPIO…)
3	force_dma_active	1'b0	R/W	1 = force Core Debug Module active
2	simulation_speed_up	1'b0	R/W	1 = speed up 32KHz clock for simulation
1	wdogirqnmi	1'b0	R/W	WDOG Interrupt CPU NMI Enable
0	lockupreset	1'b0	R/W	CPU Lockup Reset

6.1.4 RESET_INFO, 0x26000010

Bits	Bit Name	Default	Type	Comment
31	riscv_etm_enable	1'b0	R/W	RISCV AXI (target) read only through NoC 1=Enable riscv_etm mapping
30:16	sysctrl_riscv_etm[30:16]	15'b0	R/W	RISCV-ETM AXI access address [30:16]
15:5	Reserved	11'b0	R/W	Reserved
4	Reserved	1'b0	R	Reserved, always read ‘b0
3	boot_xip	N/A	R	Returns pin value of “boot_xip”
2	LOCKUPRESET	1'b0	R/W1c	Write 1 to clear
1	WDOGRESETREQ	1'b0	R/W1c	Write 1 to clear
0	SYSRESETREQ	1'b0	R/W1c	Write 1 to clear

6.1.5 SYSCTRL_RISCV, 0x26000014

Bits	Bit Name	Default	Type	Comment
31:14	h0_ rstvec_l[19:2]	18'b0	R/W	RISCV core0 low reset vector A[19:2]
13	h1_riscv_rst_vec_en	1'b0	R/W	RISCV core1 reset vector enable
12	h0_riscv_rst_vec_en	1'b0	R/W	RISCV core0 reset vector enable

Bits	Bit Name	Default	Type	Comment
11:4	Reserved	8’b0	R/W	Reserved
3	riscv_X_om	1’b0	R/W	Operational mode
2	riscv_wakeup	1’b0	R/W	RISCV wakeup
1	riscv_irq	1’b0	R/W	RISCV irq[31]
0	riscv_resetn	1’b0	R/W	0 = reset

6.1.6 SYSCTRL_RISCV_RSTVEC, 0x26000018

Bits	Bit Name	Default	Type	Comment
31:12	riscv_rstvec_h	20’b0	R/W	RISCV high reset vector A[51:32] (both cores)
11:0	h0_rstvec_l[31:20]	12’b0	R/W	RISCV core0 low reset vector A[31:20]

6.1.7 SYSCTRL_RISCV_RSTVEC1, 0x2600001C

Bits	Bit Name	Default	Type	Comment
31:2	h1_ rstvec_l[31:2]	30’b0	R/W	RISCV core1 low reset vector A[31:2]
1:0	Reserved	2’b0	R/W	Reserved

6.1.8 BMX_CONFIG, 0x26000020

Bits	Bit Name	Default	Type	Comment
31:27	Reserved	5’b0	R	Reserved, always read ‘b0
26	bmx_timeout_lat	1’b0	R/Wc	BMX time out latch/interrupt, write to clear
25	bmx_err_dec	1’b0	R	BMX error
24	bmx_err_addr_dis	1’b0	R/W	Disable BMX address error
23:20	bmx_dbg_sel	4’b0	R/W	BMX debug mux select
19	bmx_busy_option_dis	1’b0	R/W	Ignore HTRANS busy
18	bmx_err_dis	1’b0	R/W	Disable error response

17:14	bmx_slave_timeout_en	4’b0	R/W	Bus timeout enable, 1-bit per slave
				1XXX = S3
				X1XX = S2
				XX1X = S1
				XXX1 = S0

13:10	bmx_arb_mode	4’b0	R/W	Arbitration priority

Bits	Bit Name	Default	Type	Comment

9:0	bmx_master_hsel	10’b0	R/W	Ignore waiting for slave's HREADYOUT for better timing, 1-bit per master
				1XXXXXXXXX = M9
				X1XXXXXXXX = M8
				XX1XXXXXXX = M7
				XXX1XXXXXX = M6
				XXXX1XXXXX = M5
				XXXXX1XXXX = M4
				XXXXXX1XXX = M3
				XXXXXXX1XX = M2
				XXXXXXXX1X = M1
				XXXXXXXXX1 = M0

6.1.9 BMX_ERR_ADDR, 0x26000024

Bits	Bit Name	Default	Type	Comment
31:0	bmx_err_addr	32’h0	R	Access error address

6.1.10 DBL_RC0_X4_REMAP, 0x26000028

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12’b0	R	Reserved, always read ‘b0
19:0	dbl_rc0_x4_a_43_24	20’b0	R/W	dbl_rc0_x4 access window [0x58xx_xxxx] upper address bit

6.1.11 µC_RC0_X4_REMAP, 0x2600002C

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12’b0	R	Reserved, always read ‘b0
19:0	µC_rc0_x4_a_43_24	20’b0	R/W	µC_rc0_x4 access window [0x59xx_xxxx] upper address bit

6.1.12 DBL_RC1_X4_REMAP, 0x26000030

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12’b0	R	Reserved, always read ‘b0
19:0	dbl_rc1_x4_a_43_24	20’b0	R/W	dbl_rc1_x4 access window [0x5Cxx_xxxx] upper address bit

6.1.13 µC_RC1_X4_X8_REMAP, 0x26000034

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:20	µC_rc1_x8_a_43_36	0	R/W	µC_rc1_x8 access window [0x5Bxx_xxxx] upper address bit. A[43:36]

Bits	Bit Name	Default	Type	Comment
19:0	µC_rc1_x4_a_43_24	20'b0	R/W	µC_rc1_x4 access window [0x5Dxx_xxxx] upper address bit

6.1.14 INTERFACE_SEL, 0x26000038

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:12	d2d_io_a31x24	8'h2A	R/W	D2D APB (A5L) access upper address bits: A[31:24]. Default points to apbx (0x2Axxxxxx)
11:8	d2d_phy_sel	4'h0	R/W	Select D2D for configuration. 0~3: 1st 1x16 D2D_DDR[0:3] 4: MKB_D2D0 5: MKB_D2D1 6~14: Reserved (default D2D_DDR0) 15: Broadcast all, this field combines with address windows 0x27xx_xxxx for D2D decoding
7:4	ipm_phy_sel	4'h0	R/W	Select IPM PHY for configuration. (register: 0x2600_0064) 0: FLC slice 0, ipm0 1: FLC slice 0, ipm1 2: FLC slice 1, ipm0 3: FLC slice 1, ipm1 4: FLC slice 2, ipm0 5: FLC slice 2, ipm1 6: FLC slice 3, ipm0 7: FLC slice 3, ipm1 8: Reserved for ipm9 9~14: Reserved 15: Broadcast all
3:0	pcie_phy_sel	4'h0	R/W	Select CXL/PCIe interface for configuration. 0: RCx8[0] MAC [NVMe host/CXL RC] 1~4: host side slice pcie_ep[0:3] 5: RCx4[0] MAC 6: RC1_x8[1] MAC 7: RC1_x4[1] MAC 8~13: Reserved 14: Broadcast (RCx8[0:1]) 15: Broadcast all

6.1.15 SE_AES_KEY, 0x26000040~0x2600005C

Bits	Bit Name	Default	Type	Comment
255:0	se_aes_key	256'h0	R/W	Security engine key

6.1.16 D2D_IRQ, 0x26000060

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:10	d2d_irq_i[9:0]	10'b0	R	D2D Chiplet IRQ input status. Corresponding to E21 IRQ[47:40, 37:36] [9:6]: Reserved [5:4]: MKB D2D[1:0] [3:0]: goldfinch[0:3]
9:0	d2d_irq_o[9:0]	10'b0	R/W	D2D Chiplet IRQ output. [9:6]: Reserved [5:4]: MKB D2D[1:0] [3:0]: goldfinch[0:3]

6.1.17 IPM_PHYCONFIG, 0x26000064

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7	Ipm_mcu_scratchpad	1'b0	R/W	1: mcu_addr_offset=0x3000
6	Powerdn	1'b0	R/W	1: power down PHY
5	Boot_i	1'b1	R/W	"boot_i" pin of PHY
4	ipm_mcu_addr_offset	1'b0	R/W	0: mcu_addr_offset=0x2000 1: mcu_addr_offset=0x1000
3	ipm_mcu_apb_en	1'b0	R/W	
2	ipm_mcu_mem_load_en	1'b0	R/W	
1	ipm_mcu_uncore_rstn	1'b1	R/W	
0	ipm_mcu_core_rstn	1'b0	R/W	[FLC slice, IPM PHY] selected by ipm_phy_sel (0x26000038[7:4])

6.1.18 DBL_RC0_X8_REMAP, 0x26000068

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	dbl_rc0_x8_a_43_24	20'b0	R/W	dbl_rc0_x8 access window [0x50xx_xxxx] upper address bit

6.1.19 µC_RC0_X8_REMAP, 0x2600006C

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	µC_rc0_x8_a_43_24	20'b0	R/W	µC_rc0_x8 access window [0x51xx_xxxx] upper address bit

6.1.20 µC_EP01_REMAP, 0x26000070

Bits	Bit Name	Default	Type	Comment
31:16	µC_ep1_a_39_24	16'b0	R/W	µC_pcie access window [0x53xxxxxx] upper address bit
15:0	µC_ep0_a_39_24	16'b0	R/W	µC_pcie access window [0x52xxxxxx] upper address bit

6.1.21 µC_EP23_REMAP, 0x26000074

Bits	Bit Name	Default	Type	Comment
31:16	µC_ep3_a_39_24	16'b0	R/W	µC_pcie access window [0x55xxxxxx] upper address bit
15:0	µC_ep2_a_39_24	16'b0	R/W	µC_pcie access window [0x54xxxxxx] upper address bit

6.1.22 DBL_RC1_X8_REMAP, 0x26000078

Bits	Bit Name	Default	Type	Comment
31:20	µC_rc1_x8_a_35_24	12'b0	R/W	µC_rc1_x8 access window [0x5Bxx_xxxx] upper address bit. A[35:24]
19:0	dbl_rc1_x8_a_43_24	20'b0	R/W	dbl_rc1_x8 access window [0x5Axx_xxxx] upper address bit

6.1.23 µC_NIC_REMAP, 0x2600007C

Bits	Bit Name	Default	Type	Comment
31:28	µC_nic_addr_51x48	4'b0	R/W	A[51:48] of µC to NoC access
27:4	µC_nic_addr_47x24	24'h500000	R/W	A[47:24] of µC to NoC access
3:1	Reserved	3'b0	R	Reserved, always read 'b0
0	valid	1'b0	R/W	1: remap is valid

6.1.24 FLC_ADDRESS_REMAP, 0x26000084 - 0x260000F8

- Address: 0x26000084~80 (Slice0, Mapper0)
- Address: 0x2600008C~88 (Slice0, Mapper1)
- Address: 0x26000094~90 (Slice0, Mapper2)
- Address: 0x2600009C~98 (Slice0, Mapper3)
- Address: 0x260000A4~A0 (Slice1, Mapper0)
- Address: 0x260000AC~A8 (Slice1, Mapper1)
- Address: 0x260000B4~B0 (Slice1, Mapper2)
- Address: 0x260000BC~B8 (Slice1, Mapper3)
- Address: 0x260000C4~C0 (Slice2, Mapper0)
- Address: 0x260000CC~C8 (Slice2, Mapper1)
- Address: 0x260000D4~D0 (Slice2, Mapper2)
- Address: 0x260000DC~D8 (Slice2, Mapper3)

- Address: 0x260000E4~E0 (Slice3, Mapper0)
- Address: 0x260000EC~E8 (Slice3, Mapper1)
- Address: 0x260000F4~F0 (Slice3, Mapper2)
- Address: 0x260000FC~F8 (Slice3, Mapper3)

Bits	Bit Name	Default	Type	Comment
63:40	New_base_address	24'b0	R/W	A[47:24] after remap
39:36	Reserved	4'b0	R/W	Reserved
35:32	New_base_address_h	4'b0	R/W	A[51:48] after remap
31:8	Base_address	24'b0	R/W	A[47:24]. Remap block size: 16MB
7:6	Reserved	2'b0	R/W	Reserved
5:1	length	5'b0	R/W	0=16M, 1=32M, 2=64M…16=1T
0	valid	1'b0	R/W	valid

6.1.25 MKB_PLL_CTRL1, 0x26000100

Bits	Bit Name	Default	Type	Comment
31	DESKEW_PLL_LOCKED	1'b0	R	Deskew calibration settled
30	PLL_LOCKED	1'b0	R	PLL locked
29	frefcmlen	1'b0	R/W	Enable free CML input
28	foutvcoen	1'b0	R/W	VCO rate output clock enable
27:23	foutvcobyp	5'b0	R/W	Bypass undivided vco clock to specific output,
22	foutdiffen	1'b0	R/W	Enable [FOUTDIFFP/N] (frequency is FVCO/POSTDIV4)
21	foutcmlen	1'b0	R/W	Enable FOUTCML[P/N]
20	dmsen	1'b0	R/W	Enable delta-sigma modulator. 0 = DSM is powered down (integer mode). 1 = DSM is powered up (fractional mode)
19	dskewfastcal	1'b0	R/W	Deskew fast calibration enable.
18:7	dskewcalin	12'b0	R/W	Dskewcalbyp=0: initial value for deskew calibration. Dskewcalbyp=1: override value of deskew calibration
6	dskewcalen	1'b0	R/W	Deskew calibration enable
5:3	dskewcalcnt	3'b010	R/W	Counter for deskew calibration loop
2	dskewcalbyp	1'b0	R/W	Deskew calibration bypass
1	dac_en	1'b0	R/W	Enabling fractional noise canceling DAC
0	pll_en	1'b0	R/W	PLL enable

6.1.26 MKB_PLL_CTRL2, 0x26000104

Bits	Bit Name	Default	Type	Comment
31:28	posdiv2	4'b0111	R/W	PLL post divider 2 2GHz/10 = 200 MHz
27:24	posdiv1	4'b0001	R/W	PLL post divider 0 2GHz/2 = 1GHz

Bits	Bit Name	Default	Type	Comment
23:20	posdiv0	4’b0001	R/W	PLL post divider 0 2GHz/2 = 1GHz
19:18	posdiv4	2’b0	R/W	2G/4 -> fout CML (not used)
17:6	fbdiv	12’d80	R/W	Feedback clock divider 25 MHz*80 = 2 GHz VCO
5:0	refdiv	6’d1	R/W	Reference clock divider

6.1.27 MKB_PLL_CTRL3, 0x26000108

Bits	Bit Name	Default	Type	Comment
31:28	fouten	4’b1111	R/W	Bit-wise post divide enable, fout[0,1,2,3]
27:24	posdiv3	4’b0	R/W	PLL post divider 3, bypass (25MHz)
23:0	frac	24’h0	R/W	Fractional value of feed- back divider

6.1.28 AXI4TG_CTRL, 0x26000418

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14’b0	R/W	Reserved
17	axi4tg_err	1’b0	R	1 = Error detected
16	axi4tg_irq	1’b0	R	1 = Traffic generation completion
15:7	Reserved	9’b0	R/W	Reserved
6:2	axi4tg_sel	5’b	R/W	DUT selection 0={slice0, ipm0}, 1={slcie0, ipm1}, 2={slice1, ipm0}, 3={slice1, ipm1}, 4={slice2, ipm0}, 5={slcie2, ipm1}, 6={slice3, ipm0}, 7={slice3, ipm1}, 8={Goldfinch, d2d0, slice0}, 9={Goldfinch, d2d1, slice0}, 10={Goldfinch, d2d2, slice0}, 11={Goldfinch, d2d3, slice0}. 12=MKB D2D0, 13=MKB D2D1.
1	axi4tg_start	1’b0	R/W	Start generating or accepting the traffic
0	axi4tg_resetn	1’b0	R/W	Active-Low reset

6.1.29 I3C_DEBUG_PORT0, 0x2600041C

Bits	Bit Name	Default	Type	Comment
31:0	i3cm_debug_port [31:0]	32’h8000555A	R	I3C master debug ports[31:0]

6.1.30 I3C_DEBUG_PORT1, 0x26000420

Bits	Bit Name	Default	Type	Comment
31	wakeup	1’b0	R	I3C Slave rxc wakeup
30	i2c_glitch_filter_en	1’b1	R	I3C Slave I2C glitch filter enable

Bits	Bit Name	Default	Type	Comment
29:23	Reserved	7’b0	R	Reserved, always read ‘b0
22:0	i3cm_debug_port [54:32]	23’h0	R	I3C master debug ports[54:32]

6.1.31 I3CS_CTRL1, 0x26000424

Bits	Bit Name	Default	Type	Comment
31	Reserved	1’b0	R/W	Reserved
30:23	i3cs_slv_dcr[7:0]	8’b0	R/W	Device Characteristic Register value
22:20	i3cs_slv_max_wr_speed[2:0]	3’b0	R/W	Slave maximum write data rate
19:17	i3cs_slv_max_rd_speed[2:0]	3’b0	R/W	Slave maximum read data rate
16:10	i3cs_static_addr[6:0]	7’b0	R/W	Slave static address
9	i3cs_static_addr_en	1’b0	R/W	Slave static address valid
8:5	i3cs_pending_int[3:0]	4’b0	R/W	Pending interrupt information
4:3	i3cs_act_mode[1:0]	2’b0	R/W	Slave activity mode
2	i3cs_slv_test_mode	1’b0	R/W	Slave test mode
1	i3cs_mode_i2c	1’b0	R/W	I2C or I3C mode select signal
0	legacy_i2c_xfer	1’b0	R	Legacy I2C transfer enable

6.1.32 I3CS_CTRL2, 0x26000428

Bits	Bit Name	Default	Type	Comment
31:0	i3c_slave_slv_pid [31:0]	32’h0	R/W	I3C slave PID[31:0]

6.1.33 I3CS_CTRL3,0x2600042C

Bits	Bit Name	Default	Type	Comment
31:23	Reserved	9’b0	R/W	Reserved
22:20	i3cs_slv_clk_data_turn_time	3’b0	R/W	Slave maximum clock data turnaround time
19:16	Reserved	4’b0	R/W	Reserved
15:0	i3c_slave_slv_pid [47:32]	16’h0	R/W	I3C slave PID[47:32]

6.1.34 I3CS_DEBUG_PORT0, 0x2600068C

Bits	Bit Name	Default	Type	Comment
31:0	i3cs_debug_port [31:0]	32’h5140	R	I3C slave debug ports[31:0]

6.1.35 I3CS_DEBUG_PORT1, 0x26000690

Bits	Bit Name	Default	Type	Comment
31:23	Reserved	N/A	N/A	Reserved

Bits	Bit Name	Default	Type	Comment
22:0	i3cs_debug_port [54:32]	23'b0	R	I3C slave debug ports[54:32]

6.1.36 D2D_DDR[3:0]_PLL_CTRL1, 0x26000700 - 0x26000748

- Address: 0x26000700 (D2D_DDR0)
- Address: 0x26000730 (D2D_DDR1)
- Address: 0x2600073C (D2D_DDR2)
- Address: 0x26000748 (D2D_DDR3)

6.1.37 MKB_D2D_PLL_CTRL1, shared control for MKB_D2D[1:0], 0x26000754

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	2'b0	R/W	Reserved
29	frefcmlen	1'b0	R/W	Enable free CML input
28	foutvcoen	1'b1	R/W	VCO rate output clock enable, 8GHz drives BOW PHY
27:23	foutvcobyp	5'b0	R/W	Bypass undivided vco clock to specific output,
22	foutdiffen	1'b0	R/W	Enable [FOUTDIFFP/N] (frequency is FVCO/POSTDIV4)
21	foutcmlen	1'b0	R/W	Enable FOUTCML[P/N]
20	dmsen	1'b0	R/W	Enable delta-sigma modulator. 0 = DSM is powered down (integer mode). 1 = DSM is powered up (fractional mode)
19	dskewfastcal	1'b0	R/W	Deskew fast calibration enable.
18:7	dskewcalin	12'b0	R/W	Dskewcalbyp=0: initial value for deskew calibration. Dskewcalbyp=1: override value of deskew calibration
6	dskewcalen	1'b0	R/W	Deskew calibration enable
5:3	dskewcalcnt	3'b010	R/W	Counter for deskew calibration loop
2	dskewcalbyp	1'b0	R/W	Deskew calibration bypass
1	dac_en	1'b0	R/W	Enabling fractional noise canceling DAC
0	pll_en	1'b1	R/W	PLL enable

6.1.38 D2D_DDR[3:0]_PLL_CTRL2, 0x26000704 ~ 0x2600074C

- Address: 0x26000704 (D2D_DDR0)
- Address: 0x26000734 (D2D_DDR1)
- Address: 0x26000740 (D2D_DDR2)
- Address: 0x2600074C (D2D_DDR3)

6.1.39 MKB_D2D_PLL_CTRL2 shared control for MKB_D2D[1:0], 0x26000758

Bits	Bit Name	Default	Type	Comment
31:28	posdiv2	4'b0	R/W	PLL post divider 2 (not used)

Bits	Bit Name	Default	Type	Comment
27:24	posdiv1	4’b0	R/W	PLL post divider 1 (not used)
23:20	posdiv0	4’b0	R/W	PLL post divider 0 (not used)
19:18	posdiv4	2’b0	R/W	PLL post divider 4 8GHz/4 (not used)
17:6	fbdiv	12’h50	R/W	Feedback clock divider 100MHz*80 = 8GHz VCO
5:0	refdiv	6’h1	R/W	Reference clock divider

6.1.40 D2D_DDR[3:0]_PLL_CTRL3, 0x26000708 ~ 0x26000750

- Address: 0x26000708 (D2D_DDR0)
- Address: 0x26000738 (D2D_DDR1)
- Address: 0x26000744 (D2D_DDR2)
- Address: 0x26000750 (D2D_DDR3)

6.1.41 MKB_D2D_PLL_CTRL3 shared control for MKB_D2D[1:0], 0x2600075C

Bits	Bit Name	Default	Type	Comment
31:27	Reserved	5’b0	R/W	Reserved
26	LL_EXT_LPBK	0’b0	R/W	For loopback test
25	PHY_EXT_LPBK	0’b0	R/W	For loopback test
24	BC_PHY_SYSRS	0’b0	R/W	For loopback test
23:0	frac	24’h0	R/W	Fractional value of feed- back divider

6.1.42 D2D_DDR0_PLL_STATUS, 0x26000710

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18’b0	R	Reserved, always read ‘b0
13:2	d2d_dskewcallout_gfh0	12’h0	R	GFH0 D2D dskewcallout
1	d2d_deskew_pll_locked_gfh0	1’b0	R	GFH0 D2D Deskew calibration settled
0	d2d_pll_locked_gfh0	1’b0	R	GFH0 D2D PLL locked

6.1.43 D2D_DDR1_PLL_STATUS, 0x26000714

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18’b0	R	Reserved, always read ‘b0
13:2	d2d_dskewcallout_gfh1	12’h0	R	GFH1 D2D dskewcallout
1	d2d_deskew_pll_locked_gfh1	1’b0	R	GFH1 D2D Deskew calibration settled
0	d2d_pll_locked_gfh1	1’b0	R	GFH1 D2D PLL locked

6.1.44 D2D_DDR2_PLL_STATUS, 0x26000718

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18'b0	R	Reserved, always read 'b0
13:2	d2d_dskewcallout_gfh2	12'h0	R	GFH2 D2D dskewcallout
1	d2d_deskew_pll_locked_gfh2	1'b0	R	GFH2 D2D Deskew calibration settled
0	d2d_pll_locked_gfh2	1'b0	R	GFH2 D2D PLL locked

6.1.45 D2D_DDR3_PLL_STATUS, 0x2600071C

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18'b0	R	Reserved, always read 'b0
13:2	d2d_dskewcallout_gfh3	12'h0	R	GFH3 D2D dskewcallout
1	d2d_deskew_pll_locked_gfh3	1'b0	R	GFH3 D2D Deskew calibration settled
0	d2d_pll_locked_gfh3	1'b0	R	GFH3 D2D PLL locked

6.1.46 MKB_D2D_PLL_STATUS, 0x26000720

Bits	Bit Name	Default	Type	Comment
31:4	Reserved	28'b0	R	Reserved, always read 'b0
3	deskew_pll_locked1	1'b0	R	MKB_D2D1 Deskew calibration settled
2	pll_locked1	1'b0	R	MKB_D2D1 PLL locked
1	deskew_pll_locked0	1'b0	R	MKB_D2D0 Deskew calibration settled
0	pll_locked0	1'b0	R	MKB_D2D0 PLL locked

6.1.47 TS_CTRL, 0x26000760

Bits	Bit Name	Default	Type	Comment
31	an_en_5	1'b0	R/W	Enables Analog Access
30	cload_5	1'b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]
29	pd_5	1'b1	R/W	Active high power-down for the analog core
28	an_en_4	1'b0	R/W	Enables Analog Access
27	cload_4	1'b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]
26	pd_4	1'b1	R/W	Active high power-down for the analog core
25	an_en_3	1'b0	R/W	Enables Analog Access
24	cload_3	1'b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]
23	pd_3	1'b1	R/W	Active high power-down for the analog core
22	an_en_2	1'b0	R/W	Enables Analog Access
21	cload_2	1'b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]

Bits	Bit Name	Default	Type	Comment
20	pd_2	1’b1	R/W	Active high power-down for the analog core
19	an_en_1	1’b0	R/W	Enables Analog Access
18	cload_1	1’b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]
17	pd_1	1’b1	R/W	Active high power-down for the analog core
16	an_en_0	1’b0	R/W	Enables Analog Access
15	cload_0	1’b0	R/W	0 = normal operation 1 = load new values from cfg [7:0]
14	pd_0	1’b1	R/W	Active high power-down for the analog core
13:0	an_sel[3:0]	4’b0	R/W	Analog Select Control Set as 4’b0000 for normal operation
9:2	cfg[7:0]	8’h0	R/W	Configuration Selection cfg [7:5] = 3’b000 on dout [11:0] = 3’b001 ±0.4 C on dout [11:2] = 3’b010 ±1.5 C on dout [11:4] = Others Not allowed cfg [4] = 1’b0 Parallel Output = 1’b1 Serial Output cfg [3:0] = 4’b0000 Mode 1 Operation = 4’b0001 Mode 2 Operation
1	run	1’b0	R/W	Active high conversion enable
0	rstn	1’b0	R/W	Asynchronous active low reset

6.1.48 TS_OUT_1, 0x26000764

Bits	Bit Name	Default	Type	Comment
31	Reserved	1’b0	R	Reserved, always read ‘b0
30	faultn_1	1’b0	R	Fault flag Indicates an internal fault when 1’b0. Normally a ‘1’
29	dout_type_1	1’b0	R	Indicates that the data output is temperature data when 1’b0. Is set to 1’b1 in analog access, signature select and fault debug
28	rdy_latched_1	1’b0	R	Active high end of conversion pulse
27:16	Dout_1 [11:0]	12’h0	R	Conversion data
15	Reserved	1’b0	R	Reserved, always read ‘b0
14	faultn_0	1’b0	R	Fault flag Indicates an internal fault when 1’b0. Normally a ‘1’
13	dout_type_0	1’b0	R	Indicates that the data output is temperature data when 1’b0. Is set to 1’b1 in analog access, signature select and fault debug
12	rdy_latched_0	1’b0	R	Active high end of conversion pulse
11:0	Dout_0 [11:0]	12’h0	R	Conversion data

6.1.49 TS_OUT_2, 0x26000768

Bits	Bit Name	Default	Type	Comment
31	Reserved	1’b0	R	Reserved, always read ‘b0
30	faultn_3	1’b0	R	Fault flag Indicates an internal fault when 1’b0. Normally a ‘1’

Bits	Bit Name	Default	Type	Comment
29	dout_type_3	1'b0	R	Indicates that the data output is temperature data when 1'b0. Is set to 1'b1 in analog access, signature select and fault debug
28	rdy_latched_3	1'b0	R	Active high end of conversion pulse
27:16	Dout_3 [11:0]	12'h0	R	Conversion data
15	Reserved	1'b0	R	Reserved, always read 'b0
14	faultn_2	1'b0	R	Fault flag Indicates an internal fault when 1'b0. Normally a '1'
13	dout_type_2	1'b0	R	Indicates that the data output is temperature data when 1'b0. Is set to 1'b1 in analog access, signature select and fault debug
12	rdy_latched_2	1'b0	R	Active high end of conversion pulse
11:0	Dout_2 [11:0]	12'h0	R	Conversion data

6.1.50 TS_OUT_3, 0x2600076C

Bits	Bit Name	Default	Type	Comment
31	Reserved	1'b0	R	Reserved, always read 'b0
30	faultn_5	1'b0	R	Fault flag Indicates an internal fault when 1'b0. Normally a '1'
29	dout_type_5	1'b0	R	Indicates that the data output is temperature data when 1'b0. Is set to 1'b1 in analog access, signature select and fault debug
28	rdy_latched_5	1'b0	R	Active high end of conversion pulse
27:16	Dout_5 [11:0]	12'h0	R	Conversion data
15	Reserved	1'b0	R	Reserved, always read 'b0
14	faultn_4	1'b0	R	Fault flag Indicates an internal fault when 1'b0. Normally a '1'
13	dout_type_4	1'b0	R	Indicates that the data output is temperature data when 1'b0. Is set to 1'b1 in analog access, signature select and fault debug
12	rdy_latched_4	1'b0	R	Active high end of conversion pulse
11:0	Dout_4 [11:0]	12'h0	R	Conversion data

6.1.51 VS_CTRL, 0x26000780

Bits	Bit Name	Default	Type	Comment
31:24	Reserved	8'b0	R/W	Reserved
23	cload	1'b0	R/W	Configuration Load Set to 1'b1 to load new values from cfgn inputs
22	pd	1'b1	R/W	Active high power-down for the analog core
21:18	an_sel [3:0]	4'b0	R/W	Analog Select Control Set as 4'b0000 for normal operation
17:10	cfg2 [7:0]	8'b0	R/W	Sets the input source cfg2 [7] ext_ref cfg2 [6:5] Reserved cfg2 [4:0] input_sel
9:2	cfg1 [7:0]	8'b0	R/W	Sets output modes of voltage monitor cfg1 [7] Reserved cfg1 [6:5] resolution cfg1 [4] ser_mode cfg1 [3:2] pri_mode cfg1[1:0] sec_mode
1	run	1'b0	R/W	Active high conversion enable

Bits	Bit Name	Default	Type	Comment
0	rstn	8’b0	R/W	Asynchronous active low reset

6.1.52 VS_OUT, 0x26000784

Bits	Bit Name	Default	Type	Comment
31:17	Reserved	15’b0	R	Reserved, always read ‘b0
16	dout_type	1’b0	R	1'b0 Indicates that the data is valid voltage monitor data. 1’b1 indicates analog access, signature select and status output modes
15	faultn	1’b0	R	Fault flag Indicates an internal fault when 1’b0. Normally a ‘1’
14	rdy_latched	1’b0	R	Conversion data
13:0	dout [13:0]	14’h0	R	Conversion data

6.1.53 LVDS_CTRL, 0x26000790

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R/W	Reserved
7:5	RTERM_VAL [2:0]	3’b100	R/W	Termination value set bits. This should always be set to the correct value for 100 Ohms. typical case termination value: 3’b000 –> 79 ohm 3’b001 –> 83 ohm 3’b010 –> 87 ohm 3’b011 –> 92 ohm 3’b100 –> 98 ohm 3’b101 –> 104 ohm 3’b110 –> 111 ohm 3’b111 –> 120 ohm
4	VBIAS_SEL	1’b0	R/W	Select bias voltage for I/O 1’b1 –> From external bandgap reference gen on vbias in 1’b0 –> From internal resistor divide
3	BIAS_EN	1’b1	R/W	Bias circuit enable 0 –> Bias circuit disabled 1 –> Bias circuit enabled
2	RXCM_EN	1’b0	R/W	Enable receiver common mode generation internally. 1’b0 –> Internal common mode generation disabled 1’b1 –> Internal common mode generation enabled
1	RTERM_EN	1’b0	R/W	Termination Resistor enable/disable 0 –> Resistor Termination disabled 1 –> Resistor Termination enabled
0	RXEN	1’b1	R/W	LVDS receiver enable 0 –> Receiver disabled 1 –> Receiver enabled

6.1.54 DRO_0_1, x26000800

Bits	Bit Name	Default	Type	Comment
31:24	dro1_mea [7:0]	8’h0	R	DRO_1 measurement Result
23:18	Reserved	6’b0	R	Reserved, always read ‘b0
17	dro1_ready	1’b0	R	DRO_1 ready

Bits	Bit Name	Default	Type	Comment
16	dro1_start	1’b0	R/W	DRO_1 start
15:8	dro0_mea [7:0]	8’h0	R	DRO_0 measurement Result
7:2	Reserved	6’b0	R	Reserved, always read ‘b0
1	dro0_ready	1’b0	R	DRO_0 ready
0	dro0_start	1’b0	R/W	DRO_0 start

6.1.55 DRO_2_3, 0x26000804

Bits	Bit Name	Default	Type	Comment
31:24	dro3_mea [7:0]	8’h0	R	DRO_3 measurement Result
23:18	Reserved	6’b0	R	Reserved, always read ‘b0
17	dro3_ready	1’b0	R	DRO_3 ready
16	dro3_start	1’b0	R/W	DRO_3 start
15:8	dro2_mea [7:0]	8’h0	R	DRO_2 measurement Result
7:2	Reserved	6’b0	R	Reserved, always read ‘b0
1	dro2_ready	1’b0	R	DRO_2 ready
0	dro2_start	1’b0	R/W	DRO_2 start

6.1.56 DRO_4_5, 0x26000808

Bits	Bit Name	Default	Type	Comment
31:24	dro5_mea [7:0]	8’h0	R	DRO_5 measurement Result
23:18	Reserved	6’b0	R	Reserved, always read ‘b0
17	dro5_ready	1’b0	R	DRO_5 ready
16	dro5_start	1’b0	R/W	DRO_5 start
15:8	dro4_mea [7:0]	8’h0	R	DRO_4 measurement Result
7:2	Reserved	6’b0	R	Reserved, always read ‘b0
1	dro4_ready	1’b0	R	DRO_4 ready
0	dro4_start	1’b0	R/W	DRO_4 start

6.1.57 µC_SYSCTRL_PID0, 0x26000FE0

Bits	Bit Name	Default	Type	Comment
31	Reserved	1’b0	R	Reserved, always read ‘b0
30:27	REVISION [3:0]	4’h1	R	Revision
26:16	JEPID [10:0]	11’h489	R	JTAG ID
15:0	PID [15:0]	16’h2	R	Part ID: 02-Bouffalo Lab IP

6.1.58 µC_SYSCTRL_PID1, 0x26000FE4

Bits	Bit Name	Default	Type	Comment
31:4	Reserved	28’b0	R	Reserved, always read ‘b0
3:0	ECOREVNUM	4’h0	R	ECO Revision

6.1.59 µC_SYSCTRL_CID0, 0x26000FF0

Bits	Bit Name	Default	Type	Comment
31:0	CID0	32’h42495244	R	Chip ID0 ("BIRD")

6.1.60 µC_SYSCTRL_CID1, 0x26000FF4

Bits	Bit Name	Default	Type	Comment
31:0	CID1	32’h4B494E47	R	Chip ID1 ("KING")

6.1.61 µC_SYSCTRL_CID2, 0x26000FF8

Bits	Bit Name	Default	Type	Comment
31:0	CID2	32’h20204D4F	R	Chip ID2 ("MO")

6.1.62 µC_SYSCTRL_CID3, 0x26000FFC

Bits	Bit Name	Default	Type	Comment
31:0	CID3	32’h464C4354	R	Chip ID3 ("FLCT")

6.2 Serial Flash Control Register

Serial flash CSR address = `0x2600_1000 ~ 0x2600_1FFF`

6.2.1 SF_CTRL_0, 0x26001000

Bits	Bit Name	Default	Type	Comment
31:24	sf_id	8'h1A	R/W	ID
23	sf_aes_iv_endian	1’b1	R/W	0: Little-endian 1: Big-endian
22	sf_aes_key_endian	1’b1	R/W	0: Little-endian 1: Big-endian
21	sf_aes_din_endian	1’b1	R/W	0: Little-endian 1: Big-endian
20	sf_aes_dout_endian	1’b1	R/W	0: Little-endian 1: Big-endian
19	sf_if_32b_adr_en	1’b0	R/W	1: Enable 32-bit address
18	sf_if_int_set	1’b0	R/W	1: Set interrupt
17	sf_if_int_clr	1’b1	R/W	1: Clear interrupt
16	sf_if_int	1’b0	R	Interrupt value
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11	sf_if_read_dly_en	1’b0	R/W	1: Enable flash controller read delay mode
10:8	sf_if_read_dly_n	3’b0	R/W	Flash controller read delay cycle = n + 1
7:5	Reserved	3’b0	R	Reserved, always read ‘b0
4	sf_clk_out_inv_sel	1’b1	R/W	1: Select inverted clock out
3	sf_clk_out_gate_en	1’b1	R/W	1: Hardware auto gated clock out
2	sf_clk_sf_rx_inv_sel	1’b1	R/W	1: Select inverted flash Rx clock
1:0	Reserved	2’b0	R	Reserved, always read ‘b0

6.2.2 SF_CTRL_1, 0x26001004

Bits	Bit Name	Default	Type	Comment
31	sf_ahb2sram_en	1’b1	R/W	1: Enable SRAM
30	sf_ahb2sif_en	1’b1	R/W	1: Enable IAHB to flash interface
29	sf_if_en	1’b1	R/W	0: Disable sf_if 1: enable sf_if
28	sf_if_fn_sel	1’b1	R/W	0: System AHB 1: icache AHB
27	sf_ahb2sif_stop	1’b0	R/W	1: Stop CPU access flash
26	sf_ahb2sif_stopped	1’b0	R	1: CPU stopped
25	sf_if_reg_wp	1’b1	R/W	Write protect
24	sf_if_reg_hold	1’b1	R/W	Hold
23	sf_ahb2sif_diswrap	1’b0	R/W	1: Disable IAHB to flash wrap access for XTS mode
22:20	sf_if_0_ack_lat	3’b110	R/W	ACK latency cycles
19	Reserved	1’b0	R	Reserved, always read ‘b0
18	sf_if_sr_int_set	1’b0	R/W	1: Set interrupt

Bits	Bit Name	Default	Type	Comment
17	sf_if_sr_int_en	1’b0	R/W	1: Status read interrupt enable
16	sf_if_sr_int	1’b0	R	Interrupt value
15:8	sf_if_sr_pat	8’b0	R/W	Interrupt pattern
7:0	sf_if_sr_pat_mask	1’b0	R/W	Interrupt mask

6.2.3 SF_IF_SAHB_0, 0x26001008

Bits	Bit Name	Default	Type	Comment
31	sf_if_0_qpi_mode_en	1’b0	R/W	0: Normal SPI 1: QPI mode enable
30:28	sf_if_0_spi_mode	3’b000	R/W	000: Normal SPI 001: Dual output 010: Quad output 011: Dual IO 100: Quad IO
27	sf_if_0_cmd_en	1’b1	R/W	1: Command enable
26	sf_if_0_adr_en	1’b1	R/W	1: Address enable
25	sf_if_0_dmy_en	1’b0	R/W	1: Dummy enable
24	sf_if_0_dat_en	1’b1	R/W	1: Data enable
23	sf_if_0_dat_rw	1’b0	R/W	0: Read 1: write
22:20	sf_if_0_cmd_byte	3’b000	R/W	Number of command bytes minus 1
19:17	sf_if_0_adr_byte	3’b010	R/W	Number of address bytes minus 1
16:12	sf_if_0_dmy_byte	5’b0	R/W	Number of dummy bytes minus 1
11:2	sf_if_0_dat_byte	10’hFF	R/W	Number of command byte minus 1
1	sf_if_0_trig	1’b0	R/W	1: Trigger sf_if FSM
0	sf_if_busy	1’b0	R	1: Busy

6.2.4 SF_IF_SAHB_1, 0x2600100C

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_0_cmd_buf_0	32'h03000000	R/W	Command buffer 0

6.2.5 SF_IF_SAHB_2, 0x26001010

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_0_cmd_buf_1	32’h0	R/W	Command buffer 1

6.2.6 SF_IF_IAHB_0, 0x26001014

Bits	Bit Name	Default	Type	Comment
31	sf_if_1_qpi_mode_en	1’b0	R/W	0: Normal SPI 1: QPI mode enable

Bits	Bit Name	Default	Type	Comment
30:28	sf_if_1_spi_mode	3'b000	R/W	000: Normal SPI 001: Dual output 010: Quad output 011: Dual IO 100: Quad IO
27	sf_if_1_cmd_en	1'b1	R/W	1: Command enable
26	sf_if_1_adr_en	1'b1	R/W	1: Address enable
25	sf_if_1_dmy_en	1'b0	R/W	1: Dummy enable
24	sf_if_1_dat_en	1'b1	R/W	1: Data enable
23	sf_if_1_dat_rw	1'b0	R/W	0: Read 1: write
22:20	sf_if_1_cmd_byte	3'b000	R/W	Number of command bytes minus 1
19:17	sf_if_1_adr_byte	3'b010	R/W	Number of address bytes minus 1
16:12	sf_if_1_dmy_byte	5'b0	R/W	Number of dummy bytes minus 1
11:0	Reserved	12'b0	R	Reserved, always read 'b0

6.2.7 SF_IF_IAHB_1, 0x26001018

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_1_cmd_buf_0	32'h03000000	R/W	Command buffer 0

6.2.8 SF_IF_IAHB_2, 0x2600101C

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_1_cmd_buf_1	32'h0	R/W	Command buffer 1

6.2.9 SF_IF_STATUS_0, 0x26001020

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_status_0	32'h0	R	SF interface Status 0

6.2.10 SF_IF_STATUS_1, 0x26001024

Bits	Bit Name	Default	Type	Comment
31:0	sf_if_status_1	32'h20000000	R	SF interface Status 1

6.2.11 SF_AHB2SIF_STATUS, 0x2600102C

Bits	Bit Name	Default	Type	Comment
31:0	sf_ahb2sif_status	32'h1010003	R	AHB2SIF Status

6.2.12 SF_IF_IO_DLY_0, 0x26001030

Bits	Bit Name	Default	Type	Comment
31:30	sf_dqs_do_dly_sel	2’b0	R/W	IO delay selection
29:28	sf_dqs_di_dly_sel	2’b0	R/W	IO delay selection
27:26	sf_dqs_oe_dly_sel	2’b0	R/W	IO delay selection
25:10	Reserved	16’b0	R	Reserved, always read ‘b0
9:8	sf_clk_out_dly_sel	2’b0	R/W	IO delay selection
7:4	Reserved	4’b0	R	Reserved, always read ‘b0
3:2	sf_cs2_dly_sel	2’b0	R/W	IO delay selection
1:0	sf_cs_dly_sel	2’b0	R/W	IO delay selection

6.2.13 SF_IF_IO_DLY_1, 0x26001034

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14’b0	R	Reserved, always read ‘b0
17:16	sf_io_0_do_dly_sel	2’b0	R/W	IO delay selection
15:10	Reserved	6’b0	R	Reserved, always read ‘b0
9:8	sf_io_0_di_dly_sel	2’b0	R/W	IO delay selection
7:2	Reserved	6’b0	R	Reserved, always read ‘b0
1:0	sf_io_0_oe_dly_sel	2’b0	R/W	IO delay selection

6.2.14 SF_IF_IO_DLY_2, 0x26001038

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14’b0	R	Reserved, always read ‘b0
17:16	sf_io_1_do_dly_sel	2’b0	R/W	IO delay selection
15:10	Reserved	6’b0	R	Reserved, always read ‘b0
9:8	sf_io_1_di_dly_sel	2’b0	R/W	IO delay selection
7:2	Reserved	6’b0	R	Reserved, always read ‘b0
1:0	sf_io_1_oe_dly_sel	2’b0	R/W	IO delay selection

6.2.15 SF_IF_IO_DLY_3, 0x2600103C

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14’b0	R	Reserved, always read ‘b0
17:16	sf_io_2_do_dly_sel	2’b0	R/W	IO delay selection
15:10	Reserved	6’b0	R	Reserved, always read ‘b0
9:8	sf_io_2_di_dly_sel	2’b0	R/W	IO delay selection
7:2	Reserved	6’b0	R	Reserved, always read ‘b0

Bits	Bit Name	Default	Type	Comment
1:0	sf_io_2_oe_dly_sel	2'b0	R/W	IO delay selection

6.2.16 SF_IF_IO_DLY_4, 0x26001040

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14'b0	R	Reserved, always read 'b0
17:16	sf_io_3_do_dly_sel	2'b0	R/W	IO delay selection
15:10	Reserved	6'b0	R	Reserved, always read 'b0
9:8	sf_io_3_di_dly_sel	2'b0	R/W	IO delay selection
7:2	Reserved	6'b0	R	Reserved, always read 'b0
1:0	sf_io_3_oe_dly_sel	2'b0	R/W	IO delay selection

6.2.17 SF_RESERVED, 0x26001044

Bits	Bit Name	Default	Type	Comment
31:0	sf_reserved	32'hFFFF	R/W	Reserved

6.2.18 SF_CTRL_2, 0x26001070

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7	sf_id_offset_lock	1'b0	R/W	SF ID offset lock
6:0	Reserved	7'b0	R	Reserved, always read 'b0

6.2.19 SF_CTRL_3, 0x26001074

Bits	Bit Name	Default	Type	Comment
31:29	sf_if_1_ack_lat	3'b001	R/W	ACK latency cycles
28:21	Reserved	8'b0	R	Reserved, always read 'b0
20	sf_cmds_core_en	1'b1	R/W	1: Enable command splitter core
19:18	sf_cmds_1_wrap_mode [1:0]	2'b0	R/W	[0] 0: cmds bypass wrap commands to macro 1: cmds handle wrap commands. [1] 0: original mode 1: cmds force wrap16x4 is split into two wrap 8x4
17	sf_cmds_1_en	1'b0	R/W	1: Enable command splitter
16:13	sf_cmds_1_wrap_len	4'd6	R/W	Wrap length 0:8, 1:16, 2:32, 3:64, 4:128, 5:256, 6:512, 7:1024, 8:2048, 9:4096
12:0	Reserved	13'b0	R	Reserved, always read 'b0

6.2.20 SF_ID0_OFFSET, 0x260010A0

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4'b0	R	Reserved, always read 'b0
27:0	sf_id0_offset	28'b0	R/W	SF ID offset, will be locked if sf_id_offset_lock(0x26001070[7]) = 1'b1

6.2.21 SF_DBG, 0x260010B0

Note: This register is reserved, always read 'b0

Bits	Bit Name	Default	Type	Comment
31:6	Reserved	26'b0	R	Reserved, always read 'b0
5	sf_autoload_st_done	1'b0	R	sf_autoload_st_done, hardwired 1'b0
4:0	sf_autoload_st	5'd1	R	sf_autoload_st, hardwired 5'd1

6.2.22 SF_CTRL_PROT_EN_RD, 0x26001100

Note: This register is reserved, always read 'b0

Bits	Bit Name	Default	Type	Comment
31	sf_dbg_dis	1'b0	R	sf_dbg_dis, hardwired 1'b0
30	sf_if_0_trig_wr_lock	1'b0	R	sf_if_0_trig_wr_lock, hardwired 1'b0
29	Reserved	1'b0	R	Reserved, always read 'b0
28	sf_sec_tzsid_lock	1'b0	R	sf_sec_tzsid_lock, hardwired 1'b0
27:0	Reserved	28'b0	R	Reserved, always read 'b0

6.2.23 SF_SRAM, 0x26001600~0x2600167F

Bits	Bit Name	Default	Type	Comment
31:0	SF SRAM Data	32'bX	R/W	SF SRAM (128x32 bits) Address[6:0] = SF SRAM address (00~7F)

6.3 L1C Control Register

L1C CSR address = 0x2600_2000 ~ 0x2600_2FFF

6.3.1 L1C_CONFIG, 0x26002000

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18’b0	R/W	Reserved
13:12	sf_clk_sel [1:0]	2’b00	R/W	00 = 25MHz 01 = 50MHz 1X = 100MHz
11:8	l1c_way_dis	4’b0	R/W	Disable part of cache ways & used as AHB SRAM
7:2	Reserved	6’b0	R/W	Reserved
1	l1c_cnt_en	1’b0	R/W	Cache performance counter enable
0	l1c_cacheable	1’b0	R/W	Cacheable regions enable

6.3.2 L1C_HIT_CNT_LSB, 0x26002004

Bits	Bit Name	Default	Type	Comment
31:0	l1c_hit_cnt_lsb	32’h0	R	Low 32-bit hit counter

6.3.3 L1C_HIT_CNT_MSB, 0x26002008

Bits	Bit Name	Default	Type	Comment
31:0	l1c_hit_cnt_msb	32’h0	R	High 32-bit hit counter

6.3.4 L1C_MISS_CNT, 0x2600200C

Bits	Bit Name	Default	Type	Comment
31:0	l1c_miss_cnt	32’h0	R	Miss counter

6.3.5 L1C_AUX_CONFIG, 0x26002010

Bits	Bit Name	Default	Type	Comment
31	l1c_invalid_done	1’b0	R	l1c invalid done
30:4	Reserved	27’b0	R/W	Reserved
3	early_resp_dis	1’b0	R/W	early_resp_dis
2	xip_2t_access	1’b0	R/W	Set 1 for ROM 2T access if CPU freq >72MHz
1	l1c_invalid	1’b0	R/W	l1c invalid
0	l1c_en	1’b0	R/W	1: burst=3’b100 0: burst=3’b001

6.4 DMA Control Register

DMA CSR address = `0x2600_4000 ~ 0x2600_4FFF`

6.4.1 DMA_INTSTATUS, 0x26004000

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	INTSTA	8’b0	R	Status of DMA interrupts after masking

6.4.2 DMA_INTTCSTATUS, 0x26004004

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	INTTCSTA	8’b0	R	Interrupt terminal count request status

6.4.3 DMA_INTTCCLEAR, 0x26004008

Note: This register always read 32’b0

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	TCRC	8’b0	W	Terminal count request clear

6.4.4 DMA_INTERRORSTATUS, 0x2600400C

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	IES	8’b0	R	Interrupt error status

6.4.5 DMA_INTERRCLR, 0x26004010

Note: This register always read 32’b0

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	IEC	8’b0	W	Interrupt error clear

6.4.6 DMA_RAWINTTCSTATUS, 0x26004014

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	SOTCIPTM	8’b0	R	Status of terminal counter interrupt prior to masking

6.4.7 DMA_RAWINTERRORSTATUS, 0x26004018

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	SOTEIPTM	8’b0	R	Status of error interrupts prior to masking

DMA_ENBLDCHNS, 0x2600401C

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	CES	8’b0	R	Channel enable status

6.4.8 DMA_SOFTBREQ, 0x26004020

Bits	Bit Name	Default	Type	Comment
31:0	SBR	32’h0	R/W	Software burst request

6.4.9 DMA_SOFTSRE, 0x26004024

Bits	Bit Name	Default	Type	Comment
31:0	SSR	32’h0	R/W	Software single request

6.4.10 DMA_SOFTLBREQ, 0x26004028

Bits	Bit Name	Default	Type	Comment
31:0	SLBR	32’h0	R/W	Software last burst request

6.4.11 DMA_SOFTLSREQ, 0x2600402C

Bits	Bit Name	Default	Type	Comment
31:0	SLSR	32’h0	R/W	Software last single request

6.4.12 DMA_CONFIG, 0x26004030

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0
1	AHBMEC	1’b0	R/W	AHB Master endianness configuration: 0=little-endian, 1=big-endian.
0	SMDMAEN	1’b0	R/W	SMDMA enable

6.4.13 DMA_SYNC, 0x26004034

Bits	Bit Name	Default	Type	Comment
31:0	DSLFDRS	32’h0	R/W	DMA synchronization logic for DMA request signals. 0=enable, 1=disable.

6.4.14 DMA_C[3:0]SRCADDR, 0x26004[4:1]00

Bits	Bit Name	Default	Type	Comment
31:0	DMASA	32'h0	R/W	DMA source address

6.4.15 DMA_C[3:0]DSTADDR, 0x26004[4:1]04

Bits	Bit Name	Default	Type	Comment
31:0	DMADA	32'h0	R/W	DMA destination address

6.4.16 DMA_C[3:0]LLI, 0x26004[4:1]08

Bits	Bit Name	Default	Type	Comment
31:0	FLLI	32'h0	R/W	First linked list item. Bits [1:0] must be 0

6.4.17 DMA_C[3:0]CONTROL, 0x26004[4:1]0C

Bits	Bit Name	Default	Type	Comment
31	TCIEN	1'b0	R/W	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30:28	PROTECT	3'b000	R/W	Protection
27	DI	1'b1	R/W	Destination increment. When set the destination address is incremented after each transfer
26	SI	1'b1	R/W	Source Increment. When set the source address is incremented after each transfer
25	Reserved	N/A	N/A	Reserved
24:23	FIXCNT	2'b00	R/W	Only effect when dst_min_mode = 1 Destination transfer cnt = (total src byte cnt - (fix_cnt«DWidth))«DWidth
22:21	DTW	2'b10	R/W	Destination transfer width: 8/16/32
20	Reserved	N/A	N/A	Reserved
19:18	STW	2'b10	R/W	Source transfer width: 8/16/32
17	ADDMODE	1'b0	R/W	Add mode: issue remain destination traffic
16:15	DBS	2'b01	R/W	Destination burst size: 1/4/8/16
14	MINMODE	1'b0	R/W	Minus mode. Not issue all destination traffic.
13:12	SBS	2'b01	R/W	Source burst size: 1/4/8/16. Note CH FIFO Size is 16Bytes and SBSize*Swidth should <= 16B
11:0	TS	12'h0	R/W	Transfer size: 0:4095: Number of data transfer left to complete when SMDMA is the flow controller.

6.4.18 DMA_C[3:0]CONFIG, 0x26004[4:1]10

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	N/A	N/A	Reserved
29:20	LLICOUNT	10'b0	R	LLI counter: Increased 1 each LLI run. Cleared 0 when config control.

Bits	Bit Name	Default	Type	Comment
19	Reserved	N/A	N/A	Reserved
18	HALT	1’b0	R/W	Halt. 0=enable DMA requests, 1=ignore subsequent source DMA requests.
17	ACTIVE	1’b0	R	Active. 0=no data in FIFO of the channel, 1=FIFO of the channel has data.
16	LOCK	1’b0	R/W	Lock
15	TCIM	1’b0	R/W	Terminal count interrupt mask
14	IEM	1’b0	R/W	Interrupt error mask
13:11	FLOWCTRL	3’b000	R/W	000: Memory to memory DMA 001: Memory to peripheral DMA 010: Peripheral to memory DMA 011: Source peripheral to destination peripheral DMA.
10:6	DSTPH	5’h0	R/W	Destination peripherals: 0: UART TX 1: UART RX 2: SPI0 TX 3: SPI0 RX 4: SPI1 TX 5: SPI1 RX 6: I2C TX 7: I2C RX Others: Reserved
5:1	SRCPH	5’h0	R/W	Source peripherals: 0: UART TX 1: UART RX 2: SPI0 TX 3: SPI0 RX 4: SPI1 TX 5: SPI1 RX 6: I2C TX 7: I2C RX Others: Reserved
0	CHEN	1’b0	R/W	Channel enable.

6.5 Security Engine Control Register

Security Engine CSR address = `0x2600_5000 ~ 0x2600_5FFF`

6.5.1 SE_SHA_0_CTRL, 0x26005000

Bits	Bit Name	Default	Type	Comment
31:16	se_sha_0_msg_len	16'h0	R/W	Number of 512-bit block
15:12	Reserved	NA	NA	Reserved
11	se_sha_0_int_mask	1'b0	R/W	se_sha_0 interrupt mask
10	se_sha_0_int_set_1t	1'b0	W1p	1: Set Interrupt
9	se_sha_0_int_clr_1t	1'b0	W1p	1: Clear Interrupt
8	se_sha_0_int	1'b0	R	Interrupt value
7	Reserved	NA	NA	Reserved
6	se_sha_0_hash_sel	1'b0	R/W	0: New Hash 1: Accumulate Last Hash
5	se_sha_0_en	1'b0	R/W	1: Enable SHA Engine
4:2	se_sha_0_mode	3'b000	R/W	000: SHA-256 001: SHA-224 010: SHA-1 011: SHA-1 100: SHA-512 101: SHA-384 110: SHA-512/224 111: SHA-512/256
1	se_sha_0_trig_1t	1'b0	W1p	1: Trigger sha Engine
0	se_sha_0_busy	1'b0	R	1: sha Engine Busy

6.5.2 SE_SHA_0_MSA, 0x26005004

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_msa	32'h0	R/W	Message Source Address

6.5.3 SE_SHA_0_STATUS, 0x26005008

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_status	32'h41	R	

6.5.4 SE_SHA_0_ENDIAN, 0x2600500C

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	N/A	N/A	Reserved
0	se_sha_0_dout_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian

6.5.5 SE_SHA_0_HASH_L_0, 0x26005010

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_0	32’h0	R	Big-Endian Hash 0 (MSB)

6.5.6 SE_SHA_0_HASH_L_1, 0x26005014

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_1	32’h0	R	Big-Endian Hash 1

6.5.7 SE_SHA_0_HASH_L_2, 0x26005018

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_2	32’h0	R	Big-Endian Hash 2

6.5.8 SE_SHA_0_HASH_L_3, 0x2600501C

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_3	32’h0	R	Big-Endian Hash 3

6.5.9 SE_SHA_0_HASH_L_4, 0x26005020

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_4	32’h0	R	Big-Endian Hash 4

6.5.10 SE_SHA_0_HASH_L_5, 0x26005024

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_5	32’h0	R	Big-Endian Hash 5

6.5.11 SE_SHA_0_HASH_L_6, 0x26005028

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_6	32’h0	R	Big-Endian Hash 6

6.5.12 SE_SHA_0_HASH_L_7, 0x2600502C

Bits	Bit Name	Default	Type	Comment
31:0	se_sha_0_hash_l_7	32’h0	R	Big-Endian Hash 7 (LSB)

6.5.13 SE_SHA_0_CTRL_PROT, 0x260050FC

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	NA	NA	Reserved

Bits	Bit Name	Default	Type	Comment
2	se_sha_id1_en	1’b1	R/W	id1 Access Right
1	se_sha_id0_en	1’b1	R/W	id0 Access Right
0	Reserved	NA	NA	Reserved

6.5.14 SE_AES_0_CTRL, 0x26005100

Bits	Bit Name	Default	Type	Comment
31:16	se_aes_0_msg_len	16’h0	R/W	Number of 128-bit Block
15	Reserved	NA	NA	Reserved
14	se_aes_0_iv_sel	1’b0	R/W	0: New iv 1: Same iv as Last One
13:12	se_aes_0_block_mode	2’b00	R/W	00: ECB mode 01: CTR mode 10: CBC mode 11: XTS mode
11	se_aes_0_int_mask	1’b0	R/W	se_aes_0 interrupt mask
10	se_aes_0_int_set_1t	1’b0	W1p	1: Set Interrupt
9	se_aes_0_int_clr_1t	1’b0	W1p	1: Clear Interrupt
8	se_aes_0_int	1’b0	R	Interrupt Value
7	se_aes_0_hw_key_en	1’b0	R/W	0: sw Key 1: hw Key
6	se_aes_0_dec_key_sel	1’b0	R/W	0: New Key 1: Same Key as Last One
5	se_aes_0_dec_en	1’b0	R/W	0: Encode 1: Decode
4:3	se_aes_0_mode	2’b00	R/W	00: 128-bit Mode 01: 256-bit Mode 10: 192-bit Mode 11: 128-bit Double Key Mode
2	se_aes_0_en	1’b0	R/W	1: Enable
1	se_aes_0_trig_1t	1’b0	W1p	1: Trigger aes Engine
0	se_aes_0_busy	1’b0	R	1: aes Engine Busy

6.5.15 SE_AES_0_MSA, 0x26005104

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_msa	32’h0	R/W	Message Source Address

6.5.16 SE_AES_0_MDA, 0x26005108

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_mda	32’h0	R/W	Message Destination Address

6.5.17 SE_AES_0_STATUS, 0x2600510C

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_status	32'h4100	R	

6.5.18 SE_AES_0_IV_0, 0x26005110

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_iv_0	32’h0	R/W	Big Endian Initial Vector (MSB)

6.5.19 SE_AES_0_IV_1, 0x26005114

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_iv_1	32’h0	R/W	Big Endian Initial Vector

6.5.20 SE_AES_0_IV_2, 0x26005118

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_iv_2	32’h0	R/W	Big Endian Initial Vector

6.5.21 SE_AES_0_IV_3, 0x2600511C

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_iv_3	32’h0	R/W	Big Endian Initial Vector (LSB) (CTR Mode: 32-bit Counter Initial Value)

6.5.22 SE_AES_0_KEY_0, 0x26005120

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_0	32’h0	R/W	Big Endian AES Key (aes-128/256 Key MSB)

6.5.23 SE_AES_0_KEY_1, 0x26005124

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_1	32’h0	R/W	Big Endian AES Key

6.5.24 SE_AES_0_KEY_2, 0x26005128

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_2	32’h0	R/W	Big Endian AES Key

6.5.25 SE_AES_0_KEY_3, 0x2600512C

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_3	32'h0	R/W	Big Endian AES Key (aes-128 key LSB)

6.5.26 SE_AES_0_KEY_4, 0x26005130

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_4	32'h0	R/W	Big Endian AES Key

6.5.27 SE_AES_0_KEY_5, 0x26005134

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_5	32'h0	R/W	Big Endian AES Key

6.5.28 SE_AES_0_KEY_6, 0x26005138

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_6	32'h0	R/W	Big Endian AES Key

6.5.29 SE_AES_0_KEY_7, : 0x2600513C

Bits	Bit Name	Default	Type	Comment
31:0	se_aes_0_key_7	32'h0	R/W	Big Endian AES Key (aes-256 key LSB)

6.5.30 SE_AES_0_ENDIAN, 0x26005148

Bits	Bit Name	Default	Type	Comment
31:30	se_aes_0_ctr_len	2'b00	R/W	00:4-byte Counter 01:1-byte Counter 10:2-byte Counter 11:3-byte Counter
29:5	Reserved	N/A	N/A	Reserved
4	se_aes_0_twk_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian, default 1 for XTS
3	se_aes_0_iv_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian
2	se_aes_0_key_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian
1	se_aes_0_din_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian
0	se_aes_0_dout_endian	1'b1	R/W	0: Little-Endian 1: Big-Endian

6.5.31 SE_TRNG_0_CTRL_0, 0x26005200

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	N/A	N/A	Reserved
11	se_trng_0_int_mask	1’b0	R/W	se_trng_0 interrupt mask
10	se_trng_0_int_set_1t	1’b0	W1p	1: Set Interrupt
9	se_trng_0_int_clr_1t	1’b0	W1p	1: Clear Interrupt
8	se_trng_0_int	1’b0	R	Interrupt Value
7:5	Reserved	N/A	N/A	Reserved
4	se_trng_0_ht_error	1’b0	R	1: Health Test Error
3	se_trng_0_dout_clr_1t	1’b0	W1p	1: Clear trng_dout to Zero
2	se_trng_0_en	1’b0	R/W	1: Enable
1	se_trng_0_trig_1t	1’b0	W1p	1: Trigger trng Engine
0	se_trng_0_busy	1’b0	R	1: trng Engine Busy

6.5.32 SE_TRNG_0_STATUS, 0x26005204

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_status	32'h100020	R	

6.5.33 SE_TRNG_0_DOUT0, 0x26005208

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_0	32’h0	R	random value

6.5.34 SE_TRNG_0_DOUT1, 0x2600520C

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_1	32’h0	R	random value

6.5.35 SE_TRNG_0_DOUT2, 0x26005210

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_2	32’h0	R	random value

6.5.36 SE_TRNG_0_DOUT3, 0x26005214

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_3	32’h0	R	random value

6.5.37 SE_TRNG_0_DOUT4, 0x26005218

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_4	32'h0	R	random value

6.5.38 SE_TRNG_0_DOUT5, 0x2600521C

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_5	32'h0	R	random value

6.5.39 SE_TRNG_0_DOUT6, 0x26005220

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_6	32'h0	R	random value

6.5.40 SE_TRNG_0_DOUT7, 0x26005224

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_dout_7	32'h0	R	random value

6.5.41 SE_TRNG_0_TEST, 0x26005228

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	N/A	N/A	Reserved
1	se_trng_0_cp_test_en	1'b0	R/W	1: Enable trng Conditional Component Test Mode
0	se_trng_0_test_en	1'b0	R/W	1: Enable trng Test Mode

6.5.42 SE_TRNG_0_CTRL_1, 0x2600522C

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_reseed_n_lsb	32'hFFFF	R/W	Reload Seed When Number of Used Random Value is Larger than reseed_n

6.5.43 SE_TRNG_0_CTRL_2, 0x26005230

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	N/A	N/A	Reserved
15:0	se_trng_0_reseed_n_msb	16'hFF	R/W	Reload Seed When Number of Used Random Value is Larger than reseed_n

6.5.44 SE_TRNG_0_CTRL_3, 0x26005234

Bits	Bit Name	Default	Type	Comment
31	se_trng_0_rosc_en	1'b1	R/W	trng rosc Enable

Bits	Bit Name	Default	Type	Comment
30:0	Reserved	N/A	N/A	Reserved

6.5.45 SE_TRNG_0_TEST_OUT_0, 0x26005240

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_test_out_0	32’h0	R	

6.5.46 SE_TRNG_0_TEST_OUT_1, 0x26005244

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_test_out_1	32’h0	R	

6.5.47 SE_TRNG_0_TEST_OUT_2, 0x26005248

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_test_out_2	32’h0	R	

6.5.48 SE_TRNG_0_TEST_OUT_3, 0x2600524C

Bits	Bit Name	Default	Type	Comment
31:0	se_trng_0_test_out_3	32’h0	R	

6.5.49 SE_CTRL_RESERVED_0, 0x26005F04

Bits	Bit Name	Default	Type	Comment
31:0	se_ctrl_reserved_0	32’h0	R/W	Reserved

6.5.50 SE_CTRL_RESERVED_1, 0x26005F08

Bits	Bit Name	Default	Type	Comment
31:0	se_ctrl_reserved_1	32’hFFFFFFFF	R/W	Reserved

6.5.51 SE_CTRL_RESERVED_2, 0x26005F0C

Bits	Bit Name	Default	Type	Comment
31:0	se_ctrl_reserved_2	32’h0	R	Reserved Read SE_CTRL_RESERVED_0

6.6 µC-PCIe Register

6.6.1 µC-PCIe CSR, 0x2601_0000 ~ 0x2601_0FFF

pcie_phy_sel(0x2600_0038[3:0]): 0=RCx8[0]; 5=RCx4[0]; 6=RCx8[1]; 7=RC1x4[1].

6.6.2 PCIe_PHYCONFIG, 0x2601003C

Bits	Bit Name	Default	Type	Comment
31	phy_apb_write_done	1'b0	R	Status bit of setting µC_PHY_apb_write_done (0->1), PHY configuration completed and pipe_reset released
30:26	Reserved	5'b0	R	Reserved, always read 'b0
25	x4_bifurcation_en	1'b0	R/W	1 = enable 2x(4x1) bifurcation mode Note: UNUSED in EP
24	pclk_state_power	1'b1	R/W	
23	perstn	1'b1	R/W	0=Drive PERSTn=0 at RC port.
22	ictl_pma_ref_ls_ena_a	1'b0	R/W	PHY port "ictl_pma_ref_ls_ena_a"
21:20	flc1_axi_slv_config	2'b0	R/W	00 = CXL.mem -> flc1 01 = CXL.io -> flc1 1x = D2D (host) -> flc1
19	tx_lane_flip_en	1'b0	R/W	
18	rx_lane_flip_en	1'b0	R/W	
17	outband_pwrup_cmd	1'b0	R/W	
16	margining_software_ready	1'b0	R/W	
15	margining_ready	1'b0	R/W	
14	pm_xmt_pme	1'b0	R/W	Drives 'apps_pm_xmt_pme'. assert this will request the controller to wakeup from L2
13	dbi_ro_wr_disable	1'b0	R/W	
12	pf_req_retry_en	1'b0	R/W	
11	req_retry_en	1'b0	R/W	
10	xfer_pending	1'b0	R/W	Asserting this will stop ASPM L1 entry
9	req_exit_l1	1'b0	R/W	Write 1 and then 0 will trigger exit from L1.
8	ready_entr_l23	1'b0	R/W	This must be driven high once radm_pm_turnoff interrupt is received to enter L2
7	req_entr_l1	1'b0	R/W	Write 1 and then 0 will trigger entry into L1 when traffic is not there.
6	init_rst	1'b0	R/W	Set this to 0 for EP (no functionality in EP controller)
5	clk_pm_en	1'b1	R/W	
4	clk_req_n	1'b1	R/W	
3	hold_phy_rst	1'b0	R/W	
2	ltssm_enable	1'b0	R/W	Should be enabled after writing all DBI interface registers
1	sris_mode	1'b0	R/W	

Bits	Bit Name	Default	Type	Comment
0	µC_phy_apb_write_done	1’b0	R/W	Firmware set this bit after PHY programming completed

6.6.3 RC_MSTR_ARMISC0, 0x26010118

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_armisc31_0	32’h0	R	iip_mstr_armisc_info[31:0]

6.6.4 RC_MSTR_ARMISC1, 0x2601011C

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_armisc63_32	32’h0	R	[31:20]: Reserved, always read ‘b0 [19]: iip_mstr_armisc_info_last_dcmp_tlp [18]: iip_mstr_armisc_info_zeroread [17:0]: iip_mstr_armisc_info[49:32]

6.6.5 RC_MSTR_AWMISC0, 0x26010120

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_awmisc31_0	32’h0	R	iip_mstr_awmisc_info_hdr_34dw [31:0]

6.6.6 RC_MSTR_AWMISC1, 0x26010124

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_awmisc63_32	32’h0	R	iip_mstr_awmisc_info_hdr_34dw[63:32]

6.6.7 RC_MSTR_AWMISC2, 0x26010128

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_awmisc95_64	32’h0	R	iip_mstr_awmisc_info[31:0]

6.6.8 RC_MSTR_AWMISC3, 0x2601012C

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_awmisc127_96	32’h0	R	[31:20]: Reserved, always read ‘b0 [19]: iip_mstr_awmisc_info_last_dcmp_tlp [18]: iip_mstr_awmisc_info_ep [17:0]: iip_mstr_awmisc_info[49:32] Reading this register will pop awmisc FIFO after reading above 0,1,2, Note: RD_DONE signal unused

6.6.9 RC_MSTR_RESP, 0x26010130

Bits	Bit Name	Default	Type	Comment
31:0	rc_mstr_resp	32'h0	R/W	Response misc signals driven to the EP Controller [23:19]: msg type in awmisc_fifo (This FIFO fills data based on type, wr condition is added) [18:6]: mstr_rmisc_info [5:3]: mstr_rmisc_info_cpl_stat [2:0]: mstr_bmisc_info_cpl_stat

6.6.10 RC_SLV_ADDRH, 0x26010134

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_addrh	32'h0	R/W	slv_addrh

6.6.11 RC_SLV_ARMISC, 0x260101389999

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_armisc	32'h0	R/W	slv_armisc

6.6.12 RC_SLV_RESP, 0x2601013C

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_resp	32'h0	R	[31:22]: Reserved, always read 'b0 [21:11]: rc_slv_bmisc_info[10:0] [10:0]: rc_slv_rmisc_info[10:0]

6.6.13 RC_SLV_AWMISC_31_0, 0x26010140

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_awmisc31_0	32'h0	R/W	miscellanious slv_awmisc slv_awmisc_info_hdr_34dw[31:0]

6.6.14 RC_SLV_AWMISC1, 0x26010144

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_awmisc63_32	32'h0	R/W	miscellanious slv_awmisc slv_awmisc_info_hdr_34dw[63:32]

6.6.15 RC_SLV_AWMISC2, 0x26010148

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_awmisc95_64	32'h0	R/W	[31:22]: slv_awmisc_info_p_tag [21:0]: slv_awmisc_info

6.6.16 RC_SLV_AWMISC3, 0x2601014C

Bits	Bit Name	Default	Type	Comment
31:0	rc_slv_awmisc127_96	32'h0	R/W	Reserved

6.6.17 RC_MSI_INFO0, 0x26010150

Bits	Bit Name	Default	Type	Comment
31:0	rc_msi_ctrl_io	32’h0	R	rc_msi_ctrl_io

6.6.18 RC_MSI_INFO1, 0x26010154

Bits	Bit Name	Default	Type	Comment
31:0	rc_msi_ctrl_int_vec	32’h0	R	rc_msi_ctrl_int_vec

6.6.19 VMI_31_0, 0x26010158

Ideally, we should write this registers first and then VMI_63_32 so that {vmi_63_32,vmi_31_0} data gets into FIFO

Bits	Bit Name	Default	Type	Comment
31:0	vmi_31_0	32’h0	R/W	vmi_31_0 bits

6.6.20 VMI_63_32, 0x2601015C

Writing into this register will also give pulse output which is used for writing {vmi_63_32, vmi_31_0} into Async FIFO

Bits	Bit Name	Default	Type	Comment
31:15	vmi_63_47	17’h0	R/W	vmi_63_47 bits
14	vmi_gnt	1’b0	R	should be connected to vmi_gnt
13:0	vmi_45_32	14’h0	R/W	vmi_45_32 bits

6.6.21 VMI_DATA_31_0, 0x26010160

Bits	Bit Name	Default	Type	Comment
31:0	vmi_data_31_0	32’h0	R/W	vmi_data_31_0 vmi data will be writtien to the FIFO when vmi_data_63_32 also written

6.6.22 VMI_DATA_63_32, 0x26010164

Writing into this register will also give pulse output which is used for writing {vmi_data_63_32, vmi_data_31_0} into async FIFO.

Bits	Bit Name	Default	Type	Comment
31:0	vmi_data_63_32	32’h0	R/W	vmi_data_63_32

6.6.23 RC_MAC_STATUS_31_0, 0x26010170

Bits	Bit Name	Default	Type	Comment
31:0	rc_mac_status_31_0	32'h01000002	R	[31]: radm_pme_or_err_rc_muxdclk ??? [30]: radm_correctable_err_muxdclk [29]: radm_nonfatal_err_muxdclk [28]: radm_fatal_err_muxdclk [27]: radm_pm_pme_muxdclk [26]: cfg_sys_err_rc_muxdclk [25]: smlh_req_rst_not [24]: link_req_rst_not [23]: cfg_hp_slot_ctrl_access [22]: cfg_dll_state_chged_en [21]: cfg_cmd_cpled_int_en [20]: cfg_hp_int_en [19]: cfg_pre_det_chged_en [18]: cfg_mrl_sensor_chged_en [17]: cfg_pwr_fault_det_en [16]: cfg_atten_button_pressed_en [15]: surprise_down_err [14]: iip_pm_linkst_in_l2 [13]: iip_pm_linkst_in_l1 [12]: iip_pm_linkst_in_l0s [11:9]: iip_pm_curnt_state[2:0] [8:3]: rc_smlh_ltssm_state[5:0] [2]: rc_smlh_link_up [1]: rc_mstr_aw_empty_muxdclk [0]: Reserved, always read 'b0

6.6.24 RC_MAC_STATUS_63_32, 0x26010174

Bits	Bit Name	Default	Type	Comment
31:0	rc_mac_status_63_32	32'h00000010	R	[31:28]: cfg_cxl_mbar0_uncorr_err[3:0] [27:21]: cfg_cxl_mbar0_corr_err [20:16]: cfg_aer_int_msg_num [15]: cfg_pme_int [14]: cfg_pme_msi [13]: iip_cfg_send_cor_err [12]: iip_cfg_send_nf_err [11]: iip_cfg_send_f_err [10]: cfg_2nd_reset [9:4]: cfg_neg_link_width [3]: cfg_aer_rc_err_msi [2]: cfg_aer_rc_err_int [1]: radm_pme_or_err_rc_muxdclk ??? [0]: hot_reset_req_muxdclk

6.6.25 RC_MAC_STATUS_95_64, 0x26010178

Bits	Bit Name	Default	Type	Comment
31:0	rc_mac_status_95_64	32'h0	R	[31:13]: cfg_cxl_pl_reg_fatal_err[18:0] ??? [12:0]: cfg_cxl_mbar0_uncorr_err[16:4] ???

6.6.26 RC_MAC_STATUS_127_96, 0x2601017C

Bits	Bit Name	Default	Type	Comment
31:0	rc_mac_status_127_96	32'h0	R	[31:13]: Reserved, always read 'b0 ??? [12:0]: cfg_cxl_pl_reg_fatal_err[31:19] ???

6.6.27 RC_MAC_CFG, 0x26010180

Bits	Bit Name	Default	Type	Comment
31:27	Reserved	5'b0	R/W	Reserved
26:17	hdm_locked	10'b0	R/W	
16:8	pclk_state_rate	9'h20	R/W	
7:5	cxl_mem_l1_entry_latency	3'b011	R/W	8us Idle time will trigger L1 entry on L1 entry_en asserted below, when traffic is not present
4	cxl_mem_l1_entry_en	1'b0	R/W	ASPM entry enable, for mem interface. Like CXL.io (PCIe)
3	cfg_poison_on_dec_error	1'b0	R/W	asserting poisons the data when decoding error
2	cfg_hdm_dec_en	1'b0	R/W	HDM decoders gets into action
1	cxl_pm_init_cmpl	1'b0	R/W	CXL PM initialization is completed (Exchange of CXL PM VDMs)
0	sys_aux_pwr_det	1'b1	R/W	MAC configuration

6.6.28 RC_MAC_CTRL, 0x26010184

Bits	Bit Name	Default	Type	Comment
31:9	Reserved	23'b0	R	Reserved, always read 'b0
8	vdm1_fifo_wr_en	1'b0	R/W	
7	vdm0_fifo_wr_en	1'b0	R/W	
6	hot_reset_req_ack	1'b0	R/W	
5	radm_pme_or_err_ack	1'b0	R/W	
4:2	Reserved	3'b0	R	Reserved, always read 'b0
1	rc_mstr_aw_pop	1'b0	R/W	
0	apps_pm_xmt_turnoff	1'b0	R/W	

6.6.29 RC_MAC_MSTR_AWADDR31_0, 0x26010190

Bits	Bit Name	Default	Type	Comment
31:0	rc_mac_mstr_awaddr31_0	32'h0	R	mstr_awaddr[31:0]

6.6.30 RC_MAC_MSTR_AWADDR51_32, 0x26010194

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	N/A	N/A	Reserved
19:0	rc_mac_mstr_awaddr51_32	20'h0	R	{4h'0, mstr_awaddr[47:32]}

6.6.31 AHB_SRAM_BASE_ADDRESS, 0x26010198

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R/W	Reserved

Bits	Bit Name	Default	Type	Comment
19:0	sram_base_address	20’h0	R/W	sram_base_address

6.6.32 INTERRUPT_STATUS_31_0, 0x260101A0

Bits	Bit Name	Default	Type	Comment
31:0	int_sts_reg_31_0	32’h18000000	R	31: cfg_bw_mgt_msi_lat 30: cfg_link_auto_bw_msi_lat 29: iip_rdlh_link_up_int 28: vdm_1_fifo_empty 27: vdm_0_fifo_empty 26: radm_qoverflow_lat 25: radm_pm_to_ack_lat 24: radm_pm_pme_lat 23: cfg_sys_err_rc_muxdclk 22: radm_cpl_timeout_lat 21: trgt_timeout_cpl_lat 20: hp_msi_lat 19: hp_int_lat 18: hp_pme_lat 17: rc_msi_ctrl_int 16: radm_nonfatal_err_lat 15: radm_fatal_err_lat 14: radm_correctable_err_lat 13: rc_radm_intd_deassert_lat 12: rc_radm_intc_deassert_lat 11: rc_radm_intb_deassert_lat 10: rc_radm_inta_deassert_lat 9: rc_radm_intd_assert_lat 8: rc_radm_intc_assert_lat 7: rc_radm_intb_assert_lat 6: rc_radm_inta_assert_lat 5: cfg_surprise_down_err_lat 4: cfg_send_nf_err_pulse_lat 3: cfg_send_f_err_pulse_lat 2: cfg_send_cor_err_pulse_lat 1: cfg_aer_rc_err_msi_lat 0: cfg_aer_rc_err_int

6.6.33 INTERRUPT_STATUS_63_32, 0x260101A4

Bits	Bit Name	Default	Type	Comment
31:0	int_sts_reg_ 63_32	32’h0	R	[31:3]: Reserved, always read ’b0 2: ~rc_mstr_aw_empty_muxdclk 1: cfg_link_eq_req_int_lat 0: link_req_rst_not_lat

6.6.34 INTERRUPT_CONTROL_1, 0x260101A8

Bits	Bit Name	Default	Type	Comment
31:0	interrupt_control_ 1	32'h0	R/W	31: icr_31 30: icr_30 29: rdlh_link_up_done [28:27]: Reserved, always read 'b0 26: radm_qoverflow 25: radm_pm_to_ack 24: radm_pm_pme 23: sys_err_rc 22: radm_cpl_timeout 21: trgt_cpl_timeout 20: hp_msi 19: hp_int 18: hp_pme 17: rc_msi_ctrl_int 16: radm_nonfatal_err 15: radm_fatal_err 14: radm_correctable_err 13: radm_intd_deasserted 12: radm_intc_deasserted 11: radm_intb_deasserted 10: radm_inta_deasserted 9: radm_intd_asserted 8: radm_intc_asserted 7: radm_intb_asserted 6: radm_inta_asserted 5: surprise_down_err 4: cfg_send_nf_err 3: cfg_send_f_err 2: cfg_send_cor_err 1: cfg_aer_rc_err_msi 0: cfg_aer_rc_err_int

6.6.35 INTERRUPT_CONTROL_2, 0x260101AC

Bits	Bit Name	Default	Type	Comment
31:0	interrupt_control_ 2	32'h0	R/W	[31:1]: ICR_2_[31:1] [0]: link_req_rst_not_ack

6.6.36 INTERRUPT_MASK_31_0, 0x260101B0

Bits	Bit Name	Default	Type	Comment
31:0	interrupt_mask_ 1	32'h0	R/W	int_mask_reg_31_0

6.6.37 INTERRUPT_MASK_63_32, 0x260101B4

Bits	Bit Name	Default	Type	Comment
31:0	interrupt_mask_ 2	32'h0	R/W	int_mask_reg_63_32

6.6.38 BUS_DEVICE_NUM_OTHERS, 0x260101B8

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18'b0	R	Reserved, always read 'b0

Bits	Bit Name	Default	Type	Comment
13	app_unlock_msg	1’b0	R/W	
12:8	dev_num	5’b0	R/W	
7:0	bus_num	8’b0	R/W	

6.6.39 DEVICE_PM_STATE, 0x260101BC

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12’b0	R/W	Reserved
19:17	rbr_pm_curnt_state	3’b0	R	
16	pm_linkst_in_l0s	1’b0	R	
15	pm_linkst_in_l1	1’b0	R	
14	pm_linkst_in_l2	1’b0	R	
13:9	pm_slave_state	5’b0	R	
8:4	pm_master_state	5’b0	R	
3	pme_en	1’b0	R	
2:0	pm_dstate	3’b100	R	Power state 3’b000 = D0active 3’b001 = D1 3’b010 = D2 3’b011 = D3 3’b100 = D0uninit (default)

6.6.40 TRGT_TIMEOUT_PROP, 0x260101C0

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	2’b0	R	Reserved, always read ’b0
29:27	trgt_timeout_cpl_func_num	3’b0	R	
26:24	trgt_timeout_cpl_tc	3’b0	R	
23:22	trgt_timeout_cpl_attr	2’b0	R	
21:10	trgt_timeout_cpl_len	12’b0	R	
9:0	trgt_timeout_lookup_id	10’b0	R	

6.6.41 RADM_TIMEOUT_PROP, 0x260101C4

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	2’b0	R	Reserved, always read ’b0
29:27	radm_timeout_cpl_func_num	3’b0	R	
26:24	radm_timeout_cpl_tc	3’b0	R	
23:22	radm_timeout_cpl_attr	2’b0	R	
21:10	radm_timeout_cpl_len	12’b0	R	
9:0	radm_timeout_cpl_tag	10’b0	R	

6.6.42 CXPL_DEBUG_INFO_63_32, 0x260101C8

Bits	Bit Name	Default	Type	Comment
31:0	xpl_debug_info_63_32	32’h08200000	R	31: smlh_scrambler_disable 30: smlh_link_disable 29: smlh_link_in_training 28: smlh_ltssm_in_pollconfig 27: smlh_training_rst_n [26:23]: 4’b0 22: mac_phy_txdetectrx_loopback 21: info_txelec_txburst 20: info_txcomp_rxburst 19: app_init_rst [18:16]: 3’b0 [15:8]: info_link_number [7:6]: 2’b0 5: xmtbyte_skip_sent 4: smlh_link_up 3: info_inskip_rcv 2: info_ts1_rcvd 1: info_ts2_rcvd 0: info_rcvd_lane_rev

6.6.43 CXPL_DEBUG_INFO_31_0, 0x260101CC

Bits	Bit Name	Default	Type	Comment
31:0	xpl_debug_info_ 31_0	32’b0	R	[31:28]: info_ts_link_ctrl[3:0] 27: info_ts_lane_num_is_k237 26: info_ts_link_num_is_k237 25: rplh_rcvd_idle[0] 24: rplh_rcvd_idle[1] [23:8]: info_tx_data_symbol [7: 6]: info_tx_datak_ctrl [5: 0]: smlh_ltssm_state

6.6.44 CXPL_DEBUG_INFO_EI_OTHER, 0x260101D0

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12’b0	R	Reserved, always read ’b0
19	radm_xfer_pending	1’b0	R	
18	radm_q_not_empty	1’b0	R	
17	brdg_dbi_xfer_pending	1’b0	R	
16	brdg_slv_xfer_pending	1’b0	R	
15:0	cxpl_debug_info_ei	16’h0020	R	

6.6.45 VDM_FIFO0_31_0, 0x260101D4

Bits	Bit Name	Default	Type	Comment
31:0	vdm_0_31_0	32’b0	R	Vdm0[31:0]

6.6.46 VDM_FIFO0_63_32, 0x260101D8

Bits	Bit Name	Default	Type	Comment
31:0	vdm_0_63_32	32'b0	R	Vdm0[63:32]

6.6.47 VDM_FIFO0_79_64, 0x260101DC

Bits	Bit Name	Default	Type	Comment
31:17	Reserved	15'b0	R	Reserved, always read 'b0
16	vdm_0_fifo_empty	1'b1	R	Vdm0 FIFO empty
15:0	vdm_0_79_64	16'b0	R	Vdm0[79:64]

6.6.48 VDM_FIFO1_31_0, 0x260101E0

Bits	Bit Name	Default	Type	Comment
31:0	vdm_1_31_0	32'b0	R	Vdm1[31:0]

6.6.49 VDM_FIFO1_63_32, 0x260101E4

Bits	Bit Name	Default	Type	Comment
31:0	vdm_1_63_32	32'b0	R	Vdm1[63:32]

6.6.50 VDM_FIFO1_79_64, 0x260101E8

Bits	Bit Name	Default	Type	Comment
31:17	Reserved	15'b0	R	Reserved, always read 'b0
16	vdm_1_fifo_empty	1'b1	R	Vdm1 FIFO empty
15:0	vdm_1_79_64	16'b0	R	Vdm1[79:64]

6.6.51 SPARE_REG_RW, 0x26010200

Bits	Bit Name	Default	Type	Comment
31:0	spare_reg_rw	32'b0	R/W	Spare register

6.6.52 SPARE_REG_RO, 0x26010204

Bits	Bit Name	Default	Type	Comment
31:0	spare_reg_ro	32'b0	R	Reserved, always read 'b0

6.7 OTP Register

OTP CSR address = `0x2608_0000 ~ 0x2608_FFFF`

6.7.1 OFS_VERSION, 0x26083400

Bits	Bit Name	Default	Type	Comment
31:0	WRAPPER_VER	32'h20230613	R	Wrapper version

6.7.2 OFS_PART_NUM, 0x26083404

Bits	Bit Name	Default	Type	Comment
31:0	NEO_PART_NUM	32'h4547512D	R	Neo part number

6.7.3 OFS_INTRPT, 0x26083408

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14'b0	R	Reserved, always read 'b0
17:16	intrpt_en[1:0]	2'b00	R/W	Interrupt enable
15:2	Reserved	14'b0	R	Reserved, always read 'b0
1:0	intrpt_st[1:0] (R) intrpt_clr[1:0] (W)	2'b00	R/W1c	intrpt_st[1]: OTP read interrupt intrpt_st[0]: read outside memory interrupt Write 1'b1 = intrpt_clr

6.7.4 OFS_STATUS, 0x26083500

Bits	Bit Name	Default	Type	Comment
31:14	Reserved	18'b0	R	Reserved, always read 'b0
13	watch_dog_active	1'b0	R	Watch dog active
12	bist_mode	1'b0	R	BIST mode, write 0x2608_3508[31:24] = 8'hA5
11:8	bist_fail_flag[3:0]	4'b0	R	cp_flow_cnt in BIST mode
7:1	Reserved	7'b0	R	Reserved, always read 'b0
0	ctl_busy	1'b0	R	Control busy flag

6.7.5 OFS_DEEP, 0x26083504

Bits	Bit Name	Default	Type	Comment
31	bist_blank_check	1'b0	R/W	BIST blank check
30	bist_wtd_enable	1'b1	R/W	BIST watch dog enable
29:1	Reserved	29'b0	R	Reserved, always read 'b0
0	ctl_pdstb	1'b1	R/W	Power down strobe

6.7.6 OFS_CONFIG, 0x26083508

Bits	Bit Name	Default	Type	Comment
31:24	set_bist_mode	8’h0	W	Write 8'hA5 to set bist_mode(0x3500[12]) Always read 8’b0
23:11	Reserved	13’b0	R	Reserved, always read ‘b0
10:8	ctl_rdmd	3’b000	R/W	Read modes: 3’b001 = CTL_INIT_MG_RD 3’b010 = CTL_PGM_MG_RD 3’b011 = CTL_HT_INIT_MG_RD 3’b100 = CTL_HT_PGM_MG_RD 3’b101 = CTL_LT_PGM_MG_RD Others: = CTL_RD
7:3	Reserved	5’b0	R	Reserved, always read ‘b0
2	ctl_pgm_prt	1’b0	R/W	State machine control ctl_pgm_prt
1	ctl_pgm_ign	1’b0	R/W	State machine control ctl_pgm_ign
0	Reserved	1’b0	R	Reserved, always read ‘b0

6.8 SPI Control Register

SPI0 CSR address = `0x2000_0000 ~ 0x2000_0FFF`
SPI1 CSR address = `0x2000_1000 ~ 0x2000_1FFF`

6.8.1 SPI_CONFIG, 0x2000[1:0][F:0]00

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:12	cr_spi_deg_cnt	4’b0	R/W	De-glitch function cycle count.
11	cr_spi_deg_en	1’b0	R/W	Enable signal of de-glitch function.
10	cr_spi_s_3pin_mode	1’b0	R/W	SPI slave 3-pin mode 0: 4-pin mode (SS_n is enabled) 1: 3-pin mode (SS_n is disabled / don't care)
9	cr_spi_m_cont_en	1’b0	R/W	Enable signal of master continuous transfer mode 0: Disabled. SS_n will de-assert between each data frame. 1: Enabled. SS_n will stay asserted between each consecutive data frame if next data is valid in FIFO.
8	cr_spi_rxd_ignr_en	1’b0	R/W	Enable signal of RX data ignore function.
7	cr_spi_byte_inv	1’b0	R/W	Byte inverse signal for each FIFO entry. 0: Byte [0] is sent first. 1: Byte [3] is sent first.
6	cr_spi_bit_inv	1’b0	R/W	Bit inverse signal for each FIFO entry. 0: Each byte is sent MSB first. 1: Each byte [3] is sent LSB first.
5	cr_spi_sclk_ph	1’b0	R/W	SCLK clock phase inverse signal
4	cr_spi_sclk_pol	1’b0	R/W	SCLK polarity 0: SCLK output LOW at IDLE state 1: SCLK output HIGH at IDLE state
3:2	cr_spi_frame_size	2’b00	R/W	SPI frame size (also the valid width for each FIFO entry) 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit
1	cr_spi_s_en	1’b0	R/W	Enable signal of SPI slave function. Master and slave should not be both enabled at same time. This bit becomes don’t-care if “cr_spi_m_en” is enabled
0	cr_spi_m_en	1’b0	R/W	Enable signal of SPI Maser function. Asserting this bit will trigger the transaction and should be de-asserted after finish.

6.8.2 SPI_INT_STS, 0x2000[1:0][F:0]04

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	2’b0	R	Reserved, always read ‘b0
29	cr_spi_fer_en	1’b1	R/W	Interrupt enable of spi_fer_int
28	cr_spi_txu_en	1’b1	R/W	Interrupt enable of spi_txu_int
27	cr_spi_sto_en	1’b1	R/W	Interrupt enable of spi_sto_int
26	cr_spi_rxf_en	1’b1	R/W	Interrupt enable of spi_rxv_int
25	cr_spi_txf_en	1’b1	R/W	Interrupt enable of spi_txe_int
24	cr_spi_end_en	1’b1	R/W	Interrupt enable of spi_end_int
23:22	Reserved	2’b0	R	Reserved, always read ‘b0

Bits	Bit Name	Default	Type	Comment
21	Reserved	1’b0	R/W	Reserved
20	cr_spi_txu_clr	1’b0	W1c	Interrupt clear of spi_txu_int
19	cr_spi_sto_clr	1’b0	W1c	Interrupt clear of spi_sto_int
18:17	Reserved	2’b0	R/W	Reserved
16	cr_spi_end_clr	1’b0	W1c	Interrupt clear of spi_end_int
15:14	Reserved	2’b0	R	Reserved, always read ‘b0
13	cr_spi_fer_mask	1’b1	R/W	Interrupt mask of spi_fer_int
12	cr_spi_txu_mask	1’b1	R/W	Interrupt mask of spi_txu_int
11	cr_spi_sto_mask	1’b1	R/W	Interrupt mask of spi_sto_int
10	cr_spi_rxf_mask	1’b1	R/W	Interrupt mask of spi_rxv_int
9	cr_spi_txf_mask	1’b1	R/W	Interrupt mask of spi_txe_int
8	cr_spi_end_mask	1’b1	R/W	Interrupt mask of spi_end_int
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	spi_fer_int	1’b0	R	SPI TX/RX FIFO error interrupt. Auto cleared when FIFO overflow/underflow error flag is cleared.
4	spi_txu_int	1’b0	R	SPI slave mode TX underrun error flag. Triggered when TXD is not ready during transfer in slave mode.
3	spi_sto_int	1’b0	R	SPI slave mode transfer time-out interrupt. Triggered when SPI bus is idle for a given value
2	spi_rxf_int	1’b0	R	SPI RX FIFO ready (rx_fifo_cnt>rx_fifo_th) interrupt, auto cleared when data is popped.
1	spi_txf_int	1’b1	R	SPI TX FIFO ready (tx_fifo_cnt>tx_fifo_th) interrupt, auto cleared when data is pushed.
0	spi_end_int	1’b0	R	SPI transfer end interrupt. Shared by both master and slave mode.

6.8.3 SPI_BUS_BUSY, 0x2000[1:0][F:0]08

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31’b0	R	Reserved, always read ‘b0
0	sts_spi_bus_busy	1’b0	R	Indicator of SPI bus busy

6.8.4 SPI_PRD_0, 0x2000[1:0][F:0]10

Bits	Bit Name	Default	Type	Comment
31:24	cr_spi_prd_d_ph_1	8’h0F	R/W	Length of DATA phase 1 (unit: SPI source clock period)
23:16	cr_spi_prd_d_ph_0	8’h0F	R/W	Length of DATA phase 0 (unit: SPI source clock period))
15:8	cr_spi_prd_p	8’h0F	R/W	Length of STOP condition (unit: SPI source clock period)
7:0	cr_spi_prd_s	8’h0F	R/W	Length of START condition (unit: SPI source clock period)

6.8.5 SPI_PRD_1, 0x2000[1:0][F:0]14

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	cr_spi_prd_i	8’h0F	R/W	Length of INTERVAL between frame (unit: SPI source clock period)

6.8.6 SPI_RXD_IGNR, 0x2000[1:0][F:0]18

Bits	Bit Name	Default	Type	Comment
31:21	Reserved	11’b0	R	Reserved, always read ‘b0
20:16	cr_spi_rxd_ignr_s	5’b0	R/W	Starting point of RX data ignore function (unit: bit)
15:5	Reserved	11’b0	R	Reserved, always read ‘b0
4:0	cr_spi_rxd_ignr_p	5’b0	R/W	Stopping point of RX data ignore function (unit: bit)

6.8.7 SPI_STO_VALUE, 0x2000[1:0][F:0]1C

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	20’b0	R	Reserved, always read ‘b0
11:0	cr_spi_sto_value	12’hFFF	R/W	Time-out value for spi_sto_int triggering

6.8.8 SPI_FIFO_CONFIG_0, 0x2000[1:0][F:0]80

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7	rx_fifo_underflow	1’b0	R	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	1’b0	R	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	1’b0	R	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	1’b0	R	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	1’b0	W1c	Clear signal of RX FIFO
2	tx_fifo_clr	1’b0	W1c	Clear signal of TX FIFO
1	spi_dma_rx_en	1’b0	R/W	Enable signal of dma_rx_req/ack interface
0	spi_dma_tx_en	1’b0	R/W	Enable signal of dma_tx_req/ack interface

6.8.9 SPI_FIFO_CONFIG_1, 0x2000[1:0][F:0]84

Bits	Bit Name	Default	Type	Comment
31:26	Reserved	6’b0	R	Reserved, always read ‘b0
25:24	rx_fifo_th	2’b00	R/W	RX FIFO threshold, dma_rx_req will not be asserted if rx_fifo_cnt is less than this value
23:18	Reserved	6’b0	R	Reserved, always read ‘b0
17:16	tx_fifo_th	2’b00	R/W	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value

Bits	Bit Name	Default	Type	Comment
15:11	Reserved	5'b0	R	Reserved, always read 'b0
10:8	rx_fifo_cnt	3'b000	R	RX FIFO available count, means byte count of data received in RX FIFO (unit: byte)
7:3	Reserved	5'b0	R	Reserved, always read 'b0
2:0	tx_fifo_cnt	3'b100	R	TX FIFO available count, means empty space remained in TX FIFO (unit: byte)

6.8.10 SPI_FIFO_WDATA, 0x2000[1:0][F:0]88

Bits	Bit Name	Default	Type	Comment
31:0	spi_fifo_wdata	32'h0	W	SPI FIFO write data port

6.8.11 SPI_FIFO_RDATA, 0x2000[1:0][F:0]8C

Bits	Bit Name	Default	Type	Comment
31:0	spi_fifo_rdata	32'h0	R	SPI FIFO read data port

6.8.12 BACKUP_IO_EN, 0x2000[1:0][F:0]FC

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31'b0	R	Reserved, always read 'b0
0	backup_io_en	1'b0	R/W	Enable IO backup function

6.9 I2C Control Register

I2C CSR address = `0x2000_2000 ~ 0x2000_2FFF`

6.9.1 I2C_CONFIG, 0x20002000

Bits	Bit Name	Default	Type	Comment
31:28	cr_i2c_deg_cnt	4'b0	R/W	De-glitch function cycle count.
27:20	cr_i2c_pkt_len	8'h0	R/W	Packet length (unit: byte)
19:18	Reserved	2'b0	R	Reserved, always read 'b0
17:8	cr_i2c_slv_addr	10'h0	R/W	Slave address for I2C transaction (target address)
7	cr_i2c_10b_addr_en	1'b0	R/W	10 bit address enable
6:5	cr_i2c_sub_addr_bc	2'b00	R/W	Slave address field byte count 00: 1 byte 01: 2 byte 10: 3 byte 11: 4 byte
4	cr_i2c_sub_addr_en	1'b0	R/W	Enable signal of I2C sub-address field
3	cr_i2c_scl_sync_en	1'b1	R/W	Enable signal of I2C SCL synchronization. Should be enabled to support Multi-Master and Clock-Stretching (normally should be turned off)
2	cr_i2c_deg_en	1'b0	R/W	Enable signal of I2C de-glitch (for all input pins)
1	cr_i2c_pkt_dir	1'b1	R/W	Transfer direction of the packet. 0: write 1: read.
0	cr_i2c_m_en	1'b0	R/W	Enable signal of I2C Maser function. Asserting this bit will trigger the transaction and should be de-asserted after finish.

6.9.2 I2C_INT_STS, 0x20002004

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	2'b0	R	Reserved, always read 'b0
29	cr_i2c_fer_en	1'b1	R/W	Interrupt enable of i2c_fer_int
28	cr_i2c_arb_en	1'b1	R/W	Interrupt enable of i2c_arb_int
27	cr_i2c_nak_en	1'b1	R/W	Interrupt enable of i2c_nak_int
26	cr_i2c_rxf_en	1'b1	R/W	Interrupt enable of i2c_rxf_int
25	cr_i2c_txf_en	1'b1	R/W	Interrupt enable of i2c_txf_int
24	cr_i2c_end_en	1'b1	R/W	Interrupt enable of i2c_end_int
23:22	Reserved	2'b0	R	Reserved, always read 'b0
21	Reserved	1'b0	R/W	Reserved
20	cr_i2c_arb_clr	1'b0	W1c	Interrupt clear of i2c_arb_int
19	cr_i2c_nak_clr	1'b0	W1c	Interrupt clear of i2c_nak_int
18:17	Reserved	2'b0	R/W	Reserved
16	cr_i2c_end_clr	1'b0	W1c	Interrupt clear of i2c_end_int
15:14	Reserved	2'b0	R	Reserved, always read 'b0

Bits	Bit Name	Default	Type	Comment
13	cr_i2c_fer_mask	1’b1	R/W	Interrupt mask of i2c_fer_int
12	cr_i2c_arb_mask	1’b1	R/W	Interrupt mask of i2c_arb_int
11	cr_i2c_nak_mask	1’b1	R/W	Interrupt mask of i2c_nak_int
10	cr_i2c_rxf_mask	1’b1	R/W	Interrupt mask of i2c_rxf_int
9	cr_i2c_txf_mask	1’b1	R/W	Interrupt mask of i2c_txf_int
8	cr_i2c_end_mask	1’b1	R/W	Interrupt mask of i2c_end_int
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	i2c_fer_int	1’b0	R	I2C TX/RX FIFO error interrupt. Auto-cleared when FIFO overflow/underflow error flag is cleared.
4	i2c_arb_int	1’b0	R	I2C arbitration lost interrupt
3	i2c_nak_int	1’b0	R	I2C NAK received interrupt
2	i2c_rxf_int	1’b0	R	I2C RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto cleared when data is popped
1	i2c_txf_int	1’b1	R	I2C TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto cleared when data is pushed
0	i2c_end_int	1’b0	R	I2C transfer end interrupt.

6.9.3 I2C_SUB_ADDR, 0x20002008

Bits	Bit Name	Default	Type	Comment
31:24	cr_i2c_sub_addr_b3	8’h0	R/W	Sub address field byte 3
23:16	cr_i2c_sub_addr_b2	8’h0	R/W	Sub address field byte 2
15:8	cr_i2c_sub_addr_b1	8’h0	R/W	Sub address field byte 1
7:0	cr_i2c_sub_addr_b0	8’h0	R/W	Sub address field byte 0

6.9.4 I2C_BUS_BUSY, 0x2000200C

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0
1	cr_i2c_bus_busy_clr	1’b0	W1c	Clear signal of bus_busy status, not for normal usage (in case I2C bus hangs.)
0	sts_i2c_bus_busy	1’b0	R	I2C bus busy

6.9.5 I2C_PRD_START, 0x20002010

Bits	Bit Name	Default	Type	Comment
31:24	cr_i2c_prd_s_ph_3	8’h0F	R/W	Length of START condition phase 3
23:16	cr_i2c_prd_s_ph_2	8’h0F	R/W	Length of START condition phase 2
15:8	cr_i2c_prd_s_ph_1	8’h0F	R/W	Length of START condition phase 1
7:0	cr_i2c_prd_s_ph_0	8’h0F	R/W	Length of START condition phase 0

6.9.6 I2C_PRD_STOP, 0x20002014

Bits	Bit Name	Default	Type	Comment
31:24	cr_i2c_prd_p_ph_3	8'h0F	R/W	Length of STOP condition phase 3
23:16	cr_i2c_prd_p_ph_2	8'h0F	R/W	Length of STOP condition phase 2
15:8	cr_i2c_prd_p_ph_1	8'h0F	R/W	Length of STOP condition phase 1
7:0	cr_i2c_prd_p_ph_0	8'h0F	R/W	Length of STOP condition phase 0

6.9.7 I2C_PRD_DATA, 0x20002018

Bits	Bit Name	Default	Type	Comment
31:24	cr_i2c_prd_d_ph_3	8'h0F	R/W	Length of DATA phase 3
23:16	cr_i2c_prd_d_ph_2	8'h0F	R/W	Length of DATA phase 2
15:8	cr_i2c_prd_d_ph_1	8'h0F	R/W	Length of DATA phase 1
7:0	cr_i2c_prd_d_ph_0	8'h0F	R/W	Length of DATA phase 0

I2C_FIFO_CONFIG_0, 0x20002080

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7	rx_fifo_underflow	1'b0	R	Underflow flag of RX FIFO, cleared by rx_fifo_clr
6	rx_fifo_overflow	1'b0	R	Overflow flag of RX FIFO, cleared by rx_fifo_clr
5	tx_fifo_underflow	1'b0	R	Underflow flag of TX FIFO, cleared by tx_fifo_clr
4	tx_fifo_overflow	1'b0	R	Overflow flag of TX FIFO, cleared by tx_fifo_clr
3	rx_fifo_clr	1'b0	W1c	Clear signal of RX FIFO
2	tx_fifo_clr	1'b0	W1c	Clear signal of TX FIFO
1	i2c_dma_rx_en	1'b0	R/W	Enable signal of dma_rx_req/ack interface
0	i2c_dma_tx_en	1'b0	R/W	Enable signal of dma_tx_req/ack interface

6.9.8 I2C_FIFO_CONFIG_1, 0x20002084

Bits	Bit Name	Default	Type	Comment
31:25	Reserved	7'b0	R	Reserved, always read 'b0
24	rx_fifo_th	1'b0	R/W	RX FIFO threshold, dma_rx_req will not assert if rx_fifo_cnt is less than this value
23:17	Reserved	7'b0	R	Reserved, always read 'b0
16	tx_fifo_th	1'b0	R/W	TX FIFO threshold, dma_tx_req will not assert if tx_fifo_cnt is less than this value
15:10	Reserved	6'b0	R	Reserved, always read 'b0
9:8	rx_fifo_cnt	2'b00	R	RX FIFO available count
7:2	Reserved	6'b0	R	Reserved, always read 'b0
1:0	tx_fifo_cnt	2'b10	R	TX FIFO available count

6.9.9 I2C_FIFO_WDATA, 0x20002088

Bits	Bit Name	Default	Type	Comment
31:0	i2c_fifo_wdata	32’h0	W	I2C FIFO write data

6.9.10 I2C_FIFO_RDATA, 0x2000208C

Bits	Bit Name	Default	Type	Comment
31:0	i2c_fifo_rdata	32’h0	R	I2C FIFO read data

6.10 Local Timer Register

Local timer CSR address = `0x2000_3000 ~ 0x2000_3FFF`

6.10.1 TCCR, 0x20003000

Bits	Bit Name	Default	Type	Comment
31:10	Reserved	22'b0	R	Reserved, always read 'b0
9:8	cs_wdt	2'b00	R/W	Clock source for timer WDT 00: fclk 01: clk_32k7 10: clk_32k7 11: PLL 25 MHz
7	Reserved	1'b0	R	Reserved, always read 'b0
6:5	cs_3	2'b00	R/W	Clock source for timer #3 00: fclk 01: clk_32k7 10: clk_32k7 11: PLL 25 MHz
4	Reserved	1'b0	R	Reserved, always read 'b0
3:2	cs_2	2'b00	R/W	Clock source for timer #2 00: fclk 01: clk_32k7 10: clk_32k7 11: PLL 25 MHz
1:0	Reserved	2'b0	R	Reserved, always read 'b0

6.10.2 TMR2_0, 0x20003010

Bits	Bit Name	Default	Type	Comment
31:0	tmr_2_0	32'hFFFFFFFF	R/W	Timer2 match register 0

6.10.3 TMR2_1, 0x20003014

Bits	Bit Name	Default	Type	Comment
31:0	tmr_2_1	32'hFFFFFFFF	R/W	Timer2 match register 1

6.10.4 TMR2_2, 0x20003018

Bits	Bit Name	Default	Type	Comment
31:0	tmr_2_2	32'hFFFFFFFF	R/W	Timer2 match register 2

6.10.5 TMR3_0, 0x2000301C

Bits	Bit Name	Default	Type	Comment
31:0	tmr_3_0	32'hFFFFFFFF	R/W	Timer3 match register 0

6.10.6 TMR3_1, 0x20003020

Bits	Bit Name	Default	Type	Comment
31:0	tmr_3_1	32’hFFFFFFFF	R/W	Timer3 match register 1

6.10.7 TMR3_2, 0x20003024

Bits	Bit Name	Default	Type	Comment
31:0	tmr_3_2	32’hFFFFFFFF	R/W	Timer3 match register 2

6.10.8 TCR2, 0x2000302C

Bits	Bit Name	Default	Type	Comment
31:0	tcr2_counter	32’h0	R	Timer2 counter register

6.10.9 TCR3, 0x20003030

Bits	Bit Name	Default	Type	Comment
31:0	tcr3_counter	32’h0	R	Timer3 counter register

6.10.10 TMSR2, 0x20003038

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tmsr2_2	1’b0	R	Timer2 match register 2 status. Clear interrupt would also clear this bit.
1	tmsr2_1	1’b0	R	Timer2 match register 1 status. Clear interrupt would also clear this bit.
0	tmsr2_0	1’b0	R	Timer2 match register 0 status. Clear interrupt would also clear this bit.

6.10.11 TMSR3, 0x2000303C

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tmsr3_2	1’b0	R	Timer3 match register 2 status. Clear interrupt would also clear this bit.
1	tmsr3_2	1’b0	R	Timer3 match register 1 status. Clear interrupt would also clear this bit.
0	tmsr3_2	1’b0	R	Timer3 match register 0 status. Clear interrupt would also clear this bit.

6.10.12 TIER2, 0x20003044

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tier2_2	1’b0	R/W	Timer2 match value 2 interrupt enable.
1	tier2_1	1’b0	R/W	Timer2 match value 1 interrupt enable.

Bits	Bit Name	Default	Type	Comment
0	tier2_0	1’b0	R/W	Timer2 match value 0 interrupt enable.

6.10.13 TIER3, 0x20003048

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tier3_2	1’b0	R/W	Timer3 match value 2 interrupt enable.
1	tier3_1	1’b0	R/W	Timer3 match value 1 interrupt enable.
0	tier3_0	1’b0	R/W	Timer3 match value 0 interrupt enable.

6.10.14 TPLVR2, 0x20003050

Bits	Bit Name	Default	Type	Comment
31:0	tplvr2	32’b0	R/W	Timer2 Pre-Load Value

6.10.15 TPLVR3, 0x20003054

Bits	Bit Name	Default	Type	Comment
31:0	tplvr3	32’b0	R/W	Timer3 Pre-Load Value

6.10.16 TPLCR2, 0x2000305C

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0
1:0	tplcr2	2’b0	R/W	Timer2 pre-load control 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

6.10.17 TPLCR3, 0x20003060

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0
1:0	tplcr3	2’b0	R/W	Timer3 pre-load control 2'd0 - No pre-load 2'd1 - Pre-load with match comparator 0 2'd2 - Pre-load with match comparator 1 2'd3 - Pre-load with match comparator 2

6.10.18 WMER, 0x20003064

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0

Bits	Bit Name	Default	Type	Comment
1	wrie	1’b0	R/W	WDT reset/interrupt mode 1'b0 - WDT expiration to generate interrupt 1'b1 - WDT expiration to generate reset source
0	we	1’b0	R/W	WDT enable register

6.10.19 WMR, 0x20003068

Bits	Bit Name	Default	Type	Comment
31:17	Reserved	15’b0	R	Reserved, always read ‘b0
16	wdt_align	1’b0	R/W	WDT compare value update align interrupt
15:0	wmr	16’hFFFF	R/W	WDT counter match value

6.10.20 WVR, 0x2000306C

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:0	wdt_cnt	16’b0	R	WDT counter value

6.10.21 WSR, 0x20003070

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31’b0	R	Reserved, always read ‘b0
0	wts	1’b0	R/W	WDT reset status Write 0 to clear the WDT reset status Read 1 indicates reset was caused by the WDT

6.10.22 TICR2, 0x20003078

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tclr2_2	1’b0	R/W	Timer2 Interrupt clear for match comparator 2
1	tclr2_1	1’b0	R/W	Timer2 Interrupt clear for match comparator 1
0	tclr2_0	1’b0	R/W	Timer2 Interrupt clear for match comparator 0

6.10.23 TICR3, 0x2000307C

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tclr3_2	1’b0	R/W	Timer3 Interrupt clear for match comparator 2
1	tclr3_1	1’b0	R/W	Timer3 Interrupt clear for match comparator 1
0	tclr3_0	1’b0	R/W	Timer3 Interrupt clear for match comparator 0

6.10.24 WICR, 0x20003080

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31’b0	R	Reserved, always read ‘b0
0	wiclr	1’b0	R/W	WDT Interrupt Clear

6.10.25 TCER, 0x20003084

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25’b0	R	Reserved, always read ‘b0
6	tcr3_cnt_clr	1’b0	R/W	Timer3 count clear
5	tcr2_cnt_clr	1’b0	R/W	Timer2 count clear
4:3	Reserved	2’b0	R	Reserved, always read ‘b0
2	timer3_en	1’b0	R/W	Timer3 count enable
1	timer2_en	1’b0	R/W	Timer2 count enable
0	Reserved	1’b0	R	Reserved, always read ‘b0

6.10.26 TCMR, 0x20003088

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25’b0	R	Reserved, always read ‘b0
6	timer3_align	1’b0	R/W	Timer3 compare value update align interrupt
5	timer2_align	1’b0	R/W	Timer2 compare value update align interrupt
4:3	Reserved	2’b0	R	Reserved, always read ‘b0
2	timer3_mode	1’b0	R/W	0:pre-load mode; 1:free run mode
1	timer2_mode	1’b0	R/W	0:pre-load mode; 1:free run mode
0	Reserved	1’b0	R	Reserved, always read ‘b0

6.10.27 TILR2, 0x20003090

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tilr2_2	1’b0	R/W	0:level; 1:edge
1	tilr2_1	1’b0	R/W	0:level; 1:edge
0	tilr2_0	1’b0	R/W	0:level; 1:edge

6.10.28 TILR3, 0x20003094

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	tilr3_2	1’b0	R/W	0:level; 1:edge

Bits	Bit Name	Default	Type	Comment
1	tilr3_1	1’b0	R/W	0:level; 1:edge
0	tilr3_0	1’b0	R/W	0:level; 1:edge

6.10.29 WCR, 0x20003098

Note: Write only register, always read ‘b0

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31’b0	R	Reserved, always read ‘b0
0	wcr	1’b0	WO	WDT Counter Reset

6.10.30 WFAR, 0x2000309C

Note: Write only register, always read ‘b0

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:0	wcr	16’b0	WO	WDT access key1 - 16'hBABA

6.10.31 WSAR, 0x200030A0

Note: Write only register, always read ‘b0

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:0	wsar	16’b0	WO	WDT access key2 - 16'hEB10

6.10.32 TCVWR2, 0x200030A8

Bits	Bit Name	Default	Type	Comment
31:0	tcr2_cnt_lat	32’b0	R	Timer2 Counter Latch Value

6.10.33 TCVWR3, 0x200030AC

Bits	Bit Name	Default	Type	Comment
31:0	tcr3_cnt_lat	32’b0	R	Timer3 Counter Latch Value

6.10.34 TCVSTN2, 0x200030B4

Bits	Bit Name	Default	Type	Comment
31:0	tcr2_cnt_sync	32’b0	R	Timer2 Counter Sync Value (continue readable)

6.10.35 TCVSTN3, 0x200030B8

Bits	Bit Name	Default	Type	Comment
31:0	tcr3_cnt_sync	32'b0	R	Timer3 Counter Sync Value (continue readable)

6.10.36 TCDR, 0x200030BC

Bits	Bit Name	Default	Type	Comment
31:24	wcdr	8'b0	R/W	WDT clock division value register
23:16	tcdr3	8'b0	R/W	Timer3 clock division value register
15:8	tcdr2	8'b0	R/W	Timer2 clock division value register
7:0	Reserved	8'b0	R	Reserved, always read 'b0

6.11 UART Control Register

UART CSR address = 0x2000_4000 ~ 0x2000_4FFF

6.11.1 UTX_CONFIG, 0x20004000

Bits	Bit Name	Default	Type	Comment
31:16	cr_utx_len	16'b0	R/W	Length of UART TX data transfer (Unit: character/byte) (Don't-care if cr_utx_frm_en is enabled)
15:13	cr_utx_bit_cnt_b	3'b100	R/W	UART TX BREAK bit count (for LIN protocol) Note: Additional 8 bit times will be added since LIN Break field requires at least 13 bit times
12:11	cr_utx_bit_cnt_p	2'b01	R/W	UART TX STOP bit count (unit: 0.5 bit)
10:8	cr_utx_bit_cnt_d	3'b111	R/W	TX data bit count for each character
7	cr_utx_ir_inv	1'b0	R/W	Inverse signal of TX output in IR mode
6	cr_utx_ir_en	1'b0	R/W	Enable signal of TX IR mode
5	cr_utx_prt_sel	1'b0	R/W	Select signal of TX parity bit 1: Odd parity 0: Even parity
4	cr_utx_prt_en	1'b0	R/W	Enable signal of TX parity bit
3	cr_utx_lin_en	1'b0	R/W	Enable signal of TX LIN mode (LIN header will be sent before sending data)
2	cr_utx_frm_en	1'b0	R/W	Enable signal of TX free run mode (utx_end_int will be disabled)
1	cr_utx_cts_en	1'b0	R/W	Enable signal of TX CTS flow control function.
0	cr_utx_en	1'b0	R/W	Enable signal of TX function

6.11.2 URX_CONFIG, 0x20004004

Bits	Bit Name	Default	Type	Comment
31:16	cr_urx_len	16'b0	R/W	Length of RX data (unit: character/byte). Urx_end_int will assert when this length is reached
15:12	cr_urx_deg_cnt	4'b0	R/W	De-glitch function cycle count
11	cr_urx_deg_en	1'b0	R/W	Enable signal of RXD input de-glitch function
10:8	cr_urx_bit_cnt_d	3'b111	R/W	RX data bit count for each character
7	cr_urx_ir_inv	1'b0	R/W	Inverse signal of RX output in IR mode

Bits	Bit Name	Default	Type	Comment
6	cr_urx_ir_en	1’b0	R/W	Enable signal of RX IR mode
5	cr_urx_prt_sel	1’b0	R/W	Select signal of RX parity bit 1: Odd parity 0: Even parity
4	cr_urx_prt_en	1’b0	R/W	Enable signal of RX parity bit
3	cr_urx_lin_en	1’b0	R/W	Enable signal of RX LIN mode (LIN header will be required and checked before receiving data)
2	Reserved	1’b0	R	Reserved, always read ‘b0
1	cr_urx_abr_en	1’b0	R/W	Enable signal of RX auto baud rate detection function.
0	cr_urx_en	1’b0	R/W	Enable signal of RX function

6.11.3 UART_BIT_PRD, 0x20004008

Bits	Bit Name	Default	Type	Comment
31:16	cr_urx_bit_prd	16’h00FF	R/W	Period of each UART RX bit, related to baud rate.
15:0	cr_utx_bit_prd	16’h00FF	R/W	Period of each UART TX bit, related to baud rate.

6.11.4 DATA_CONFIG, 0x2000400C

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	31’b0	R	Reserved, always read ‘b0
0	cr_uart_bit_inv	1’b0	R/W	Bit inverse signal for each data byte 0: Each byte is sent LSB-first 1: Each byte is sent MSB-first

6.11.5 UTX_IR_POSITION, 0x20004010

Bits	Bit Name	Default	Type	Comment
31:16	cr_utx_ir_pos_p	16’h009F	R/W	Stop position of UART TX IR pulse.
15:0	cr_utx_ir_pos_s	16’h0070	R/W	Start position of UART TX IR pulse.

6.11.6 URX_IR_POSITION, 0x20004014

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:0	cr_urx_ir_pos_s	16’h6F	R/W	Start position of UART RXD pulse recovered from IR signal.

6.11.7 URX_RTO_TIMER, 0x20004018

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	cr_urx_rto_value	8’hF	R/W	Time-out value for triggering RTO interrupt (unit: bit time)

6.11.8 UART_SW_MODE, 0x2000401C

Bits	Bit Name	Default	Type	Comment
31:4	Reserved	28'b0	R	Reserved, always read 'b0
3	cr_urx_rts_sw_val	1'b0	R/W	UART RX RTS output SW control value
2	cr_urx_rts_sw_mode	1'b0	R/W	UART RX RTS output SW control mode
1	cr_utx_txd_sw_val	1'b0	R/W	UART TX TXD output SW control value
0	cr_utx_txd_sw_mode	1'b0	R/W	UART TX TXD output SW control mode

6.11.9 UART_INT_STS, 0x20004020

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	20'b0	R	Reserved, always read 'b0
11	urx_ad5_int	1'b0	R	UART RX ABR Detection finish interrupt using codeword 0x55
10	urx_ads_int	1'b0	R	UART RX ABR Detection finish interrupt using START bit
9	urx_bcr_int	1'b0	R	UART RX byte count reached interrupt
8	urx_lse_int	1'b0	R	UART RX LIN mode sync field error interrupt
7	urx_fer_int	1'b0	R	UART RX FIFO error interrupt. Auto cleared when FIFO overflow/underflow error flag is cleared.
6	utx_fer_int	1'b0	R	UART TX FIFO error interrupt. Auto cleared when FIFO overflow/underflow error flag is cleared.
5	urx_pce_int	1'b0	R	UART RX parity check error interrupt.
4	urx_rto_int	1'b0	R	UART RX timeout interrupt
3	urx_frdy_int	1'b0	R	UART RX FIFO ready (rx_fifo_cnt > rx_fifo_th) interrupt, auto cleared when data is popped.
2	utx_frdy_int	1'b1	R	UART TX FIFO ready (tx_fifo_cnt > tx_fifo_th) interrupt, auto cleared when data is pushed.
1	urx_end_int	1'b0	R	UART RX transfer end interrupt (set according to cr_urx_len)
0	utx_end_int	1'b0	R	UART TX transfer end interrupt (set according to cr_utx_len)

6.11.10 UART_INT_MASK, 0x20004024

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	20'b0	R	Reserved, always read 'b0
11	cr_urx_ad5_mask	1'b1	R/W	Interrupt mask of urx_ad5_int
10	cr_urx_ads_mask	1'b1	R/W	Interrupt mask of urx_ads_int
9	cr_urx_bcr_mask	1'b1	R/W	Interrupt mask of urx_bcr_int
8	cr_urx_lse_mask	1'b1	R/W	Interrupt mask of urx_lse_int
7	cr_urx_fer_mask	1'b1	R/W	Interrupt mask of urx_fer_int
6	cr_utx_fer_mask	1'b1	R/W	Interrupt mask of utx_fer_int
5	cr_urx_pce_mask	1'b1	R/W	Interrupt mask of urx_pce_int
4	cr_urx_rto_mask	1'b1	R/W	Interrupt mask of urx_rto_int
3	cr_urx_frdy_mask	1'b1	R/W	Interrupt mask of urx_fifo_int
2	cr_utx_frdy_mask	1'b1	R/W	Interrupt mask of utx_fifo_int
1	cr_urx_end_mask	1'b1	R/W	Interrupt mask of urx_end_int
0	cr_utx_end_mask	1'b1	R/W	Interrupt mask of utx_end_int

6.11.11 UART_INT_CLR, 0x20004028

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	20'b0	R	Reserved, always read 'b0
11	cr_urx_ad5_clr	1'b0	R/W1c	Interrupt clear of urx_ad5_int
10	cr_urx_ads_clr	1'b0	R/W1c	Interrupt clear of urx_ads_int
9	cr_urx_bcr_clr	1'b0	R/W1c	Interrupt clear of urx_bcr_int
8	cr_urx_lse_clr	1'b0	R/W1c	Interrupt clear of urx_lse_int
7:6	Reserved	2'b0	R/W	Reserved
5	cr_urx_pce_clr	1'b0	R/W1c	Interrupt clear of urx_pce_int
4	cr_urx_rto_clr	1'b0	R/W1c	Interrupt clear of urx_rto_int
3:2	Reserved	2'b0	R/W	Reserved
1	cr_urx_end_clr	1'b0	R/W1c	Interrupt clear of urx_end_int
0	cr_utx_end_clr	1'b0	R/W1c	Interrupt clear of utx_end_int

6.11.12 UART_INT_EN, 0x2000402C

Bits	Bit Name	Default	Type	Comment
31:12	Reserved	20'b0	R	Reserved, always read 'b0
11	cr_urx_ad5_en	1'b1	R/W	Interrupt enable of urx_ad5_int
10	cr_urx_ads_en	1'b1	R/W	Interrupt enable of urx_ads_int
9	cr_urx_bcr_en	1'b1	R/W	Interrupt enable of urx_bcr_int
8	cr_urx_lse_en	1'b1	R/W	Interrupt enable of urx_lse_int
7	cr_urx_fer_en	1'b1	R/W	Interrupt enable of urx_fer_int
6	cr_utx_fer_en	1'b1	R/W	Interrupt enable of utx_fer_int
5	cr_urx_pce_en	1'b1	R/W	Interrupt enable of urx_pce_int
4	cr_urx_rto_en	1'b1	R/W	Interrupt enable of urx_rto_int
3	cr_urx_frdy_en	1'b1	R/W	Interrupt enable of urx_fifo_int
2	cr_utx_frdy_en	1'b1	R/W	Interrupt enable of utx_fifo_int
1	cr_urx_end_en	1'b1	R/W	Interrupt enable of urx_end_int
0	cr_utx_end_en	1'b1	R/W	Interrupt enable of utx_end_int

6.11.13 UART_STATUS, 0x20004030

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30'b0	R	Reserved, always read 'b0
1	sts_urx_bus_busy	1'b0	R	Indicator of UART RX bus busy
0	sts_utx_bus_busy	1'b0	R	Indicator of UART TX bus busy

6.11.14 STS_URX_ABR_PRD, 0x20004034

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_abr_prd_0x55	16'b0	R	Bit period of Auto Baud Rate detection using codeword 0x55
15:0	sts_urx_abr_prd_start	16'b0	R	Bit period of Auto Baud Rate detection using START bit

6.11.15 URX_ABR_PRD_B01, 0x20004038

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_abr_prd_bit1	16’b0	R	Bit period of Auto Baud Rate detection - bit[1]
15:0	sts_urx_abr_prd_bit0	16’b0	R	Bit period of Auto Baud Rate detection - bit[0]

6.11.16 URX_ABR_PRD_B23, 0x2000403C

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_abr_prd_bit3	16’b0	R	Bit period of Auto Baud Rate detection - bit[3]
15:0	sts_urx_abr_prd_bit2	16’b0	R	Bit period of Auto Baud Rate detection - bit[2]

6.11.17 URX_ABR_PRD_B45, 0x20004040

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_abr_prd_bit5	16’b0	R	Bit period of Auto Baud Rate detection - bit[5]
15:0	sts_urx_abr_prd_bit4	16’b0	R	Bit period of Auto Baud Rate detection - bit[4]

6.11.18 URX_ABR_PRD_B67, 0x20004044

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_abr_prd_bit7	16’b0	R	Bit period of Auto Baud Rate detection - bit[7]
15:0	sts_urx_abr_prd_bit6	16’b0	R	Bit period of Auto Baud Rate detection - bit[6]

6.11.19 URX_ABR_PW_TOL, 0x20004048

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24’b0	R	Reserved, always read ‘b0
7:0	cr_urx_abr_pw_tol	8’h3	R/W	Auto Baud Rate detection pulse-width tolerance for using codeword 0x55

6.11.20 URX_BCR_INT_CFG, 0x20004050

Bits	Bit Name	Default	Type	Comment
31:16	sts_urx_bcr_count	16’b0	R	Current byte count of urx_bcr_int counter, auto-cleared bt cr_urx_bcr_clr
15:0	cr_urx_bcr_value	16’hFFFF	R/W	Byte count setting for urx_bcr_int counter

6.11.21 UTX_RS485_CFG, 0x20004054

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30’b0	R	Reserved, always read ‘b0
1	cr_utx_rs485_pol	1’b1	R/W	cr_utx_rs485_pol
0	cr_utx_rs485_en	1’b0	R/W	cr_utx_rs485_pol

6.11.22 UART_FIFO_CONFIG_0, 0x20004080

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7	rx_fifo_underflow	1'b0	R	Underflow flag of RX FIFO, can be cleared by rx_fifo_clr
6	rx_fifo_overflow	1'b0	R	Overflow flag of RX FIFO, can be cleared by rx_fifo_clr
5	tx_fifo_underflow	1'b0	R	Underflow flag of TX FIFO, can be cleared by tx_fifo_clr
4	tx_fifo_overflow	1'b0	R	Overflow flag of TX FIFO, can be cleared by tx_fifo_clr
3	rx_fifo_clr	1'b0	R/W1c	Clear signal of RX FIFO
2	tx_fifo_clr	1'b0	R/W1c	Clear signal of TX FIFO
1	uart_dma_rx_en	1'b0	R/W	Enable signal of dma_rx_req/ack interface
0	uart_dma_tx_en	1'b0	R/W	Enable signal of dma_tx_req/ack interface

6.11.23 UART_FIFO_CONFIG_1, 0x20004084

Bits	Bit Name	Default	Type	Comment
31	Reserved	1'b0	R	Reserved, always read 'b0
30:24	rx_fifo_th	7'b0	R/W	RX FIFO threshold, dma_rx_req will not be asserted if rx_fifo_cnt is less than this value
23	Reserved	1'b0	R	Reserved, always read 'b0
22:16	tx_fifo_th	7'b0	R/W	TX FIFO threshold, dma_tx_req will not be asserted if tx_fifo_cnt is less than this value
15:8	rx_fifo_cnt	8'b0	R	RX FIFO available count
7:0	tx_fifo_cnt	8'h80	R	TX FIFO available count

6.11.24 UART_FIFO_WDATA, 0x20004088

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7:0	uart_fifo_wdata	8'b0	W	UART FIFO write data

6.11.25 UART_FIFO_RDATA, 0x2000408C

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	24'b0	R	Reserved, always read 'b0
7:0	uart_fifo_rdata	8'b0	R	UART FIFO read data

6.12 GPIO Control Register

GPIO CSR address = `0x2000_5000 ~ 0x2000_5FFF`

6.12.1 GPIO_MISC, 0x20005080

Bits	Bit Name	Default	Type	Comment
31:2	Reserved	30'b0	R	Reserved, always read 'b0
1	reg_spi_swap	1'b0	R/W	SPI config: swap miso/mosi
0	reg_spi_master_mode	1'b0	R/W	SPI config: master/slave mode

6.12.2 GPIO_CFGCTL0, 0x20005100

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4'b0	R	Reserved, always read 'b0
27:24	reg_gpio_1_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2'b0	R	Reserved, always read 'b0
21	reg_gpio_1_pd	1'b0	R/W	GPIO pull down control
20	reg_gpio_1_pu	1'b0	R/W	GPIO pull up control
19:18	reg_gpio_1_drv	2'b0	R/W	GPIO driving control
17	reg_gpio_1_smt	1'b1	R/W	GPIO SMT control
16	reg_gpio_1_ie	1'b1	R/W	GPIO input enable
15:12	Reserved	4'b0	R	Reserved, always read 'b0
11:8	reg_gpio_0_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2'b0	R	Reserved, always read 'b0
5	reg_gpio_0_pd	1'b0	R/W	GPIO pull down control
4	reg_gpio_0_pu	1'b0	R/W	GPIO pull up control
3:2	reg_gpio_0_drv	2'b0	R/W	GPIO driving control
1	reg_gpio_0_smt	1'b1	R/W	GPIO SMT control
0	reg_gpio_0_ie	1'b1	R/W	GPIO input enable

6.12.3 GPIO_CFGCTL1, 0x20005104

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4'b0	R	Reserved, always read 'b0
27:24	reg_gpio_3_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2'b0	R	Reserved, always read 'b0
21	reg_gpio_3_pd	1'b0	R/W	GPIO pull down control
20	reg_gpio_3_pu	1'b0	R/W	GPIO pull up control
19:18	reg_gpio_3_drv	2'b0	R/W	GPIO driving control
17	reg_gpio_3_smt	1'b1	R/W	GPIO SMT control
16	reg_gpio_3_ie	1'b1	R/W	GPIO input enable
15:12	Reserved	4'b0	R	Reserved, always read 'b0
11:8	reg_gpio_2_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2'b0	R	Reserved, always read 'b0
5	reg_gpio_2_pd	1'b0	R/W	GPIO pull down control
4	reg_gpio_2_pu	1'b0	R/W	GPIO pull up control
3:2	reg_gpio_2_drv	2'b0	R/W	GPIO driving control
1	reg_gpio_2_smt	1'b1	R/W	GPIO SMT control

Bits	Bit Name	Default	Type	Comment
0	reg_gpio_2_ie	1’b1	R/W	GPIO input enable

6.12.4 GPIO_CFGCTL2, 0x20005108

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_5_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_5_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_5_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_5_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_5_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_5_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_4_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_4_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_4_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_4_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_4_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_4_ie	1’b1	R/W	GPIO input enable

6.12.5 GPIO_CFGCTL3, 0x2000510C

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_7_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_7_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_7_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_7_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_7_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_7_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_6_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_6_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_6_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_6_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_6_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_6_ie	1’b1	R/W	GPIO input enable

6.12.6 GPIO_CFGCTL4, 0x20005110

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_9_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_9_pd	1’b0	R/W	GPIO pull down control

Bits	Bit Name	Default	Type	Comment
20	reg_gpio_9_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_9_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_9_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_9_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_8_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_8_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_8_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_8_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_8_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_8_ie	1’b1	R/W	GPIO input enable

6.12.7 GPIO_CFGCTL5, 0x20005114

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_11_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_11_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_11_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_11_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_11_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_11_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_10_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_10_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_10_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_10_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_10_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_10_ie	1’b1	R/W	GPIO input enable

6.12.8 GPIO_CFGCTL6, 0x20005118

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_13_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_13_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_13_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_13_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_13_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_13_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_12_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_12_pd	1’b0	R/W	GPIO pull down control

Bits	Bit Name	Default	Type	Comment
4	reg_gpio_12_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_12_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_12_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_12_ie	1’b1	R/W	GPIO input enable

6.12.9 GPIO_CFGCTL7, 0x2000511C

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_15_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_15_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_15_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_15_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_15_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_15_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_14_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_14_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_14_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_14_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_14_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_14_ie	1’b1	R/W	GPIO input enable

6.12.10 GPIO_CFGCTL8, 0x20005120

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0
27:24	reg_gpio_17_func_sel	4’b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2’b0	R	Reserved, always read ‘b0
21	reg_gpio_17_pd	1’b0	R/W	GPIO pull down control
20	reg_gpio_17_pu	1’b0	R/W	GPIO pull up control
19:18	reg_gpio_17_drv	2’b0	R/W	GPIO driving control
17	reg_gpio_17_smt	1’b1	R/W	GPIO SMT control
16	reg_gpio_17_ie	1’b1	R/W	GPIO input enable
15:12	Reserved	4’b0	R	Reserved, always read ‘b0
11:8	reg_gpio_16_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2’b0	R	Reserved, always read ‘b0
5	reg_gpio_16_pd	1’b0	R/W	GPIO pull down control
4	reg_gpio_16_pu	1’b0	R/W	GPIO pull up control
3:2	reg_gpio_16_drv	2’b0	R/W	GPIO driving control
1	reg_gpio_16_smt	1’b1	R/W	GPIO SMT control
0	reg_gpio_16_ie	1’b1	R/W	GPIO input enable

6.12.11 GPIO_CFGCTL9, 0x20005124

Bits	Bit Name	Default	Type	Comment
31:28	Reserved	4’b0	R	Reserved, always read ‘b0

Bits	Bit Name	Default	Type	Comment
27:24	reg_gpio_19_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
23:22	Reserved	2'b0	R	Reserved, always read 'b0
21	reg_gpio_19_pd	1'b0	R/W	GPIO pull down control
20	reg_gpio_19_pu	1'b0	R/W	GPIO pull up control
19:18	reg_gpio_19_drv	2'b0	R/W	GPIO driving control
17	reg_gpio_19_smt	1'b1	R/W	GPIO SMT control
16	reg_gpio_19_ie	1'b1	R/W	GPIO input enable
15:12	Reserved	4'b0	R	Reserved, always read 'b0
11:8	reg_gpio_18_func_sel	4'b0	R/W	GPIO function select (Default: SWGPIO)
7:6	Reserved	2'b0	R	Reserved, always read 'b0
5	reg_gpio_18_pd	1'b0	R/W	GPIO pull down control
4	reg_gpio_18_pu	1'b0	R/W	GPIO pull up control
3:2	reg_gpio_18_drv	2'b0	R/W	GPIO driving control
1	reg_gpio_18_smt	1'b1	R/W	GPIO SMT control
0	reg_gpio_18_ie	1'b1	R/W	GPIO input enable

6.12.12 GPIO_CFGCTL30, 0x20005180

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19	reg_gpio_19_i	1'b0	R	Register controlled GPIO input value
18	reg_gpio_18_i	1'b0	R	Register controlled GPIO input value
17	reg_gpio_17_i	1'b0	R	Register controlled GPIO input value
16	reg_gpio_16_i	1'b0	R	Register controlled GPIO input value
15	reg_gpio_15_i	1'b0	R	Register controlled GPIO input value
14	reg_gpio_14_i	1'b0	R	Register controlled GPIO input value
13	reg_gpio_13_i	1'b0	R	Register controlled GPIO input value
12	reg_gpio_12_i	1'b0	R	Register controlled GPIO input value
11	reg_gpio_11_i	1'b0	R	Register controlled GPIO input value
10	reg_gpio_10_i	1'b0	R	Register controlled GPIO input value
9	reg_gpio_9_i	1'b0	R	Register controlled GPIO input value
8	reg_gpio_8_i	1'b0	R	Register controlled GPIO input value
7	reg_gpio_7_i	1'b0	R	Register controlled GPIO input value
6	reg_gpio_6_i	1'b0	R	Register controlled GPIO input value
5	reg_gpio_5_i	1'b0	R	Register controlled GPIO input value
4	reg_gpio_4_i	1'b0	R	Register controlled GPIO input value
3	reg_gpio_3_i	1'b0	R	Register controlled GPIO input value
2	reg_gpio_2_i	1'b0	R	Register controlled GPIO input value
1	reg_gpio_1_i	1'b0	R	Register controlled GPIO input value
0	reg_gpio_0_i	1'b0	R	Register controlled GPIO input value

6.12.13 GPIO_CFGCTL32, 0x20005188

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19	reg_gpio_19_o	1'b0	R/W	Register controlled GPIO output Value
18	reg_gpio_18_o	1'b0	R/W	Register controlled GPIO output Value
17	reg_gpio_17_o	1'b0	R/W	Register controlled GPIO output Value

Bits	Bit Name	Default	Type	Comment
16	reg_gpio_16_o	1'b0	R/W	Register controlled GPIO output Value
15	reg_gpio_15_o	1'b0	R/W	Register controlled GPIO output Value
14	reg_gpio_14_o	1'b0	R/W	Register controlled GPIO output Value
13	reg_gpio_13_o	1'b0	R/W	Register controlled GPIO output Value
12	reg_gpio_12_o	1'b0	R/W	Register controlled GPIO output Value
11	reg_gpio_11_o	1'b0	R/W	Register controlled GPIO output Value
10	reg_gpio_10_o	1'b0	R/W	Register controlled GPIO output Value
9	reg_gpio_9_o	1'b0	R/W	Register controlled GPIO output Value
8	reg_gpio_8_o	1'b0	R/W	Register controlled GPIO output Value
7	reg_gpio_7_o	1'b0	R/W	Register controlled GPIO output Value
6	reg_gpio_6_o	1'b0	R/W	Register controlled GPIO output Value
5	reg_gpio_5_o	1'b0	R/W	Register controlled GPIO output Value
4	reg_gpio_4_o	1'b0	R/W	Register controlled GPIO output Value
3	reg_gpio_3_o	1'b0	R/W	Register controlled GPIO output Value
2	reg_gpio_2_o	1'b0	R/W	Register controlled GPIO output Value
1	reg_gpio_1_o	1'b0	R/W	Register controlled GPIO output Value
0	reg_gpio_0_o	1'b0	R/W	Register controlled GPIO output Value

6.12.14 GPIO_CFGCTL34, 0x20005190

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19	reg_gpio_19_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
18	reg_gpio_18_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
17	reg_gpio_17_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
16	reg_gpio_16_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
15	reg_gpio_15_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
14	reg_gpio_14_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
13	reg_gpio_13_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
12	reg_gpio_12_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
11	reg_gpio_11_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
10	reg_gpio_10_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
9	reg_gpio_9_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
8	reg_gpio_8_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
7	reg_gpio_7_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
6	reg_gpio_6_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
5	reg_gpio_5_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
4	reg_gpio_4_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)

Bits	Bit Name	Default	Type	Comment
3	reg_gpio_3_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
2	reg_gpio_2_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
1	reg_gpio_1_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)
0	reg_gpio_0_oe	1'b0	R/W	Register controlled GPIO output enable (used when GPIO function select to register control GPIO)

6.12.15 GPIO_INT_MASK1, 0x200051A0

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	reg_gpio_int_mask1	20'hFFFFF	R/W	reg_gpio_int_mask [19:0]

6.12.16 GPIO_INT_STAT1, 0x200051A8

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	gpio_int_stat1	20'b0	R	gpio_int_stat [19:0]

6.12.17 GPIO_INT_CLR1, 0x200051B0

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:0	reg_gpio_int_clr1	20'b0	R/W	reg_gpio_int_clr [19:0]

6.12.18 GPIO_INT_MODE_SET1, 0x200051C0

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	12'b0	R	Reserved, always read 'b0
29:0	reg_gpio_int_mode_set1	20'b0	R/W	{reg_gpio9_int_mode [2:0], ……. reg_gpio0_int_mode [2:0]} GPIO X Interrupt Mode Control [2:0] Bit[2]: 1=async mode, 0=sync mode; Bit[1:0]: 00 negedge pulse, 01=posedge pulse, 10=negedge level (32k 3T), 11=posedge level (32k 3T)

6.12.19 GPIO_INT_MODE_SET2, 0x200051C4

Bits	Bit Name	Default	Type	Comment
31:30	Reserved	12’b0	R	Reserved, always read ‘b0
29:0	reg_gpio_int_mode_set2	20’b0	R/W	{reg_gpio19_int_mode [2:0], ……. reg_gpio10_int_mode [2:0]} GPIO X Interrupt Mode Control [2:0] Bit[2]: 1=async mode, 0=sync mode; Bit[1:0]: 00 negedge pulse, 01=posedge pulse, 10=negedge level (32k 3T), 11=posedge level (32k 3T)

6.13 EMAC Register

EMAC register address = `0x2B00_0000 ~ 0x2B00_FFFF`

6.13.1 MODE, 0x2B000000

Bits	Bit Name	Default	Type	Comment
31:18	Reserved	14'b0	R	Reserved, always read 'b0
17	RMII_EN	1'b0	R/W	RMII mode enable 0: MII PHY I/F is used 1: RMII PHY I/F is used
16	RECSMALL	1'b0	R/W	Receive small frame enable 0: Frames smaller than MINFL are ignored. 1: Frames smaller than MINFL are accepted
15	PAD	1'b1	R/W	Padding enable 0: Do not add pads to frames shorter than MINFL 1: Add pads to short frames, until the length equals MINFL
14	HUGEN	1'b0	R/W	Huge frames enable 0: The maximum frame length is MAXFL. All additional bytes are dropped 1: Frame size is not limited by MAXFL and can be up to 64K bytes
13	CRCEN	1'b1	R/W	CRC Enable 0: TX MAC does not append CRC field 1: TX MAC will append CRC field to every frame
12	DlyCrcEn	1'b0	R/W	
11	Reserved	1'b0	R/W	Reserved
10	FULLD	1'b0	R/W	Full duplex 0: Half duplex mode 1: Full duplex mode
9	ExDfrEn	1'b0	R/W	
8	NoBckof	1'b0	R/W	
7	LoopBck	1'b0	R/W	
6	IFG	1'b0	R/W	Inter frame gap check 0: IFG is verified before each frame be received 1: All frames are received regardless to IFG requirement
5	PRO	1'b0	R/W	Promiscuous mode enable 0: The destination address is checked before receiving 1: All frames received regardless of the address
4	Iam	1'b0	R/W	
3	BRO	1'b1	R/W	Broadcast address enable 0: Reject all frames containing the broadcast address unless the PRO bit is asserted. 1: Receive all frames containing broadcast address
2	NOPRE	1'b0	R/W	No preamble mode 0: 7-byte preamble will be sent 1: No preamble will be sent
1	TXEN	1'b0	R/W	Transmit enable 0: Transmitter is disabled 1: Transmitter is enabled If TX_BD_NUM equals 0x0 (zero buffer descriptors are used), then the transmitter is disabled regardless of TXEN

Bits	Bit Name	Default	Type	Comment
0	RXEN	1'b0	R/W	Receive enable 0: Receiver is disabled 1: Receiver is enabled If TX_BD_NUM equals 0x80 (all buffer descriptors are used for TX), then the receiver is disabled regardless of RXEN

6.13.2 INT_SOURCE, 0x2B000004

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25'b0	R	Reserved, always read 'b0
6	RXC	1'b0	R	Receive control frame This bit indicates that the control frame was received. It is cleared by writing 1 to it. Bit RXFLOW in the CTRLMODE register must be set to 1 to get the RXC bit set.
5	TXC	1'b0	R	Transmit control frame This bit indicates that a control frame was transmitted. It is cleared by writing 1 to it. Bit TXFLOW in the CTRLMODE register must be set to 1 to get the TXC bit set.
4	BUSY	1'b0	R	Busy This bit indicates that RX packet is being received and there is no empty buffer descriptor to use. It is cleared by writing 1 to it. This bit appears regardless to the IRQ bits in the Receive Buffer Descriptor.
3	RXE	1'b0	R	Receive error This bit indicates that an error occurred while receiving data (overrun, receiver error, dribble nibble, too long, >64K, CRC error, bus error or late collision. It is cleared by writing 1 to it.
2	RXB	1'b0	R	Receive frame This bit indicates that a frame was received. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Receive Buffer Descriptor.
1	TXE	1'b0	R	Transmit error This bit indicates that a buffer was not transmitted due to a transmit error (underrun, retransmission limit, late collision, bus error or defer timeout). It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.
0	TXB	1'b0	R	Transmit buffer This bit indicates that a buffer has been transmitted. It is cleared by writing 1 to it. This bit appears only when IRQ bit is set in the Transmit Buffer Descriptor.

6.13.3 INT_MASK, 0x2B000008

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25'b0	R	Reserved, always read 'b0
6	RXC_M	1'b1	R/W	Receive control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked

Bits	Bit Name	Default	Type	Comment
5	TXC_M	1'b1	R/W	Transmit control frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
4	BUSY_M	1'b1	R/W	Busy mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
3	RXE_M	1'b1	R/W	Receive error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
2	RXB_M	1'b1	R/W	Receive frame mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
1	TXE_M	1'b1	R/W	Transmit error mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked
0	TXB_M	1'b1	R/W	Transmit buffer mask ENABLE 0: Interrupt is un-masked 1: Interrupt is masked

6.13.4 IPGT, 0x2B00000C

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25'b0	R	Reserved, always read 'b0
6:0	IPGT	7'h18	R/W	Inter packet gap The recommended value is 0x18 (24 clock cycles), which equals 9.6 us for 10 Mbps and 0.96 us for 100 Mbps mode

6.13.5 IPGR1, 0x2B000010

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25'b0	R	Reserved, always read 'b0
6:0	IPGR1	7'h18	R/W	

6.13.6 IPGR2, 0x2B000014

Bits	Bit Name	Default	Type	Comment
31:7	Reserved	25'b0	R	Reserved, always read 'b0
6:0	IPGR2	7'h18	R/W	

6.13.7 PACKETLEN, 0x2B000018

Bits	Bit Name	Default	Type	Comment
31:16	MINFL	16'h40	R/W	Reserved Minimum frame length The minimum Ethernet packet is 64 bytes long (0x40). To receive small packets, assert the RECSMALL bit or change the MINFL value. To transmit small packets, assert the PAD bit or change the MINFL value.

Bits	Bit Name	Default	Type	Comment
15:0	MAXFL	16'h600	R/W	Maximum frame length The maximum Ethernet packet is 1518 bytes long. To support this and to have some additional space for tags, a default maximum packet length equals to 1536 bytes (0x600). For bigger packets, you can assert the HUGEN bit or increase the value of MAXFL field.

6.13.8 COLLCONFIG, 0x2B00001C

Bits	Bit Name	Default	Type	Comment
31:20	Reserved	12'b0	R	Reserved, always read 'b0
19:16	MAXRET	4'hF	R/W	Maximum retry This field specifies the maximum number of consequential retransmission attempts after the collision is detected. When the maximum number has been reached, the TX MAC reports an error and stops transmitting the current packet. According to the Ethernet standard, the MAXRET default value is set to 0xf (15).
15:6	Reserved	10'b0	R	Reserved, always read 'b0
5:0	COLLVALID	6'h3F	R/W	Collision valid This field specifies a collision time window. A collision that occurs later than the time window is reported as a "Late Collisions" and transmission of the current packet is aborted.

6.13.9 TX_BD_NUM, 0x2B000020

Bits	Bit Name	Default	Type	Comment
31	Reserved	1'b0	R	Reserved, always read 'b0
30:24	RXBDPTR	7'b0	R	RX buffer descriptors (BD) pointer, pointing at the RXBD currently being used
23	Reserved	1'b0	R	Reserved, always read 'b0
22:16	TXBDPTR	7'b0	R	TX buffer descriptors (BD) pointer, pointing at the TXBD currently being used
15:8	Reserved	8'b0	R	Reserved, always read 'b0
7:0	TXBDNUM	8'h40	R/W	TX buffer descriptors (BD) number Number of TX BD. TX and RX share 128 (0x80) descriptors, so the number of RX BD equals 0x80 - TXBDNUM. The maximum number of TXBDNUM is 0x80. Values greater then 0x80 cannot be written into this register.

6.13.10 CTRLMODER, 0x2B000024

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29'b0	R	Reserved, always read 'b0
2	TxFlow	1'b0	R/W	
1	RxFlow	1'b0	R/W	
0	PassAll	1'b0	R/W	

6.13.11 MIIMODE, 0x2B000028

Bits	Bit Name	Default	Type	Comment
31:9	Reserved	23'b0	R	Reserved, always read 'b0
8	MIINOPRE	1'b0	R/W	No preamble for Management Data (MD) 0: 32-bit preamble will be sent 1: No preamble will be sent
7:0	CLKDIV	8'h64	R/W	Clock divider for Management Data Clock (MDC) The source clock is bus clock and can be divided by any even number

6.13.12 MIICOMMAND, 0x2B00002C

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29'b0	R	Reserved, always read 'b0
2	WCTRLDATA	1'b0	R/W	Write control data, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
1	RSTAT	1'b0	R/W	Read status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time
0	SCANSTST	1'b0	R/W	Scan status, setting this bit to 1 will trigger the command (auto cleared) Note: [2]/[1]/[0] cannot be asserted at the same time, execute one command at a time

6.13.13 MIIADDRESS, 0x2B000030

Bits	Bit Name	Default	Type	Comment
31:13	Reserved	19'b0	R	Reserved, always read 'b0
12:8	RGAD	5'b0	R/W	Register Address
7:5	Reserved	3'b0	R	Reserved, always read 'b0
4:0	FIAD	5'b0	R/W	PHY Address

6.13.14 MIITX_DATA, 0x2B000034

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16'b0	R	Reserved, always read 'b0
15:0	CTRLDATA	16'b0	R/W	Control Data to be written to PHY

6.13.15 MIIRX_DATA, 0x2B000038

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16'b0	R	Reserved, always read 'b0
15:0	PRSD	16'b0	R	Received Data from PHY

6.13.16 MIISTATUS, 0x2B00003C

Bits	Bit Name	Default	Type	Comment
31:3	Reserved	29’b0	R	Reserved, always read ‘b0
2	MIIM_NVALID	1'b0	R	
1	MIIM_BUSY	1'b0	R	MIIM I/F busy signal 0: The MIIM I/F is ready 1: The MIIM I/F is busy
0	MIIM_LINKFAIL	1'b0	R	MIIM I/F link fail signal

6.13.17 MAC_ADDR0, 0x2B000040

Bits	Bit Name	Default	Type	Comment
31:24	MAC_B2	8'b0	R/W	Ethernet MAC address byte 2
23:16	MAC_B3	8'b0	R/W	Ethernet MAC address byte 3
15:8	MAC_B4	8'b0	R/W	Ethernet MAC address byte 4
7:0	MAC_B5	8'b0	R/W	Ethernet MAC address byte 5

6.13.18 MAC_ADDR1, 0x2B000044

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	16’b0	R	Reserved, always read ‘b0
15:8	MAC_B0	8'b0	R/W	Ethernet MAC address byte 0
7:0	MAC_B1	8'b0	R/W	Ethernet MAC address byte 1

6.13.19 HASH0_ADDR, 0x2B000048

Bits	Bit Name	Default	Type	Comment
31:0	HASH0	32'h0	R/W	Lower 32-bit of HASH register

6.13.20 HASH1_ADDR, 0x2B00004C

Bits	Bit Name	Default	Type	Comment
31:0	HASH1	32'h0	R/W	Upper 32-bit of HASH register

6.13.21 TXCTRL, 0x2B000050

Bits	Bit Name	Default	Type	Comment
31:17	Reserved	15’b0	R	Reserved, always read ‘b0
16	TXPAUSERQ	1'b0	R/W	TX Pause Request Writing 1 to this bit starts sending control frame and is automatically cleared to zero
15:0	TXPAUSETV	16'b0	R/W	TX Pause Timer Value The value that is sent in the pause control frame

6.13.22 DBG, 0x2B000058

Bits	Bit Name	Default	Type	Comment
31:0	DBG_DAT	32'h0	R	Debug data ???

6.14 I3C Register

I3C Master Register Base address = 0x2A000000

I3C Slave Register Base address = 0x2A010000

For more information, see Chapter 5 of DWC_mipi_i3c_databook.pdf here

6.15 AXI4TG Register

AXI4TG Register Base address = 0x2A020000

For more information, see axi-traffic-gen.pdf. here

Notes: Address width is configured to 32, the Address RAM is not present and cannot be accessed.

6.15.1 Master Control, 0x00

Bits	Bit Name	Default	Type	Comment
31:24	Reserve	8’b0	R	Revision of the core
23:16	MSTID	3’b0	R	M_ID_WIDTH, where: 0x0 = Indicates 0 or 1-bit width 0x1 = Indicates 2-bit width ... 0x7 = Indicates 8-bit width
20	MSTEN	1’b0	R/W	Master Enable When set, the master logic begins. When both the Read and Write state machines complete, this bit is automatically cleared to indicate to software that the AXI Traffic Generator is done.
19	Loop Enable	1’b0	R/W	Loop Enable 1. Loops through the command set created using CMDRAM and PARAMRAM (as applicable) indefinitely when set to 1. 2. When this bit is reset to 0, core stops looping after the current command set of transactions are completed. 3. Dependency (if any, both mydepend and otherdepend) is ignored when loop enable is set. Dependency gets honored after the loop enable is reset to 0. 4. Both channels loopback to their first command independently without waiting for the outstanding transactions to get completed. 5. If interrupt is enabled, core generates irq_out after completing the command set following the reset of loop enable to 0. Note: Dependency for the last command set run is based on the point at which the loop enable is reset to 0. For example, a command set with 12 writes and 16 reads are present with the 13th read is dependent on sixth write. Now if the loop enable is reset to 0 before sixth write and 13th read of command run, you see the dependency in the last run else the dependency is not seen even after loop enable is reset. For bullet point 4, consider a case of a command set with 50 write commands and two read commands. In sµCh a case, the read command should get repeated more than once before one set of write commands are completed.

6.15.2 Slave Control, 0x04

Bits	Bit Name	Default	Type	Comment
31:20	Reserve	N/A	N/A	Reserved
19	BLKRD	0x0	R/W	Enable Block Read When set, slave reads are not processed if there are any pending writes. On completing each write, at least one read data is returned to prevent starvation
18	DISEXCL	0X0	R/W	Disable Exclusive Access When set, disables exclusive access support and error response ability for reads on Slave Error register..

Bits	Bit Name	Default	Type	Comment
17	WORDR	0X0	R/W	Enable in Order Write Response When set, forces all BRESPs to be issued in the order the requests were received.
16	RORDR	0X0	R/W	Enable in Order Read Response When set, forces all slave reads to be done in the order received
15	ERREN	0X0	R/W	Enable Error Generation When set, if any bit in Error Status register Bits[15:0] is set, then err_out is asserted.
14:0	Reserved	N/A	N/A	Reserved

6.15.3 Error Status, 0x08

Bits	Bit Name	Default	Type	Comment
31	MSTDONE	0x0	R/W1C	Master Completion Set when both master write and master read CMD logic completes and Error Enable register Bit[31] is 1. When set, irq_out is driven to 1.
30:21	Reserved	N/A	N/A	Reserved
20	RIDER	0x0	R/W1C	Master Read ID Error On master interface Received an RVALID with a RID that did not match any pending reads.
19	WIDER	0x0	R/W1C	Master Write ID Error Received a BVALID with a BID that did not match any pending writes.
18	WRSPER	N/A	R/W1C	Master Write Response Error On a master write completion, the response returned was not allowed by expected_resp[2:0].
17	RERRSP	0x0	R/W1C	Master Read Response Error On a master read completion, the response returned was not allowed by expected_resp[2:0].
16	RLENER	0x0	R/W1C	Master Read Length Error On the master interface Rlast either when it was not expected or was not signaled when it was expected.
15:2	Reserved	N/A	N/A	Reserved
1	SWSTRB	0x0	R/W1C	Slave Write Strobe Error On the slave interface, a WSTRB assertion was detected on an illegal byte lane
0	SWLENER	0x0	R/W1C	Slave Write Length Error On the slave interface W, Last was signaled either when it was not expected or was not signaled when it was expected.

6.15.4 Error Enable, 0x0C

Bits	Bit Name	Default	Type	Comment
31	MSTIRQEN	0x1	R/W	Enables interrupt generation for Master transfer completion.
30:21	Reserved	N/A	N/A	Reserved
20	RIDEREN	0x0	R/W	Enables Read ID Error for Error Status register Bit[20].
19	WIDEREN	0x0	R/W	Enables Write ID error for Error Status register Bit[19].
18	WRSPER	N/A	R/W	Enables write response error for Error Status register Bit[18].
17	RERRSP	0x0	R/W	Enables read response error for Error Status register Bit[17]
16	RLENER	0x0	R/W	Enables read length error for Error Status register Bit[16].
15:2	Reserved	N/A	N/A	Reserved
1	SWSTRBEN	0x0	R/W	Enables slave write strobe error for Error Status register Bit[1]
0	SWLENEREN	0x0	R/W	Enables slave write length error for Error Status register Bit[0]

6.15.5 Master Error Interrupt Enable, 0x10

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	N/A	N/A	Reserved
15	MINTREN	0x0	R/W	Enables Master Interrupt When set, if any bit in Error Status register Bits[30:16] is set, then err_out is asserted.
14:0	Reserved	N/A	N/A	Reserved

6.15.6 Config Status, 0x14

Bits	Bit Name	Default	Type	Comment
31	Reserved	N/A	N/A	Reserved
30:28	MWIDTH	0x0	R	Master Width 0x0 = 32-bit 0x1 = 64-bit 0x2 = 128-bit 0x3 = 256-bit 0x4 = 512-bit
27:25	SWIDTH	0x0	N/A	Slave Width 000 = 32-bit 001 = 64-bit
24	MADV	0x0	R	ATG Mode is Advanced
23	MBASIC	0x0	R	ATG Mode is Basic
22:0	Reserved	N/A	N/A	Reserved

6.15.7 Transfer Length, 0x38

Bits	Bit Name	Default	Type	Comment
31:16	TCNT	0x0	R/W	Transaction Count Core generates this many transaction on AXI4-Stream master channel and stops. If set to 0, core infinitely generates transactions.
15:0	TLEN	0x0	R/W	Length of Transaction When Random Length in Streaming Config register is not set, Length programmed in this register is used. Actual number of beats are one more than the register setting. For example, setting to 0 gives 1 beat, setting to 1 gives 2 beats, and further.

6.15.8 Transfer Count, 0x3C

Bits	Bit Name	Default	Type	Comment
31:0	TLSTCNT	0x0	R	Master Only – Reports number of streaming transactions (count of tlast) on master interface Master Loopback – Reports number of streaming transaction (count of tlast) on slave interface Slave Loopback – Reports number of streaming transaction (count of tlast) on master interface

6.15.9 User STRB / TKEEP Set 1 to 4, 0x40 ~ 0x4c

Bits	Bit Name	Default	Type	Comment
31:0	TKTS	0x0	R/W	TSTRB/TKEEP value to be appeared on the last beat of the transfer.

6.15.10 Extended Transfer Length, 0x50

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	N/A	N/A	Reserved
7:0	Ext-TLEN	0x0	R/W	When Random Length in Streaming Config register is not set, Length programmed in this register is used. TLEN in a packet. This value is concatenated to a value in the Transfer Length Bits[15:0] to give a maximum value of 224 – 1 beats in the Streaming packet.

6.15.11 Streaming Error Status Register, 0x70

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	N/A	N/A	Reserved
0	AXIS_ESR	0x0	R	Error Status This bit is set to 1 only when an error is occurred and Error Enable bit is enabled.

6.15.12 Streaming Error Enable Register, 0x74

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	N/A	N/A	Reserved

Bits	Bit Name	Default	Type	Comment
0	AXIS_EER	0x0	R/W	Error Status 1= Enables the Error Status bit 0= Disables the Error Status bit

6.15.13 Streaming Error Interrupt Enable Register, 0x78

Bits	Bit Name	Default	Type	Comment
31:1	Reserved	N/A	N/A	Reserved
0	AXIS_EIER	0x0	R/W	Error Interrupt Enable 1= Enables the Error Out bit 0= Disables the Error Out bit

6.15.14 Streaming Error Count Register, 0x7C

Bits	Bit Name	Default	Type	Comment
31:16	Reserved	N/A	N/A	Reserved
15:0	AXIS_ECR	0x0	R	Error Count Reports number of errors occurred.

6.15.15 Static Mode Control, 0x60

Bits	Bit Name	Default	Type	Comment
31:24	Version	0x20	R	Version value
23:2	Reserved	N/A	N/A	Reserved
1	DONE	0x0	R/W1C	Transfer Done 0 = Indicates core is generating traffic when STEN is 1, else core is in idle mode 1 = Indicates traffic generation completed This bit is set to 1 when the core is disabled by setting STTEN to 0 and the current transfer is completed. This bit resets to 0 either writing 1 to this bit or enabling the core with STEN.
0	STEN	0x0	R/W	Static Enable 0 = Disable traffic generation 1 = Enable traffic generation

6.15.16 Static Length, 0x64

Bits	Bit Name	Default	Type	Comment
31:8	Reserved	N/A	N/A	Reserved
7:0	BLEN	Burst Length	R/W	Burst Length Configures burst length for AXI4 master interface. Reset value is the value configured for “Burst Length” Vivado IDE.

6.16 D2D PHY Register

D2D PHY Register Base address = 0x27000000

Alternate window GFH0 = 0x2C000000; GFH1 = 0x2C010000; GFH2 = 0x2C020000; GFH3 = 0x2C030000;

For more information, see `blynx_phy_csr.pdf.` here

6.17 D2D Register

D2D Register Base address = 0x27100000

For more information, see `d2d_reg.html`. here

6.18 D2D IO

6.19 IPM MC Registers

6.20 IPM PHY Registers

6.20.1 wdq_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdq_pipe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.2 wdqs_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.3 wdq_pipe_oe_dly_D0R0F0

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdq_pipe_oe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.4 wdqs_pipe_oe_dly_D0R0F0

$1=0..3, $2=0, $3=0..10

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_oe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.5 write_en_early_pipe_dly_D0R0F0

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	write_en_early_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.6 rdqsen_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqsen_pipe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.7 rxdqen_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rxdqen_pipe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.8 rxdqsen_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rxdqsen_pipe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.9 rdqodten_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqodten_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.10 rdqsodten_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqsodten_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.11 read_en_early_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	read_en_early_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.12 rxcs_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rxcs_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.13 w_dqs_cs_pipe_dly_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	w_dqs_cs_pipe_dly	RW	0x0	Description1
7:4	w_dq_cs_pipe_dly	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.14 wdqs_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_stretch	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.15 wdq_pipe_oe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdq_pipe_oe_stretch	RW	0x8	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.16 wdqs_pipe_oe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_oe_stretch	RW	0x8	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.17 write_en_early_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	write_en_early_pipe_stret	chRW	0xc	Description1

Bits	Field name	Acc	Reset	Description
15:4	Reserved	RO	0x0	Reserved for future use

6.20.18 rdqsen_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqsen_pipe_stretch	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.19 rxdqen_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rxdqen_pipe_stretch	RW	0x6	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.20 rdqodten_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqodten_pipe_stretch	RW	0x6	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.21 rdqodten_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqodten_pipe_stretch	RW	0x6	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.22 rdqodten_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqodten_pipe_stretch	RW	0x6	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.23 rdqsodten_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdqsodten_pipe_stretch	RW	0x6	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.24 read_en_early_pipe_stretch_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	read_en_early_pipe_stret	chRW	0x8	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.25 rdpipe_control_D1_R2_F$3

$1=0..3, $2=0, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	rdpipe_control	RW	0x2	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.26 cross_sel_D1_F3

$1=0..3,$2=0..1

Bits	Field name	Acc	Reset	Description
0	cross_sel	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.27 rdfifo_reset_n_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
0	rdfifo_reset_n	RW	0x1	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.28 ovr_wren_early_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
1:0	ovr_wren_early	RW	0x0	Description1
3:2	ovr_rden_early	RW	0x0	Description1
4	trn_write_en_as_dqs	RW	0x0	Description1

Bits	Field name	Acc	Reset	Description
15:5	Reserved	RO	0x0	Reserved for future use

6.20.29 ovr_txdqoe_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
1:0	ovr_txdqoe	RW	0x0	Description1
3:2	ovr_txdqsoe	RW	0x0	Description1

6.20.30 ld_cnt_wr_pipe_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
2:0	ld_cnt_wr_pipe	RW	0x7	Description1
5:3	ld_cnt_rd_pipe	RW	0x7	Description1
7:6	ser_reset_stagesel_wr_pi	peRW	0x0	Description1
9:8	ser_reset_stagesel_rd_pip	eRW	0x0	Description1
15:10	Reserved	RO	0x0	Reserved for future use

6.20.31 rx_last_data01_p0_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
0	rx_last_data_0	RO	0x0	Description1
1	rx_last_data_1	RO	0x0	Description1
2	rx_last_data_2	RO	0x0	Description1
3	rx_last_data_3	RO	0x0	Description1
4	rx_last_data_4	RO	0x0	Description1
5	rx_last_data_5	RO	0x0	Description1
6	rx_last_data_6	RO	0x0	Description1
7	rx_last_data_7	RO	0x0	Description1
8	rx_last_data_8	RO	0x0	Description1
9	rx_last_data_9	RO	0x0	Description1
10	rx_last_data_10	RO	0x0	Description1
11	rx_last_data_11	RO	0x0	Description1
12	rx_last_data_12	RO	0x0	Description1
13	rx_last_data_13	RO	0x0	Description1
14	rx_last_data_14	RO	0x0	Description1

Bits	Field name	Acc	Reset	Description
15	rx_last_data_15	RO	0x0	Description1

6.20.32 rx_last_data_1_p0_D0

Bits	Field name	Acc	Reset	Description
0	rx_last_data_16	RO	0x0	Description1
1	rx_last_data_17	RO	0x0	Description1
2	rx_last_data_18	RO	0x0	Description1
3	rx_last_data_19	RO	0x0	Description1
4	rx_last_data_20	RO	0x0	Description1
5	rx_last_data_21	RO	0x0	Description1
6	rx_last_data_22	RO	0x0	Description1
7	rx_last_data_23	RO	0x0	Description1
8	rx_last_data_24	RO	0x0	Description1
9	rx_last_data_25	RO	0x0	Description1
10	rx_last_data_26	RO	0x0	Description1
11	rx_last_data_27	RO	0x0	Description1
12	rx_last_data_28	RO	0x0	Description1
13	rx_last_data_29	RO	0x0	Description1
14	rx_last_data_30	RO	0x0	Description1
15	rx_last_data_31	RO	0x0	Description1

6.20.33 rx_last_data_2_p0_D0

Bits	Field name	Acc	Reset	Description
0	rx_last_data_32	RO	0x0	Description1
1	rx_last_data_33	RO	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.34 rdfifo_write_pointer_even_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
3:0	rdfifo_write_pointer_even	RO	0x0	Description1
7:4	rdfifo_write_pointer_odd	RO	0x0	Description1
11:8	rdfifo_read_pointer	RO	0x0	Description1
12	rdq_ptrnbuf_cmp_pf	RO	0x0	Description1
15:13	Reserved	RO	0x0	Reserved for future use

6.20.35 wdq_ptrnbuf_start_ptr_PB$1

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	wdq_ptrnbuf_start_ptr	RW	0x0	Description1
15:8	wdq_ptrnbuf_stop_ptr	RW	0x0	Description1

6.20.36 wdq_ptrnbuf_rpt_cnt_PB$1

Bits	Field name	Acc	Reset	Description
7:0	wdq_ptrnbuf_rpt_cnt	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.37 wdq_ptrnbuf_delay_PB0_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	wdq_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.38 wdq_ptrnbuf_delay_PB1_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	wdq_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.39 wdq_ptrnbuf_delay_PB2_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	wdq_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.40 wdq_ptrnbuf_delay_PB3_0

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	wdq_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.41 wdq_ptrnbuf_dq0_PB0_$1

$1=0..15

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_0	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_1	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_2	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_3	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_4	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_5	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_6	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_7	RW 0x0 Description1

6.20.42 wdq_ptrnbuf_dq8_PB0_$1

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_8	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_9	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_10	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_11	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_12	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_13	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_14	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_15	RW 0x0 Description1

6.20.43 wdq_ptrnbuf_dq16_PB0_$1

Bits	Field name	Acc	Reset	Description
1:0	wdq_ptrnbuf_dq_16	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.44 wdq_ptrnbuf_dq0_PB1_$1

$1=0..15

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_0	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_1	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_2	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_3	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_4	RW 0x0 Description1

Bits	Field name	Acc Reset Description
11:10	wdq_ptrnbuf_dq_5	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_6	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_7	RW 0x0 Description1

6.20.45 wdq_ptrnbuf_dq8_PB1_$1

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_8	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_9	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_10	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_11	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_12	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_13	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_14	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_15	RW 0x0 Description1

6.20.46 wdq_ptrnbuf_dq16_PB1_$1

Bits	Field name	Acc	Reset	Description
1:0	wdq_ptrnbuf_dq_16	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.47 wdq_ptrnbuf_dq0_PB2_$1

$1=0..15

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_0	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_1	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_2	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_3	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_4	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_5	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_6	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_7	RW 0x0 Description1

6.20.48 wdq_ptrnbuf_dq8_PB2_0

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_8	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_9	RW 0x0 Description1

Bits	Field name	Acc Reset Description
5:4	wdq_ptrnbuf_dq_10	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_11	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_12	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_13	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_14	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_15	RW 0x0 Description1

6.20.49 wdq_ptrnbuf_dq16_PB2_0

Bits	Field name	Acc	Reset	Description
1:0	wdq_ptrnbuf_dq_16	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.50 wdq_ptrnbuf_dq0_PB3_$1

$1=0..15

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_0	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_1	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_2	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_3	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_4	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_5	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_6	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_7	RW 0x0 Description1

6.20.51 wdq_ptrnbuf_dq8_PB3_0

Bits	Field name	Acc Reset Description
1:0	wdq_ptrnbuf_dq_8	RW 0x0 Description1
3:2	wdq_ptrnbuf_dq_9	RW 0x0 Description1
5:4	wdq_ptrnbuf_dq_10	RW 0x0 Description1
7:6	wdq_ptrnbuf_dq_11	RW 0x0 Description1
9:8	wdq_ptrnbuf_dq_12	RW 0x0 Description1
11:10	wdq_ptrnbuf_dq_13	RW 0x0 Description1
13:12	wdq_ptrnbuf_dq_14	RW 0x0 Description1
15:14	wdq_ptrnbuf_dq_15	RW 0x0 Description1

6.20.52 wdq_ptrnbuf_dq16_PB3_0

Bits	Field name	Acc	Reset	Description
1:0	wdq_ptrnbuf_dq_16	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.53 wdq_ptrnbuf_wr_en_PB0_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
0	wdq_ptrnbuf_wr_en	RW	0x0	Description1
1	wdq_ptrnbuf_wr_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.54 wdq_ptrnbuf_wr_en_PB1_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
0	wdq_ptrnbuf_wr_en	RW	0x0	Description1
1	wdq_ptrnbuf_wr_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.55 wdq_ptrnbuf_wr_en_PB2_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
0	wdq_ptrnbuf_wr_en	RW	0x0	Description1
1	wdq_ptrnbuf_wr_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.56 wdq_ptrnbuf_wr_en_PB3_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
0	wdq_ptrnbuf_wr_en	RW	0x0	Description1
1	wdq_ptrnbuf_wr_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.57 rden_ptrnbuf_start_ptr_PB$1

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	rden_ptrnbuf_start_ptr	RW	0x0	Description1
15:8	rden_ptrnbuf_stop_ptr	RW	0x0	Description1

6.20.58 rden_ptrnbuf_rpt_cnt_PB$1

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	rden_ptrnbuf_rpt_cnt	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.59 rden_ptrnbuf_delay_PB0_0

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	rden_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.60 rden_ptrnbuf_delay_PB1_$1

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	rden_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.61 rden_ptrnbuf_delay_PB2_0

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	rden_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.62 rden_ptrnbuf_delay_PB3_0

$1=0..15

Bits	Field name	Acc	Reset	Description
4:0	rden_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.63 rden_ptrnbuf_rd_en_PB0_0

$1=0..15

Bits	Field name	Acc	Reset	Description
0	rden_ptrnbuf_rd_en	RW	0x0	Description1
1	rden_ptrnbuf_rd_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.64 rden_ptrnbuf_rd_en_PB1_0

$1=0..15

Bits	Field name	Acc	Reset	Description
0	rden_ptrnbuf_rd_en	RW	0x0	Description1
1	rden_ptrnbuf_rd_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.65 rden_ptrnbuf_rd_en_PB2_0

$1=0..15

Bits	Field name	Acc	Reset	Description
0	rden_ptrnbuf_rd_en	RW	0x0	Description1
1	rden_ptrnbuf_rd_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.66 rden_ptrnbuf_rd_en_PB3_0

$1=0..15

Bits	Field name	Acc	Reset	Description
0	rden_ptrnbuf_rd_en	RW	0x0	Description1
1	rden_ptrnbuf_rd_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.67 rdq_ptrnbuf_start_ptr_PB$1

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	rdq_ptrnbuf_start_ptr	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.68 rdq_ptrnbuf_cmpr_start_ptr_PB0

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	rdq_ptrnbuf_cmpr_start_p	trRW	0x0	Description1
12:8	rdq_ptrnbuf_cmpr_stop_pt	rRW	0x0	Description1
15:13	Reserved	RO	0x0	Reserved for future use

6.20.69 rdq_ptrnbuf_dq0_PB$2_$1

$1=0..15, $2=0..3

Bits	Field name	Acc Reset Description
1:0	rdq_ptrnbuf_dq_0	RO 0x0 Description1
3:2	rdq_ptrnbuf_dq_1	RO 0x0 Description1
5:4	rdq_ptrnbuf_dq_2	RO 0x0 Description1
7:6	rdq_ptrnbuf_dq_3	RO 0x0 Description1
9:8	rdq_ptrnbuf_dq_4	RO 0x0 Description1
11:10	rdq_ptrnbuf_dq_5	RO 0x0 Description1
13:12	rdq_ptrnbuf_dq_6	RO 0x0 Description1
15:14	rdq_ptrnbuf_dq_7	RO 0x0 Description1

6.20.70 rdq_ptrnbuf_dq8_PB0_$1

Bits	Field name	Acc Reset Description
1:0	rdq_ptrnbuf_dq_8	RO 0x0 Description1
3:2	rdq_ptrnbuf_dq_9	RO 0x0 Description1
5:4	rdq_ptrnbuf_dq_10	RO 0x0 Description1
7:6	rdq_ptrnbuf_dq_11	RO 0x0 Description1
9:8	rdq_ptrnbuf_dq_12	RO 0x0 Description1
11:10	rdq_ptrnbuf_dq_13	RO 0x0 Description1
13:12	rdq_ptrnbuf_dq_14	RO 0x0 Description1
15:14	rdq_ptrnbuf_dq_15	RO 0x0 Description1

6.20.71 rdq_ptrnbuf_dq16_PB0_$1

Bits	Field name	Acc	Reset	Description
1:0	rdq_ptrnbuf_dq_16	RO	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.72 rdq_ptrnbuf_cmpr_pf_0_PB$1

$1=0..3

Bits	Field name	Acc	Reset	Description
0	rdq_ptrnbuf_cmpr_pf_0	RO	0x0	Description1
1	rdq_ptrnbuf_cmpr_pf_1	RO	0x0	Description1
2	rdq_ptrnbuf_cmpr_pf_2	RO	0x0	Description1
3	rdq_ptrnbuf_cmpr_pf_3	RO	0x0	Description1
4	rdq_ptrnbuf_cmpr_pf_4	RO	0x0	Description1
5	rdq_ptrnbuf_cmpr_pf_5	RO	0x0	Description1
6	rdq_ptrnbuf_cmpr_pf_6	RO	0x0	Description1
7	rdq_ptrnbuf_cmpr_pf_7	RO	0x0	Description1
8	rdq_ptrnbuf_cmpr_pf_8	RO	0x0	Description1
9	rdq_ptrnbuf_cmpr_pf_9	RO	0x0	Description1
10	rdq_ptrnbuf_cmpr_pf_10	RO	0x0	Description1
11	rdq_ptrnbuf_cmpr_pf_11	RO	0x0	Description1
12	rdq_ptrnbuf_cmpr_pf_12	RO	0x0	Description1
13	rdq_ptrnbuf_cmpr_pf_13	RO	0x0	Description1
14	rdq_ptrnbuf_cmpr_pf_14	RO	0x0	Description1
15	rdq_ptrnbuf_cmpr_pf_15	RO	0x0	Description1

6.20.73 rdq_ptrnbuf_cmpr_pf_1_PB0

Bits	Field name	Acc	Reset	Description
0	rdq_ptrnbuf_cmpr_pf_16	RO	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.74 dq_prbs_seed_D$1_0

$1=0..3

Bits	Field name	Acc	Reset	Description
14:0	dq_prbs_seed	RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.75 dq_prbs_seed_D$1_1

Bits	Field name	Acc	Reset	Description
14:0	dq_prbs_seed	RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.76 dq_prbs_seed_D$1_2

Bits	Field name	Acc	Reset	Description
14:0	dq_prbs_seed	RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.77 dq_prbs_type_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
1:0	dq_prbs_type_0	RW	0x0	Description1
3:2	dq_prbs_type_1	RW	0x0	Description1
5:4	dq_prbs_type_2	RW	0x0	Description1
15:6	Reserved	RO	0x0	Reserved for future use

6.20.78 dq_prbsgen_id_dq0_D$1

$1=0..3

Bits	Field name	Acc Reset Description
1:0	dq_prbsgen_id_dq_0	RW 0x0 Description1
3:2	dq_prbsgen_id_dq_1	RW 0x0 Description1
5:4	dq_prbsgen_id_dq_2	RW 0x0 Description1
7:6	dq_prbsgen_id_dq_3	RW 0x0 Description1
9:8	dq_prbsgen_id_dq_4	RW 0x0 Description1
11:10	dq_prbsgen_id_dq_5	RW 0x0 Description1
13:12	dq_prbsgen_id_dq_6	RW 0x0 Description1
15:14	dq_prbsgen_id_dq_7	RW 0x0 Description1

6.20.79 dq_prbsgen_id_dq8_D$1

Bits	Field name	Acc	Reset	Description
1:0	dq_prbsgen_id_dq_8	RW	0x0	Description1
3:2	dq_prbsgen_id_dq_9	RW	0x0	Description1
5:4	dq_prbsgen_id_dq_10	RW	0x0	Description1
7:6	dq_prbsgen_id_dq_11	RW	0x0	Description1
9:8	dq_prbsgen_id_dq_12	RW	0x0	Description1
11:10	dq_prbsgen_id_dq_13	RW	0x0	Description1
13:12	dq_prbsgen_id_dq_14	RW	0x0	Description1
15:14	dq_prbsgen_id_dq_15	RW	0x0	Description1

6.20.80 dq_prbsgen_id_dq16_D$1

Bits	Field name	Acc	Reset	Description
1:0	dq_prbsgen_id_dq_16	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.81 dq_prbsgen_inject_error_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
0	dq_prbsgen_inject_error	WC	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.82 dq_prbscheck_fail_pin_id_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
4:0	dq_prbscheck_fail_pin_id	0R W	0x0	Description1
9:5	dq_prbscheck_fail_pin_id	1R W	0x0	Description1
14:10	dq_prbscheck_fail_pin_id	2R W	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.83 result_prbs_dq0_D$1

$1=0..3

Bits	Field name	Acc Reset Description
1:0	result_prbs_dq_0	RO 0x0 Description1
3:2	result_prbs_dq_1	RO 0x0 Description1
5:4	result_prbs_dq_2	RO 0x0 Description1
7:6	result_prbs_dq_3	RO 0x0 Description1
9:8	result_prbs_dq_4	RO 0x0 Description1
11:10	result_prbs_dq_5	RO 0x0 Description1
13:12	result_prbs_dq_6	RO 0x0 Description1
15:14	result_prbs_dq_7	RO 0x0 Description1

6.20.84 result_prbs_dq8_D$1

Bits	Field name	Acc	Reset	Description
1:0	result_prbs_dq_8	RO	0x0	Description1
3:2	result_prbs_dq_9	RO	0x0	Description1
5:4	result_prbs_dq_10	RO	0x0	Description1
7:6	result_prbs_dq_11	RO	0x0	Description1
9:8	result_prbs_dq_12	RO	0x0	Description1
11:10	result_prbs_dq_13	RO	0x0	Description1
13:12	result_prbs_dq_14	RO	0x0	Description1
15:14	result_prbs_dq_15	RO	0x0	Description1

6.20.85 result_prbs_dq16_D$1

Bits	Field name	Acc	Reset	Description
1:0	result_prbs_dq_16	RO	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.86 dq_prbs_error_count_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
15:0	dq_prbs_error_count	RO	0x0	Description1

6.20.87 dq_prbs_error_count_pin_D$2_$1

$1=0..2, $2=0..3

Bits	Field name	Acc	Reset	Description
15:0	dq_prbs_error_count_pin	RO	0x0	Description1

6.20.88 dfi_freq_ratio

Bits	Field name	Acc	Reset	Description
1:0	dfi_freq_ratio	RO	0x0	Description1
3:2	dfi_freq_fsp	RO	0x0	Description1
8:4	dfi_frequency	RO	0x0	Description1
15:9	Reserved	RO	0x0	Reserved for future use

6.20.89 dfi_init_complete

Bits	Field name	Acc	Reset	Description
0	dfi_init_complete	RW	0x0	Description1

Bits	Field name	Acc	Reset	Description
15:1	Reserved	RO	0x0	Reserved for future use

6.20.90 dfi_ctrlupd_ack

Bits	Field name	Acc	Reset	Description
0	dfi_ctrlupd_ack	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.91 dfi_phyupd_req

Bits	Field name	Acc	Reset	Description
0	dfi_phyupd_req	RW	0x0	Description1
2:1	dfi_phyupd_type	RW	0x0	Description1
15:3	Reserved	RO	0x0	Reserved for future use

6.20.92 dfi_phymstr_req

Bits	Field name	Acc	Reset	Description
0	dfi_phymstr_req	RW	0x0	Description1
2:1	dfi_phymstr_type	RW	0x0	Description1
3	dfi_phymstr_state_sel	RW	0x0	Description1
5:4	dfi_phymstr_cs_state	RW	0x0	Description1
15:6	Reserved	RO	0x0	Reserved for future use

6.20.93 dfi_init_start

Bits	Field name	Acc	Reset	Description
0	dfi_init_start	RO	0x0	Description1
1	dfi_ctrlupd_req	RO	0x0	Description1
2	dfi_phyupd_ack	RO	0x0	Description1
3	dfi_phymstr_ack	RO	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.94 dfi_lp_ctrl_req

Bits	Field name	Acc	Reset	Description
0	dfi_lp_ctrl_req	RO	0x0	Description1
4:1	dfi_lp_ctrl_wakeup	RO	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.95 dfi_lp_data_req

Bits	Field name	Acc	Reset	Description
0	dfi_lp_data_req	RO	0x0	Description1
4:1	dfi_lp_data_wakeup	RO	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.96 dfi_ctrlmsg_req

Bits	Field name	Acc	Reset	Description
0	dfi_ctrlmsg_req	RO	0x0	Description1
8:1	dfi_ctrlmsg	RO	0x0	Description1
15:9	Reserved	RO	0x0	Reserved for future use

6.20.97 dfi_ctrlmsg_data

Bits	Field name	Acc	Reset	Description
15:0	dfi_ctrlmsg_data	RO	0x0	Description1

6.20.98 dfi_disconnect_error

Bits	Field name	Acc	Reset	Description
0	dfi_disconnect_error	RO	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.99 dfi_lp_ctrl_ack

Bits	Field name	Acc	Reset	Description
0	dfi_lp_ctrl_ack	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.100 dfi_lp_data_ack

Bits	Field name	Acc	Reset	Description
0	dfi_lp_data_ack	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.101 dfi_error

Bits	Field name	Acc	Reset	Description
0	dfi_error	RW	0x0	Description1
4:1	dfi_error_info	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.102 dfi_ctrlmsg_ack

Bits	Field name	Acc	Reset	Description
0	dfi_ctrlmsg_ack	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.103 intr_dfi_init_start

Bits	Field name	Acc	Reset	Description
0	intr_dfi_init_start	W1C	0x0	Description1
1	intr_dfi_ctrlupd_req	W1C	0x0	Description1
2	intr_dfi_phyupd_ack	W1C	0x0	Description1
3	intr_dfi_phymstr_ack	W1C	0x0	Description1
4	intr_dfi_lp_ctrl_req_l2h	W1C	0x0	Description1
5	intr_dfi_lp_ctrl_req_h2l	W1C	0x0	Description1
6	intr_dfi_lp_ctrl_wakeup_c	hng W1C	0x0	Description1
7	intr_dfi_lp_data_req_l2h	W1C	0x0	Description1
8	intr_dfi_lp_data_req_h2l	W1C	0x0	Description1
9	intr_dfi_lp_data_wakeup_	chng W1C	0x0	Description1
10	intr_dfi_ctrlmsg_req	W1C	0x0	Description1
11	intr_dfi_disconnect_error	W1C	0x0	Description1
12	intr_need_to_retrain_0	W1C	0x0	Description1
13	intr_need_to_retrain_1	W1C	0x0	Description1
14	intr_need_to_retrain_2	W1C	0x0	Description1
15	intr_need_to_retrain_3	W1C	0x0	Description1

6.20.104 intr_dfi_init_start_enable

Bits	Field name	Acc	Reset	Description
0	intr_dfi_init_start_enable	RW	0x1	Description1
1	intr_dfi_ctrlupd_req_enabl	eRW	0x1	Description1
2	intr_dfi_phyupd_ack_enab	leRW	0x1	Description1
3	intr_dfi_phymstr_ack_ena	ble RW	0x1	Description1
4	intr_dfi_lp_ctrl_req_l2h_en	able RW	0x1	Description1
5	intr_dfi_lp_ctrl_req_h2l_en	able RW	0x1	Description1
6	intr_dfi_lp_ctrl_wakeup_c	hng_enaRW	ble 0x1	Description1
7	intr_dfi_lp_data_req_l2h_	nable RW	0x1	Description1

Bits	Field name	Acc	Reset	Description
8	intr_dfi_lp_data_req_h2l_	nable RW	0x1	Description1
9	intr_dfi_lp_data_wakeup_	chng_enRW	able 0x1	Description1
10	intr_dfi_ctrlmsg_req_enab	leRW	0x1	Description1
11	intr_dfi_disconnect_error_	enable RW	0x1	Description1
12	intr_need_to_retrain_enab	le_0 RW	0x0	Description1
13	intr_need_to_retrain_enab	le_1 RW	0x0	Description1
14	intr_need_to_retrain_enab	le_2 RW	0x0	Description1
15	intr_need_to_retrain_enab	le_3 RW	0x0	Description1

6.20.105 dqs_pattern

Bits	Field name	Acc	Reset	Description
1:0	dqs_pattern	RW	0x1	Description1
3:2	ck_pattern	RW	0x1	Description1
4	training_en	RW	0x1	Description1
6:5	cur_fsp	RW	0x0	Description1
7	rdfifo_low_lat	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.106 wdqs_post_amble_1p5

Bits	Field name	Acc	Reset	Description
0	wdqs_post_amble_1p5	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.107 cc_ptrn_en

Bits	Field name	Acc	Reset	Description
0	cc_ptrn_en	RW	0x0	Description1
1	ck_hv_ptrn_en	RW	0x0	Description1
2	dqwr_data_ptrn_en_0	RW	0x0	Description1
3	dqwr_data_ptrn_en_1	RW	0x0	Description1
4	dqwr_data_ptrn_en_2	RW	0x0	Description1
5	dqwr_data_ptrn_en_3	RW	0x0	Description1
6	dqrd_en_ptrn_en_0	RW	0x0	Description1
7	dqrd_en_ptrn_en_1	RW	0x0	Description1
8	dqrd_en_ptrn_en_2	RW	0x0	Description1
9	dqrd_en_ptrn_en_3	RW	0x0	Description1

Bits	Field name	Acc	Reset	Description
10	dqrd_data_ptrn_en_0	RW	0x0	Description1
11	dqrd_data_ptrn_en_1	RW	0x0	Description1
12	dqrd_data_ptrn_en_2	RW	0x0	Description1
13	dqrd_data_ptrn_en_3	RW	0x0	Description1
15:14	Reserved	RO	0x0	Reserved for future use

6.20.108 gbl_ptrn_en

Bits	Field name	Acc	Reset	Description
0	gbl_ptrn_en	WC	0x0	Description1
1	gbl_inf_stop	WC	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.109 cc_prbs_gen_en

Bits	Field name	Acc	Reset	Description
0	cc_prbs_gen_en	RW	0x0	Description1
1	hv_prbs_gen_en	RW	0x0	Description1
2	ck_prbs_gen_en	RW	0x0	Description1
3	cc_prbs_check_en	RW	0x0	Description1
4	hv_prbs_check_en	RW	0x0	Description1
5	ck_prbs_check_en	RW	0x0	Description1
6	cc_prbs_result_reset	RW	0x0	Description1
7	hv_prbs_result_reset	RW	0x0	Description1
8	ck_prbs_result_reset	RW	0x0	Description1
15:9	Reserved	RO	0x0	Reserved for future use
8	ck_prbs_result_reset	RW	0x0	Description1
15:9	Reserved	RO	0x0	Reserved for future use

6.20.110 dq_prbs_gen_en

Bits	Field name	Acc	Reset	Description
0	dq_prbs_gen_en_0	RW	0x0	Description1
1	dq_prbs_gen_en_1	RW	0x0	Description1
2	dq_prbs_gen_en_2	RW	0x0	Description1
3	dq_prbs_gen_en_3	RW	0x0	Description1
4	dq_prbs_check_en_0	RW	0x0	Description1

Bits	Field name	Acc	Reset	Description
5	dq_prbs_check_en_1	RW	0x0	Description1
6	dq_prbs_check_en_2	RW	0x0	Description1
7	dq_prbs_check_en_3	RW	0x0	Description1
8	dq_prbs_result_reset_0	RW	0x0	Description1
9	dq_prbs_result_reset_1	RW	0x0	Description1
10	dq_prbs_result_reset_2	RW	0x0	Description1
11	dq_prbs_result_reset_3	RW	0x0	Description1
15:12	Reserved	RO	0x0	Reserved for future use

6.20.111 cc_pipe_dly_ca_R0F0

Bits	Field name	Acc	Reset	Description
3:0	cc_pipe_dly_ca	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.112 cc_pipe_dly_ca_R0F1

Bits	Field name	Acc	Reset	Description
3:0	cc_pipe_dly_ca	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.113 hv_pipe_dly_R0F0

Bits	Field name	Acc	Reset	Description
3:0	hv_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.114 hv_pipe_dly_R0F1

Bits	Field name	Acc	Reset	Description
3:0	hv_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.115 cc_pipe_dly_cs_R0F0

Bits	Field name	Acc	Reset	Description
3:0	cc_pipe_dly_cs	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.116 cc_pipe_dly_cs_R0F1

Bits	Field name	Acc	Reset	Description
3:0	cc_pipe_dly_cs	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.117 cc_rank_sel_pipe_dly_R0F0

Bits	Field name	Acc	Reset	Description
3:0	cc_rank_sel_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.118 cc_rank_sel_pipe_dly_R0F1

Bits	Field name	Acc	Reset	Description
3:0	cc_rank_sel_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.119 ck_pipe_dly_F0

Bits	Field name	Acc	Reset	Description
3:0	ck_pipe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.120 ck_pipe_dly_F1

Bits	Field name	Acc	Reset	Description
3:0	ck_pipe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.121 cchvck_cross_sel_F0

Bits	Field name	Acc	Reset	Description
0	cchvck_cross_sel	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.122 cchvck_cross_sel_F1

Bits	Field name	Acc	Reset	Description
0	cchvck_cross_sel	RW	0x0	Description1
15:1	Reserved	RO	0x0	Reserved for future use

6.20.123 cc_rx_crosssel

Bits	Field name	Acc	Reset	Description
0	cc_rx_crosssel	RW	0x0	Description1
2:1	cc_rx_nui	RW	0x0	Description1
3	cc_rx_rfirst	RW	0x0	Description1
4	cc_rx_eo_and	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.124 hv_rx_crosssel

Bits	Field name	Acc	Reset	Description
0	hv_rx_crosssel	RW	0x0	Description1
2:1	hv_rx_nui	RW	0x0	Description1
3	hv_rx_rfirst	RW	0x0	Description1
4	hv_rx_eo_and	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.125 ck_rx_crosssel

Bits	Field name	Acc	Reset	Description
0	ck_rx_crosssel	RW	0x0	Description1
2:1	ck_rx_nui	RW	0x0	Description1
3	ck_rx_rfirst	RW	0x0	Description1
4	ck_rx_eo_and	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.126 ld_cnt_cmd

Bits	Field name	Acc	Reset	Description
2:0	ld_cnt_cmd	RW	0x7	Description1
3	rd_ld_cnt_reset_n	RW	0x1	Description1
5:4	ser_reset_stagesel_cmd	RW	0x0	Description1
15:6	Reserved	RO	0x0	Reserved for future use

6.20.127 ca_dfi_mapping_$1

$1=0..31

Bits	Field name	Acc	Reset	Description
3:0	ca_dfi_mapping_0	RW	0x0	Description1
7:4	ca_dfi_mapping_1	RW	0x1	Description1

Bits	Field name	Acc	Reset	Description
11:8	ca_dfi_mapping_2	RW	0x2	Description1
15:12	ca_dfi_mapping_3	RW	0x3	Description1

cc_ptrnbuf_ca0_$1

$1=0..31

Bits	Field name	Acc	Reset	Description
1:0	cc_ptrnbuf_ca_0	RW	0x0	Description1
3:2	cc_ptrnbuf_ca_1	RW	0x0	Description1
5:4	cc_ptrnbuf_ca_2	RW	0x0	Description1
7:6	cc_ptrnbuf_ca_3	RW	0x0	Description1
9:8	cc_ptrnbuf_ca_4	RW	0x0	Description1
11:10	cc_ptrnbuf_ca_5	RW	0x0	Description1
13:12	cc_ptrnbuf_ca_6	RW	0x0	Description1
15:14	cc_ptrnbuf_ca_7	RW	0x0	Description1

6.20.128 cc_ptrnbuf_ca8_$1

Bits	Field name	Acc	Reset	Description
1:0	cc_ptrnbuf_ca_8	RW	0x0	Description1
3:2	cc_ptrnbuf_ca_9	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.129 cc_ptrnbuf_cs_$1

$1=0..31

Bits	Field name	Acc	Reset	Description
1:0	cc_ptrnbuf_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.130 ckhv_ptrnbuf_start_ptr

Bits	Field name	Acc	Reset	Description
7:0	ckhv_ptrnbuf_start_ptr	RW	0x0	Description1
15:8	ckhv_ptrnbuf_stop_ptr	RW	0x0	Description1

6.20.131 ckhv_ptrnbuf_rpt_cnt

Bits	Field name	Acc	Reset	Description
7:0	ckhv_ptrnbuf_rpt_cnt	RW	0x0	Description1

Bits	Field name	Acc	Reset	Description
15:8	Reserved	RO	0x0	Reserved for future use

6.20.132 ckhv_ptrnbuf_delay_$1

$1=0..3

Bits	Field name	Acc	Reset	Description
4:0	ckhv_ptrnbuf_delay	RW	0x0	Description1
15:5	Reserved	RO	0x0	Reserved for future use

6.20.133 ckhv_ptrnbuf_cke_$3

$1=0..3

Bits	Field name	Acc	Reset	Description
1:0	ckhv_ptrnbuf_cke_0	RW	0x0	Description1
3:2	ckhv_ptrnbuf_reset	RW	0x0	Description1
5:4	ckhv_ptrnbuf_ckdis	RW	0x0	Description1
15:6	Reserved	RO	0x0	Reserved for future use

6.20.134 ch_prbs_seed_$1

$1=0..2

Bits	Field name	Acc	Reset	Description
14:0	ch_prbs_seed	RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.135 ck_prbs_seed

Bits	Field name	Acc	Reset	Description
14:0	ck_prbs_seed	RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.136 ch_prbs_type

Bits	Field name	Acc	Reset	Description
1:0	ch_prbs_type_0	RW	0x0	Description1
3:2	ch_prbs_type_1	RW	0x0	Description1
5:4	ch_prbs_type_2	RW	0x0	Description1
7:6	ck_prbs_type	RW	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.137 cc_prbsgen_id_ca0

Bits	Field name	Acc Reset Description
1:0	cc_prbsgen_id_ca_0	RW 0x0 Description1
3:2	cc_prbsgen_id_ca_1	RW 0x0 Description1
5:4	cc_prbsgen_id_ca_2	RW 0x0 Description1
7:6	cc_prbsgen_id_ca_3	RW 0x0 Description1
9:8	cc_prbsgen_id_ca_4	RW 0x0 Description1
11:10	cc_prbsgen_id_ca_5	RW 0x0 Description1
13:12	cc_prbsgen_id_ca_6	RW 0x0 Description1
15:14	cc_prbsgen_id_ca_7	RW 0x0 Description1

6.20.138 cc_prbsgen_id_ca8

Bits	Field name	Acc	Reset	Description
1:0	cc_prbsgen_id_ca_8	RW	0x0	Description1
3:2	cc_prbsgen_id_ca_9	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.139 cc_prbsgen_id_cs

Bits	Field name	Acc	Reset	Description
1:0	cc_prbsgen_id_cs_0	RW	0x0	Description1
15:2	Reserved	RO	0x0	Reserved for future use

6.20.140 hv_prbsgen_id_cke

Bits	Field name	Acc	Reset	Description
1:0	hv_prbsgen_id_cke_0	RW	0x0	Description1
3:2	hv_prbsgen_id_resetn	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.141 cc_prbsgen_inject_error

Bits	Field name	Acc	Reset	Description
0	cc_prbsgen_inject_error	WC	0x0	Description1
1	hv_prbsgen_inject_error	WC	0x0	Description1
2	ck_prbsgen_inject_error	WC	0x0	Description1
15:3	Reserved	RO	0x0	Reserved for future use

6.20.142 result_prbs_ca0

Bits	Field name	Acc Reset Description
1:0	result_prbs_ca_0	RO 0x0 Description1
3:2	result_prbs_ca_1	RO 0x0 Description1
5:4	result_prbs_ca_2	RO 0x0 Description1
7:6	result_prbs_ca_3	RO 0x0 Description1
9:8	result_prbs_ca_4	RO 0x0 Description1
11:10	result_prbs_ca_5	RO 0x0 Description1
13:12	result_prbs_ca_6	RO 0x0 Description1
15:14	result_prbs_ca_7	RO 0x0 Description1

6.20.143 result_prbs_ca8

Bits	Field name	Acc	Reset	Description
1:0	result_prbs_ca_8	RO	0x0	Description1
3:2	result_prbs_ca_9	RO	0x0	Description1
5:4	result_prbs_cs_0	RO	0x0	Description1
15:6	Reserved	RO	0x0	Reserved for future use

6.20.144 result_prbs_cke

Bits	Field name	Acc	Reset	Description
1:0	result_prbs_cke_0	RO	0x0	Description1
3:2	result_prbs_resetn	RO	0x0	Description1
5:4	result_prbs_ck	RO	0x0	Description1
7:6	result_prbs_ckb	RO	0x0	Description1
15:8	Reserved	RO	0x0	Reserved for future use

6.20.145 chc_prbs_error_count

Bits	Field name	Acc	Reset	Description
9:0	chc_prbs_error_count	RO	0x0	Description1
15:10	Reserved	RO	0x0	Reserved for future use

6.20.146 cc_prbs_error_count_chc_$1

$1=0..2

Bits	Field name	Acc	Reset	Description
9:0	cc_prbs_error_count_chc	RO	0x0	Description1
15:10	Reserved	RO	0x0	Reserved for future use

6.20.147 chc_prbscheck_fail_pin_id

Bits	Field name	Acc	Reset	Description
4:0	chc_prbscheck_fail_pin_id	_0 RW	0x0	Description1
9:5	chc_prbscheck_fail_pin_id	_1 RW	0x0	Description1
14:10	chc_prbscheck_fail_pin_id	_2 RW	0x0	Description1
15	Reserved	RO	0x0	Reserved for future use

6.20.148 acsr_addr_$1

$1=0..14

Bits	Field name	Acc	Reset	Description
15:0	acsr_addr_0	RW	0x0	Description1

6.20.149 acsr_wdata_$1

Bits	Field name	Acc	Reset	Description
15:0	acsr_wdata_0	RW	0x0	Description1

6.20.150 acsr_rdata_$1

Bits	Field name	Acc	Reset	Description
15:0	acsr_rdata_0	RO	0x0	Description1

6.20.151 retrain_cmd_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
0	retrain_freeze	RW	0x0	All register fields that require synchronization between the APB and the reference clock domains (all the other CSR control bits and all the CSR status bits) are double buffered with two sets of registers -- one in the APB clock domain and the other in the reference clock domain. This bit disables the load enable for the reference-clocked register of the register pair -- the first rank for status input bits and the second rank for control output bits. Software must assert this bit prior to writing or reading any of these asynchronous fields. Active high. The assertion of freeze requires a maximum of 3 reference clock periods. The APB bus master must allow at least this mµCh time between the assertion of freeze and the subsequent writing or reading of 'frozen' CSR bits. This is typically easily met by the ratio of the APB to reference clock frequencies and the APB transaction repetition rate, but it is not guaranteed. This bit applies only to the retrain detector CSRs.
1	retrain_rst_n	RW	0x0	Reset for all retrain detector logic. Active low. This is the only control bit that is synchronized in the normal way (with a multi-rank synchronizer)(in addition to participating in the freeze methodology). There are 2 reasons for this: 1. Reset can bypass the freeze methodology (asserting/deasserting reset can be done with 1 CSR write instead of 3). 2. Reset can be deasserted in the same freeze group as other control signals. There is a required 1 clock delay between updating ref_per_mult (for example) and deasserting reset. The synchronizer provides this 1 clock delay (plus more). This is the 1st of 3 ways to clear both the {csr,int}_need_to_retrain outputs.
15:2	Reserved	RO	0x0	Reserved for future use

6.20.152 retrain_ref_per_mult_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
15:0	retrain_ref_per_mult	RW	0x1	There is a frequency detector which monitors the frequency of the oscillator (OSC) clock output. It counts the number of positive (rising) OSC clock edges which occur in a known period of time. This known period of time is determined by a given number of reference clock periods. This number of periods is contained in the current field and is known as the reference clock period multiplier. Minimum value is 1. Maximum value is 65,536 and is encoded as 0 (ie 0x0000 is interpreted as 0x1_0000).

6.20.153 retrain_osc_cnt_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
15:0	osc_cnt	RO	0x0	The retrain detector determines if the operating point (voltage and temperature) has changed significantly from the last time the analog IP was calibrated (trained) and, therefore, indicates this IP needs to be retrained. It does this by monitoring the frequency of a local ring oscillator (OSC) and detecting when this frequency has changed significantly. This field holds the current value of the OSC frequency (the count of the number of positive (rising) OSC clock edges observed during the previous measurement period). The counter saturates at its maximum value (0xFFFF). If this value is ever observed, the counter has probably overflowed. This value must be ignored and the frequency measurement must be redone with a smaller ref_per_mult.

6.20.154 retrain_min_osc_cnt_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
15:0	min_osc_cnt	RW	0x0	After the initial oscillator frequency measurement following an analog IP calibration, the current osc_cnt is compared to this initial measurement to determine if it has changed significantly. This field contains the minimum value of osc_cnt for the frequency change to be considered insignificant.

retrain_max_osc_cnt_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
15:0	max_osc_cnt	RW	0xffff	After the initial oscillator frequency measurement following an analog IP calibration, the current osc_cnt is compared to this initial measurement to determine if it has changed significantly. This field contains the maximum value of osc_cnt for the frequency change to be considered insignificant.

6.20.155 retrain_retrain_cmd_D$1

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	retrain_cycles	RW	0x0	The number of sequential measurement periods in which the current oscillator frequency (osc_cnt) must be significantly different than the initial value (ie either less than min_osc_cnt or greater than max_osc_cnt) in order to request a need for retraining. A value of 0 disables the ability to generate a need for retraining request. This is the 2nd of 3 ways to clear both the {csr,int}_need_to_retrain outputs.
8	clr_retrain	RW	0x0	Clear the retrain_cycle counter. Active high. This is the 3rd of 3 ways to clear both the {csr,int}_need_to_retrain outputs.
15:9	Reserved	RO	0x0	Reserved for future use

6.20.156 retrain_retrain_stat_D$3

$1=0..3

Bits	Field name	Acc	Reset	Description
7:0	retrain_cnt	RO	0x0	The current number of sequential measurement periods in which the current oscillator frequency has been observed to be significantly different than the initial value. When (retrain_cnt >= retrain_cycles), a need for retraining request will be generated by asserting both the {csr,int}_need_to_retrain outputs.
8	need_to_retrain	RO	0x0	When asserted, a need to retrain the analog IP is being requested. Active high.
15:9	Reserved	RO	0x0	Reserved for future use

6.20.157 ch1_wdq_pipe_dly_D$1R$2F$3

$1=0..3, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdq_pipe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.158 ch1_wdqs_pipe_dly_D$1R$2F$3

$1=0..3, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.159 ch1_wdq_pipe_oe_dly_D0R0F1

$1=0..3, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdq_pipe_oe_dly	RW	0x4	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.160 ch1_wdqs_pipe_oe_dly_D0R0F0

$1=0..3, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	wdqs_pipe_oe_dly	RW	0x1	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.20.161 ch1_write_en_early_pipe_dly_D$0R$1F$2

$1=0..3, $3=0..1

Bits	Field name	Acc	Reset	Description
3:0	write_en_early_pipe_dly	RW	0x0	Description1
15:4	Reserved	RO	0x0	Reserved for future use

6.21 FLC1 Registers

Purpose This register provides debug features to the FLC1. Usage constrains Register must be accessed in secure mode.

6.21.1 FLC_Debug_Control

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30	prefetch_en	RW	0x0	0x0 Disable Prefetch to FLC1 0x1 Enable Prefetch to FLC1
29	precise_Interrupt_reporting_enable	RW	0x0	• 0x1 - Enabled. FLC will hold until Interrupt is cleared to make sure SW has capture all the interrupt information. • 0x0 - Disabled FLC will keep running and older data will get overwritten if subsequent interrupt comes before the current interrupt is cleared. Note: this bit is only used for default imprecise interrupts: Linefill Done, Eviction Occur and Hash-tag Collision Occur Notification.
28:26	RSVD	RO	0x0	Reserved
25	CCF_Enable	RW	0x0	Critical Chunk First Wrapper Order enable
24	CWF_lineFill_Enable	RW	0x1	Critical Word First (CWF) LineFill performance feature enable.
23	SRAM_SLP_Mode_Enable	RW	0x0	• 0x0 Disable SRAM Sleep Mode • 0x1 Enable SRAM Sleep Mode Note: SLP and DSLP mode cannot be enabled at the same time
22	SRAM_DSLP_Mode_Enable	RW	0x0	• 0x0 Disable SRAM Deep Sleep Mode • 0x1 Enable SRAM Deep Sleep Mode Note: SLP and DSLP mode cannot be enabled at the same time
23:17	RSVD	RO	0x0	Reserved
16	L1_Tag_Hit_Disable	RW	0x0	• 0x0 Normal operation. • 0x1 Mask L1 tag hit to zero for debug purpose.
15:3	RSVD	RO	0x0	Reserved
2:1	Forced_Cache_Entry	RW	0x0	Specify the number of Cacheline when FLC_Debug is enabled. • 0x1 32 • 0x2 64 • 0x3 128
0	FLC_Cache_Debug	RW	0x0	• 0x0 Normal operation. • 0x1 Use specified number of entries.

6.21.2 FLC_Mnt_Req

Purpose: This register provides interface for Cache maintenance operations. See Chapter 5 for description and programming procedure of each operation.

Usage constrains: Program the register fields to issue the corresponding maintenance operations, only one operation can be processed at a time.

Bits	Field Name	Type	Reset	Description
31:26	RSVD	RO	0x0	Reserved
25:16	req_line_count	WO	0x0	Number of cacheline to perform for Allocate/Invalidate/Clean/Lock. 0 = 1 cache line, up to 1024 cache line.
15	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
14	use_tid	WO	0x0	• 0x1 - Use TID+PA for allocation or use TID instead of PA for all other operations. TID field tid_sel should be set first. • 0x0 - Do not use TID for the operation.
13	use_cache_index	WO	0x0	Use cache index instead of physical address for the supported operations.
12	unlock_enb	WO	0x0	Do unlock instead of lock for lock_request.
11	lock_req	WO	0x0	Request to lock/unlock an entry to the specified address/cache index.
10	clean_req	WO	0x0	Request to clean the dirty cacheline of the specified address.
9	clear_dirty_req	WO	0x0	Request to clear the dirty bit of the specified address, dirty granularity is 4KB.
8	chk_state_req	WO	0x0	Request to check if the specified address is hit/miss, locked/unlocked.
7	clean_all	WO	0x0	Request to clean all the dirty cacheline.
6	invld_all	WO	0x0	Invalid/clean all cache lines, fields use_tid, use_cache_index and req_line_count is ignored when this bit is set.
5	invld_clean	WO	0x0	Clean operation will be done before invalidation.
4	invld_req	WO	0x0	Request to invalidate the specified cache line.
3	allocate_fill_zero	WO	0x0	Fill cacheline content to zero after Allocation.
2	allocate_linefill	WO	0x0	Do Linefill after allocation.
1	allocate_lock	WO	0x0	Lock the cache line after allocation.
0	allocate_req	WO	0x0	Request to allocate a cache line for the specified address.

6.21.3 FLC_Mnt_Addr

Purpose: This register specifies the address bits [31:0] for maintenance operations.

Usage constrains: Use physical address if field use_cache_index in FLC_Mnt_Req is not set, use cache index if field use_cache_index in FLC_Mnt_Req is asserted.

Bits	Field Name	Type	Reset	Description
31:0	mnt_addr	RW	0x0	FLC Maintenance Address [31:0]

6.21.4 FLC_Mnt_Addr_H

Purpose: This register specifies the address bits [63:32] for maintenance operations.

Usage constrains: Use physical address if field use_cache_index in FLC_Mnt_Req is not set, use cache index if field use_cache_index in FLC_Mnt_Req is asserted.

Bits	Field Name	Type	Reset	Description
31:0	mnt_addr_h	RW	0x0	FLC Maintenance Address [63:32]

6.21.5 FLC_Hash_Collide_Req_Addr

Purpose: This register specifies the address bits [31:0] of the missed read/write that triggers allocation collision.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no hash collision.

Bits	Field Name	Type	Reset	Description
31:0	interupt_addr	RO	0x0	FLC hash tag collision interrupt request address [31:0]

6.21.6 FLC_Hash_Collide_Req_Addr_H

Purpose: This register specifies the address bits [63:32] of the missed read/write that triggers allocation collision.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no hash collision.

Bits	Field Name	Type	Reset	Description
31:0	interupt_addr_h	RO	0x0	FLC hash tag collision interrupt request address [63:32]

6.21.7 FLC_Hash_Collide_Addr_0

Purpose: This register specifies the address bits [31:0] of the Hash table 0 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 0.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_0	RO	0x0	FLC hash tag collision interrupt address_0 [31:0]

6.21.8 FLC_Hash_Collide_Addr_H_0

Purpose: This register specifies the address bits [63:32] of the Hash table 0 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 0.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_h_0	RO	0x0	FLC hash tag collision interrupt address_0 [63:32]

6.21.9 FLC_Hash_Collide_Addr_1

Purpose: This register specifies the address bits [31:0] of the Hash table 1 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 1.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_1	RO	0x0	FLC hash tag collision interrupt address_1 [31:0]

6.21.10 FLC_Hash_Collide_Addr_H_1

Purpose: This register specifies the address bits [63:32] of the Hash table 1 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 1.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_h_1	RO	0x0	FLC hash tag collision interrupt address_1 [63:32]

6.21.11 FLC_Hash_Collide_Addr_2

Purpose: This register specifies the address bits [31:0] of the Hash table 2 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 2.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_2	RO	0x0	FLC hash tag collision interrupt address_2 [31:0]

6.21.12 FLC_Hash_Collide_Addr_H_2

Purpose: This register specifies the address bits [63:32] of the Hash table 2 entry that is collided with the missed read/write.

Usage constrains: Check this register when hash collision occurs. Value is un-defined if there is no collision in Hash table 2.

Bits	Field Name	Type	Reset	Description
31:0	collide_addr_h_2	RO	0x0	FLC hash tag collision interrupt address_2 [63:32]

6.21.13 FLC_Mnt_Status

Purpose: This register provides the execution result of maintenance operations.

Usage constrains: Check this register after maintenance done interrupt received.

Bits	Field Name	Type	Reset	Description
31:15	RSVD	RO	0x0	Reserved
14	err_all_dirty	RO	0x0	Maintenance operation unsμCcessful due to all lines are dirty, SW needs to retry.
13	err_collision	RO	0x0	Maintenance operation unsμCcessful due to collision, SW needs to retry.
12:8	chk_state_tid	RO	0x0	Check state tid [4:0].
7	lock_done	RO	0x0	Lock operation done.
6	clean_done	RO	0x0	Clean operation done.
5	clear_dirty_done	RO	0x0	Clear dirty bits operation done.
4	chk_state_hit	RO	0x0	Check state hit status 0x0 Miss 0x1 Hit
3	chk_state_lock	RO	0x0	Check state lock status 0x0 Unlocked 0x1 Locked
2	chk_state_done	RO	0x0	Check state operation done.
1	invld_done	RO	0x0	Invalidate operation done.
0	alloc_done	RO	0x0	Allocate operation done.

6.21.14 FLC_Mnt_Lock_Status

Purpose: This register indicates the number of unlocked cache lines in FLC.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:16	RSVD	RO	0x0	Reserved
15:0	unlocked_line_count	RO	0x8000	Unlocked cache line count

6.21.15 FLC_Mnt_Chk_Status_Addr

Purpose: This register indicates bits [31:0] of the physical address or cache index of the cache line targeted by check state operation.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	chk_st_addr	RO	0x0	Return PA if check status via CA, or return CA if check status via PA

6.21.16 FLC_Mnt_Chk_Status_Addr_H

Purpose: This register indicates bits [63:32] of the physical address or cache index of the cache line targeted by check state operation.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	chk_st_addr_h	RO	0x0	Return PA if check status via CA, or return CA if check status via PA

6.21.17 FLC_Mnt_Collide_Addr_0

Purpose: This register specifies the address bits [31:0] of the Hash table 0 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 0.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_0	RO	0x0	FLC maintenance collision address_0 [31:0]

6.21.18 FLC_Mnt_Collide_Addr_H_0

Purpose: This register specifies the address bits [63:32] of the Hash table 0 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 0.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_h_0	RO	0x0	FLC maintenance collision address_0 [63:32]

6.21.19 FLC_Mnt_Collide_Addr_1

Purpose: This register specifies the address bits [31:0] of the Hash table 1 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 1.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_1	RO	0x0	FLC maintenance collision address_1 [31:0]

6.21.20 FLC_Mnt_Collide_Addr_H_1

Purpose: This register specifies the address bits [63:32] of the Hash table 1 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 1.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_h_1	RO	0x0	FLC maintenance collision address_1 [63:32]

6.21.21 FLC_Mnt_Collide_Addr_2

Purpose: This register specifies the address bits [31:0] of the Hash table 2 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 2.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_2	RO	0x0	FLC maintenance collision address_2 [31:0]

6.21.22 FLC_Mnt_Collide_Addr_H_2

Purpose: This register specifies the address bits [63:32] of the Hash table 2 entry that is collided with the maintenance allocation operation.

Usage constrains: Check this register when maintenance hash collision occurs. Value is un-defined if there is no collision in Hash table 2.

Bits	Field Name	Type	Reset	Description
31:0	mnt_collide_addr_h_2	RO	0x0	FLC maintenance collision address_2 [63:32]

6.21.23 FLC_Hash_Collide_Info

Purpose: This register provides the lock and dirty status of the collided hash table entries.

Usage constrains: Check this register when hash collision occurs.

Bits	Field Name	Type	Reset	Description
31:12	RSVD	RO	0x0	Reserved
11:10	mnt_collide_addr_0_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked
9:8	mnt_collide_addr_1_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked

Bits	Field Name	Type	Reset	Description
7:6	mnt_collide_addr_2_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked
5:4	collide_addr_0_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked
3:2	collide_addr_1_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked
1:0	collide_addr_2_status	RO	0x0	0x2 dirty but not locked 0x1 clean but locked 0x3 dirty and locked

6.21.24 FLC_Mnt_Fail_Req_Addr

Purpose: This register specifies the address bits [31:0] of the unsμCcessful maintenance allocation.

Usage constrains: Check this register when maintenance operation fails.

Bits	Field Name	Type	Reset	Description
31:0	mnt_fail_addr	RO	0x0	FLC maintenance allocation fail request address [31:0]

6.21.25 FLC_Mnt_Fail_Req_Addr_H

Purpose: This register specifies the address bits [63:32] of the unsμCcessful maintenance allocation.

Usage constrains: Check this register when maintenance operation fails.

Bits	Field Name	Type	Reset	Description
31:0	mnt_fail_addr_h	RO	0x0	FLC maintenance allocation fail request address [63:32]

6.21.26 FLC_LF_Err_Int_Addr

Purpose: This register specifies the address bits [31:0] of the cache line that triggers linefill error interrupt.

Usage constrains: Check this register when Linefill Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:0	lf_err_addr	RO	0x0	FLC linefill error interrupt address [31:0]

6.21.27 FLC_LF_Err_Int_Addr_H

Purpose: This register specifies the address bits [63:32] of the cache line that triggers linefill error interrupt.

Usage constrains: Check this register when Linefill Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:0	lf_err_addr_h	RO	0x0	FLC linefill error interrupt address [63:32]

6.21.28 FLC_LF_Err_Int_Info

Purpose: This register provides detailed info about the linefill error interrupt.

Usage constrains: Check this register when Linefill Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:6	RSVD	RO	0x0	Reserved
5:4	lf_axi_resp	RO	0x0	FLC linefill error axi response.
3:2	lf_chi_resperr	RO	0x0	FLC linefill error chi response.
1	lf_err_source	RO	0x0	FLC linefill error interrupt source: Internal (0) or External (1)
0	lf_err_rw	RO	0x0	FLC linefill error interrupt Read (0) or Write (1)

6.21.29 FLC_WB_Err_Int_Addr

Purpose: This register specifies the address bits [31:0] of the cache line that triggers write back error interrupt.

Usage constrains: Check this register when write-back Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:0	wb_err_addr	RO	0x0	FLC write back error interrupt address [31:0]

6.21.30 FLC_WB_Err_Int_Addr_H

Purpose: This register specifies the address bits [63:32] of the cache line that triggers write back error interrupt.

Usage constrains: Check this register when write-back Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:0	wb_err_addr_h	RO	0x0	FLC write back error interrupt address [63:32]

6.21.31 FLC_WB_Err_Int_Info

Purpose: This register provides detailed info about the write back error interrupt.

Usage constrains: Check this register when write-back Error interrupt received.

Bits	Field Name	Type	Reset	Description
31:6	RSVD	RO	0x0	Reserved
5:4	wb_axi_resp	RO	0x0	FLC write back error axi response.
3:2	wb_chi_resperr	RO	0x0	FLC write back error chi response.
1	wb_err_source	RO	0x0	FLC write back error interrupt source: Internal (0) or External (1)
0	wb_err_rw	RO	0x0	FLC write back error interrupt Read (0) or Write (1)

6.21.32 FLC_LF_Done_Int_Addr

Purpose: This register specifies address bits [31:0] of the cache line that triggers linefill done interrupt.

Usage constrains: Check this register when Linefill Done interrupt received.

Bits	Field Name	Type	Reset	Description
31:0	lf_int_addr	RO	0x0	FLC Linefill Done Interrupt Address [31:0]

6.21.33 FLC_LF_Done_Int_Addr_H

Purpose: This register specifies address bits [63:32] of the cache line that triggers linefill done interrupt.

Usage constrains Check this register when Linefill Done interrupt received.

Bits	Field Name	Typ e	Rese t	Description
31:0	lf_int_addr_h	RO	0x0	FLC Linefill Done Interrupt Address [63:32]

6.21.34 FLC_Evict_Int_Addr

Purpose: This register specifies address bits [31:0] of the cache line that triggers eviction done interrupt.

Usage constrains: Check this register when Eviction interrupt received.

Bits	Field Name	Typ e	Rese t	Description
31:0	evict_addr	RO	0x0	FLC Eviction Occurrence Interrupt Address [31:0]

6.21.35 FLC_Evict_Int_Addr_H

Purpose: This register specifies address bits [63:32] of the cache line that triggers linefill done interrupt.

Usage constrains: Check this register when Eviction interrupt received.

Bits	Field Name	Typ e	Rese t	Description
31:0	evict_addr_h	RO	0x0	FLC Eviction Occurrence Interrupt Address [63:32]

6.21.36 FLC_Misc_Status

Purpose: This register provides miscellaneous information about FLC1 status.

Usage constrains: None

Bits	Field Name	Type	Rese t	Description
31:6	RSVD	RO	0x0	Reserved
5	writeback_engine_idle	RO	0x0	write-back Engine is in idle state.
4	linefill_fifo_empty	RO	0x0	Linefill FIFO is in empty state.
3:1	RSVD	RO	0x0	Reserved
0	FLC_idle	RO	0x0	FLC is in idle state.

6.21.37 LC_TID_Select

Purpose: This register is used for TID field when enabling use_tid option in the cache maintenance instrµCtions in FLC_Mnt_Req.

Usage constrains: Must be set before issue a cache maintenance request to FLC_Mnt_Req.

Bits	Field Name	Typ e	Rese t	Description
31:5	RSVD	RO	0x0	Reserved
4:0	Tid_sel	RW	0x0	FLC TID Selection

6.21.38 FLC_CHI_Node

Purpose: This register is used to define the AMBA CHI node ID for FLC.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:27	RSVD	RO	0x0	Reserved
26:16	downstream_nid	RW	0x3	Downstream node ID
15:11	RSVD	RO	0x0	Reserved
10:0	FLC_nid	RW	0x2	FLC node ID

6.21.39 FLC_Interleave_Control

Purpose: This register is used to program the address interleaving granularity.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1:0	interleave_granularity	RW	0x0	interleave granularity 0x0 interleave disabled 0x1 4KB interleave 0x2 8KB interleave 0x3 16KB interleave Note: if non-zero granularity is programmed, FLC decoders must be programmed to cover the address space of the whole FLC cluster. Otherwise FLC decoders should be programmed to cover the address space of this FLC slice only.

6.21.40 FLC_Addr_Map_0

Purpose: This register is used to define system address map for FLC1 cacheable region 0.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_0	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_0	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......

Bits	Field Name	Typ e	Rese t	Description
0	addr_map_vld_0	RW	0x1	Address map valid

6.21.41 FLC_Addr_Map_0_H

Purpose: This register is used to define system address map for FLC1 cacheable region 0.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_0_h	RW	0x0	Start address [63:32]

6.21.42 FLC_Addr_Map_1

Purpose: This register is used to define system address map for FLC1 cacheable region 1.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Typ e	Rese t	Description
31:12	start_addr_1	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_1	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_1	RW	0x0	Address map valid

6.21.43 FLC_Addr_Map_1_H

Purpose: This register is used to define system address map for FLC1 cacheable region 1.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_1_h	RW	0xff	Start address [63:32]

6.21.44 FLC_Addr_Map_2

Purpose: This register is used to define system address map for FLC1 cacheable region 2.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_2	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_2	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_2	RW	0x0	Address map valid

6.21.45 FLC_Addr_Map_2_H

Purpose: This register is used to define system address map for FLC1 cacheable region 2.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_2_h	RW	0xff	Start address [63:32]

6.21.46 FLC_Addr_Map_3

Purpose: This register is used to define system address map for FLC1 cacheable region 3.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_3	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
5:1	area_length_3	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_3	RW	0x0	Address map valid

6.21.47 FLC_Addr_Map_3_H

Purpose: This register is used to define system address map for FLC1 cacheable region 3.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_3_h	RW	0xff	Start address [63:32]

6.21.48 FLC_Addr_Map_4

Purpose: This register is used to define system address map for FLC1 cacheable region 4.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_4	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_4	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_4	RW	0x0	Address map valid

6.21.49 FLC_Addr_Map_4_H

Purpose: This register is used to define system address map for FLC1 cacheable region 4.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_4_h	RW	0xff	Start address [63:32]

6.21.50 FLC_Addr_Map_5

Purpose: This register is used to define system address map for FLC1 cacheable region 5.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory

Bits	Field Name	Type	Reset	Description
31:12	start_addr_5	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_5	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_5	RW	0x0	Address map valid

6.21.51 FLC_Addr_Map_5_H

Purpose: This register is used to define system address map for FLC1 cacheable region 5.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_5_h	RW	0xff	Start address [63:32]

6.21.52 FLC_Addr_Map_6

Purpose: This register is used to define system address map for FLC1 cacheable region 6.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable

region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_6	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_6	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_6	RW	0x0	Address map valid

6.21.53 FLC_Addr_Map_6_H

Purpose: This register is used to define system address map for FLC1 cacheable region 6.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_6_h	RW	0xff	Start address [63:32]

6.21.54 FLC_Addr_Map_7

Purpose: This register is used to define system address map for FLC1 cacheable region 7.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_7	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
5:1	area_length_7	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_7	RW	0x0	Address map valid

6.21.55 FLC_Addr_Map_7_H

Purpose: This register is used to define system address map for FLC1 cacheable region 7.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_7_h	RW	0xff	Start address [63:32]

6.21.56 FLC_Addr_Map_8

Purpose: This register is used to define system address map for FLC1 cacheable region 8.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_8	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_8	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_8	RW	0x0	Address map valid

6.21.57 FLC_Addr_Map_8_H

Purpose: This register is used to define system address map for FLC1 cacheable region 8.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_8_h	RW	0xff	Start address [63:32]

6.21.58 FLC_Addr_Map_9

Purpose: This register is used to define system address map for FLC1 cacheable region 9.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_9	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_9	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_9	RW	0x0	Address map valid

6.21.59 FLC_Addr_Map_9_H

Purpose: This register is used to define system address map for FLC1 cacheable region 9.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_9_h	RW	0xff	Start address [63:32]

6.21.60 FLC_Addr_Map_10

Purpose: This register is used to define system address map for FLC1 cacheable region 10.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable

region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_10	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_10	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_10	RW	0x0	Address map valid

6.21.61 FLC_Addr_Map_10_H

Purpose: This register is used to define system address map for FLC1 cacheable region 10.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_10_h	RW	0xff	Start address [63:32]

6.21.62 FLC_Addr_Map_11

Purpose: This register is used to define system address map for FLC1 cacheable region 11.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_11	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
5:1	area_length_11	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_11	RW	0x0	Address map valid

6.21.63 FLC_Addr_Map_11_H

Purpose: This register is used to define system address map for FLC1 cacheable region 11.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_11_h	RW	0xff	Start address [63:32]

6.21.64 FLC_Addr_Map_12

Purpose: This register is used to define system address map for FLC1 cacheable region 12.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_12	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_12	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_12	RW	0x0	Address map valid

6.21.65 FLC_Addr_Map_12_H

Purpose: This register is used to define system address map for FLC1 cacheable region 12.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_12_h	RW	0xff	Start address [63:32]

6.21.66 FLC_Addr_Map_13

Purpose: This register is used to define system address map for FLC1 cacheable region 13.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_13	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_13	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_13	RW	0x0	Address map valid

6.21.67 FLC_Addr_Map_13_H

Purpose: This register is used to define system address map for FLC1 cacheable region 13.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_13_h	RW	0xff	Start address [63:32]

6.21.68 FLC_Addr_Map_14

Purpose: This register is used to define system address map for FLC1 cacheable region 14.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable

region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_14	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_14	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_14	RW	0x0	Address map valid

6.21.69 FLC_Addr_Map_14_H

Purpose: This register is used to define system address map for FLC1 cacheable region 14.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_14_h	RW	0xff	Start address [63:32]

6.21.70 FLC_Addr_Map_15

Purpose: This register is used to define system address map for FLC1 cacheable region 15.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_15	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
5:1	area_length_15	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_15	RW	0x0	Address map valid

6.21.71 FLC_Addr_Map_15_H

Purpose: This register is used to define system address map for FLC1 cacheable region 15

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC1 cacheable region will be bypassed to downstream memory.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_15_h	RW	0xff	Start address [63:32]

6.21.72 FLC_Shadow_Addr_Map

Purpose: This register is used to define FLC2 Shadow address region.

Usage constrains: None

Bits	Field Name	Typ e	Res et	Description
31:12	shadow_start_addr	RW	0x0	Shadow Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	shadow_area_length	RW	0x12	Shadow address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB Note: must be the same as External DDR size
0	shadow_map_vld	RW	0x1	Shadow address map valid
31:0	start_addr_15_h	RW	0xff	Start address [63:32]

6.21.73 FLC_Shadow_Addr_Map_h

Purpose: This register is used to define FLC2 Shadow address region.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	shadow_start_addr_h	RW	0x0	Shadow Start Address[63:32]

6.21.74 FLC1_NC_Addr_Map

Purpose: This register is used to define FLC1 Non-cacheable region.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:12	nc1_start_addr	RW	0x0	Non-Cacheable Region start address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc1_area_length	RW	0xF	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	nc1_addr_map_vld	RW	0x1	Non-Cacheable Memory Region valid

6.21.75 FLC1_NC_Addr_Map_h

Purpose: This register is used to define FLC1 Non-cacheable region.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	nc1_start_addr_h	RW	0x0	Non-Cacheable Region start address[63:32]

6.21.76 FLC2_NC_Addr_Map

Purpose: This register is used to define FLC2 Non-cacheable region.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:12	nc2_start_addr	RW	0x0	Non-Cacheable Region start address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
5:1	nc2_area_length	RW	0xF	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	nc2_addr_map_vld	RW	0x1	Non-Cacheable Memory Region valid

6.21.77 LC2_NC_Addr_Map_h

Purpose: This register is used to define FLC2 Non-cacheable region.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	nc2_start_addr_h	RW	0x0	Non-Cacheable Region start address[63:32]

6.21.78 FLC2_NC_Addr_Map_4

Purpose: This register is used to define FLC2 Non-cacheable address map register 4.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:12	nc2_start_addr_4	RW	0x0	Non-Cacheable Region start address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc2_area_length_4	RW	0xF	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	nc2_addr_map_vld_4	RW	0x1	Non-Cacheable Memory Region valid

6.21.79 FLC2_NC_Addr_Map_4_h

Purpose: This register is used to define FLC2 Non-cacheable address map register 4.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	nc2_start_addr_4_h	RW	0x0	Non-Cacheable Region start address[63:32]

6.21.80 FLC_Trfc_Rglt_QoS_Control_fg2ipm_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_0	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_0	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_0	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_0	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_0	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_0	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.81 FLC_Trfc_Rglt_QoS_Control_fg2ipm_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_1	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_1	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_1	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_1	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_1	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_1	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.82 FLC_Trfc_Rglt_QoS_Control_bg2ipm_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Background to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_2	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_2	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_2	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_2	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_2	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_2	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.83 FLC_Trfc_Rglt_QoS_Control_bg2ipm_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_3	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_3	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_3	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_3	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_3	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_3	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.84 FLC_Trfc_Rglt_QoS_Control_fg2ext_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_4	RW	0x0	QoS override value

Bits	Field Name	Type	Reset	Description
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_4	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_4	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_4	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_4	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_4	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.85 FLC_Trfc_Rglt_QoS_Control_fg2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_5	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_5	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_5	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_5	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_5	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_5	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.86 FLC_Trfc_Rglt_QoS_Control_bg2ext_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_6	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_6	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
6	qos_control_enable_6	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_6	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_6	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_6	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.87 FLC_Trfc_Rglt_QoS_Control_bg2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Background to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_7	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_7	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_7	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_7	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_7	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_7	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.88 FLC_Trfc_Rglt_QoS_Control_rtb2ipm_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_8	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_8	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_8	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_8	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
2	rate_en_8	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_8	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.21.89 FLC_QoS_Lat_fg2ipm_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_0	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_0	RW	0x0	Transaction Target Latency / Period

6.21.90 FLC_QoS_Lat_fg2ipm_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_1	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_1	RW	0x0	Transaction Target Latency / Period

6.21.91 FLC_QoS_Lat_bg2ipm_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Backgound to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_2	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_2	RW	0x0	Transaction Target Latency / Period

6.21.92 FLC_QoS_Lat_bg2ipm_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_3	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_3	RW	0x0	Transaction Target Latency / Period

6.21.93 FLC_QoS_Lat_fg2ext_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_4	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_4	RW	0x0	Transaction Target Latency / Period

6.21.94 FLC_QoS_Lat_fg2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_5	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_5	RW	0x0	Transaction Target Latency / Period

6.21.95 FLC_QoS_Lat_bg2ext_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Backgound to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_6	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10

Bits	Field Name	Typ e	Rese t	Description
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_6	RW	0x0	Transaction Target Latency / Period

6.21.96 FLC_QoS_Lat_bg2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Backgound to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_7	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_7	RW	0x0	Transaction Target Latency / Period

6.21.97 FLC_QoS_Lat_rtb2ipm_rd

Purpose: This register is used to define FLC QoS Latency Target Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_8	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_8	RW	0x0	Transaction Target Latency / Period

6.21.98 FLC_QoS_Lat_Range_fg2ipm_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to IPM Read Traffic

Usage constrains

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_0	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_0	RW	0x0	Minimum QoS Value

6.21.99 FLC_QoS_Lat_Range_fg2ipm_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_1	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_1	RW	0x0	Minimum QoS Value

6.21.100 FLC_QoS_Lat_Range_bg2ipm_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Background to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_2	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_2	RW	0x0	Minimum QoS Value

6.21.101 FLC_QoS_Lat_Range_bg2ipm_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_3	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_3	RW	0x0	Minimum QoS Value

6.21.102 FLC_QoS_Lat_Range_fg2ext_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_4	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_4	RW	0x0	Minimum QoS Value

6.21.103 FLC_QoS_Lat_Range_fg2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_5	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_5	RW	0x0	Minimum QoS Value

6.21.104 FLC_QoS_Lat_Range_bg2ext_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_6	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_6	RW	0x0	Minimum QoS Value

6.21.105 FLC_QoS_Lat_Range_bg2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Background to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_7	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_7	RW	0x0	Minimum QoS Value

6.21.106 4.1. FLC_QoS_Lat_Range_rtb2ipm_rd

Purpose: This register is used to define FLC QoS Latency Range Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_8	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_8	RW	0x0	Minimum QoS Value

6.21.107 FLC_Max_Ot_fg2ipm_rd

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_0	RW	0x0	Max outstanding transaction

6.21.108 FLC_Max_Ot_fg2ipm_wr

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_1	RW	0x0	Max outstanding transaction

6.21.109 FLC_Max_Ot_bg2ipm_rd

Purpose: This register is used to define FLC Max Outstanding Register - Background to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_2	RW	0x0	Max outstanding transaction

6.21.110 FLC_Max_Ot_bg2ipm_wr

Purpose: This register is used to define FLC Max Outstanding Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_3	RW	0x0	Max outstanding transaction

6.21.111 FLC_Max_Ot_fg2ext_rd

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_4	RW	0x0	Max outstanding transaction

6.21.112 FLC_Max_Ot_fg2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_5	RW	0x0	Max outstanding transaction

6.21.113 FLC_Max_Ot_bg2ext_rd

Purpose: This register is used to define FLC Max Outstanding Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_6	RW	0x0	Max outstanding transaction

6.21.114 FLC_Max_Ot_bg2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - Background to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_7	RW	0x0	Max outstanding transaction

6.21.115 FLC_Max_Ot_rtb2ipm_rd

Purpose: This register is used to define FLC Max Outstanding Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_8	RW	0x0	Max outstanding transaction

6.21.116 FLC_Peak_Rate_fg2ipm_rd

Purpose: This register is used to define FLC Peak Rate Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_0	RW	0x0	Peak rate

6.21.117 FLC_Peak_Rate_fg2ipm_wr

Purpose: This register is used to define FLC Peak Rate Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_1	RW	0x0	Peak rate

6.21.118 FLC_Peak_Rate_bg2ipm_rd

Purpose: This register is used to define FLC Peak Rate Register - Background to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_2	RW	0x0	Peak rate

6.21.119 FLC_Peak_Rate_bg2ipm_wr

Purpose: This register is used to define FLC Peak Rate Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_3	RW	0x0	Peak rate

6.21.120 FLC_Peak_Rate_fg2ext_rd

Purpose: This register is used to define FLC Peak Rate Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_4	RW	0x0	Peak rate

6.21.121 FLC_Peak_Rate_fg2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_5	RW	0x0	Peak rate

6.21.122 FLC_Peak_Rate_bg2ext_rd

Purpose: This register is used to define FLC Peak Rate Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_6	RW	0x0	Peak rate

6.21.123 FLC_Peak_Rate_bg2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - Background to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_7	RW	0x0	Peak rate

6.21.124 FLC_Peak_Rate_rtb2ipm_rd

Purpose: This register is used to define FLC Peak Rate Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_8	RW	0x0	Peak rate

6.21.125 FLC_Burst_Allow_fg2ipm_rd

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_0	RW	0x0	Burstiness allowance

6.21.126 FLC_Burst_Allow_fg2ipm_wr

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_1	RW	0x0	Burstiness allowance

6.21.127 FLC_Burst_Allow_bg2ipm_rd

Purpose: This register is used to define FLC Burst Allowance Register - Background to IPM Read Traffic

Usage constrains: None-

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_2	RW	0x0	Burstiness allowance

6.21.128 FLC_Burst_Allow_bg2ipm_wr

Purpose: This register is used to define FLC Burst Allowance Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_3	RW	0x0	Burstiness allowance

6.21.129 FLC_Burst_Allow_fg2ext_rd

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_4	RW	0x0	Burstiness allowance

6.21.130 FLC_Burst_Allow_fg2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_5	RW	0x0	Burstiness allowance

6.21.131 FLC_Burst_Allow_bg2ext_rd

Purpose: This register is used to define FLC Burst Allowance Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_6	RW	0x0	Burstiness allowance

6.21.132 FLC_Burst_Allow_bg2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - Background to External Memory Write

Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_7	RW	0x0	Burstiness allowance

6.21.133 FLC_Burst_Allow_rtb2ipm_rd

Purpose: This register is used to define FLC Burst Allowance Register - RTB to IPM Read Traffic

Usage constrains: : None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_8	RW	0x0	Burstiness allowance

6.21.134 FLC_Avg_Rate_fg2ipm_rd

Purpose: This register is used to define FLC Average Rate Register - Foreground to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_0	RW	0x0	Average rate

6.21.135 FLC_Avg_Rate_fg2ipm_wr

Purpose: This register is used to define FLC Average Rate Register - Foreground to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_1	RW	0x0	Average rate

6.21.136 FLC_Avg_Rate_bg2ipm_rd

Purpose: This register is used to define FLC Average Rate Register - Background to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_2	RW	0x0	Average rate

6.21.137 FLC_Avg_Rate_bg2ipm_wr

Purpose: This register is used to define FLC Average Rate Register - Background to IPM Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_3	RW	0x0	Average rate

6.21.138 FLC_Avg_Rate_fg2ext_rd

Purpose: This register is used to define FLC Average Rate Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_4	RW	0x0	Average rate

6.21.139 FLC_Avg_Rate_fg2ext_wr

Purpose: This register is used to define FLC Average Rate Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_5	RW	0x0	Average rate

6.21.140 FLC_Avg_Rate_bg2ext_rd

Purpose: This register is used to define FLC Average Rate Register - Background to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_6	RW	0x0	Average rate

6.21.141 FLC_Avg_Rate_bg2ext_wr

Purpose: This register is used to define FLC Average Rate Register - Background to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_7	RW	0x0	Average rate

6.21.142 FLC_Avg_Rate_rtb2ipm_rd

Purpose: This register is used to define FLC Average Rate Register - RTB to IPM Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_8	RW	0x0	Average rate

6.21.143 FLC_ISR

Purpose: This register is used to present the status of FLC1 interrupts.

Usage constrains: Check this register for specific events that calling interrupt. Write 1 to clear the status bit after corresponding interrupt is handled.

Bits	Field Name	Type	Rese t	Description
31	FLC_lf_done_int	RW	0x0	FLC Linefill done interrupt status 0x1 FLC has finished the linefill and the data is ready to be used
30	FLC_mnt_done_int	RW	0x0	FLC Maintenance done interrupt status 0x1 FLC has finished the maintenance instrµCtion sent from register FLC_Mnt_Req
29	FLC_htag_collide_int	RW	0x0	FLC Hash-tag Collision interrupt status 0x1 FLC has a hash-tag collision triggered by foreground traffic
28	FLC_lf_err_int	RW	0x0	FLC Linefill Error interrupt status 0x1 FLC receives error response during linefill
27	FLC_wb_err_int	RW	0x0	FLC write-back Error interrupt status 0x1 FLC receives error response during write-back
26	FLC_evict_int	RW	0x0	FLC Eviction occur interrupt status 0x1 FLC has encountered a cache line eviction
25	FLC_pm_int	RW	0x0	FLC Performance Monitor interrupt status 0x1 FLC Performance Monitor has triggered an interrupt
24:0	RSVD	RO	0x0	Reserved

6.21.144 FLC_IER

Purpose: This register is used to enable FLC1 interrupts.

Usage constrains: Write 1 to specific bit to enable the interrupt; un-enabled events will not call interrupt upon happen.

Bits	Field Name	Type	Rese t	Description
31	FLC_lf_done_int_en	RW	0x0	FLC Linefill done interrupt enable
30	FLC_mnt_done_int_en	RW	0x1	FLC Maintenance done interrupt enable
29	FLC_htag_collide_int_en	RW	0x0	FLC Hash-tag Collision interrupt enable
28	FLC_lf_err_int_en	RW	0x0	FLC Linefill Error interrupt enable
27	FLC_wb_err_int_en	RW	0x0	FLC write-back Error interrupt enable
26	FLC_evict_int_en	RW	0x0	FLC Eviction occur interrupt enable
25	FLC_pm_int_en	RW	0x0	FLC Performance Monitor interrupt enable
24:0	RSVD	RO	0x0	Reserved

6.21.145 FLC_User_Trigger_IR

Purpose: This register is used to manually trigger interrupt for debug purpose.

Usage constrains: Register must be accessed in secure mode. Write 1 to specific bit to trigger the corresponding interrupt.

Bits	Field Name	Type	Reset	Description
31	FLC_lf_done_usr_int	WO	0x0	FLC Linefill done interrupt
30	FLC_mnt_done_usr_int	WO	0x0	FLC Maintenance done interrupt
29	FLC_htag_collide_usr_int	WO	0x0	FLC Hash-tag Collision interrupt
28	FLC_lf_err_usr_int	WO	0x0	FLC Linefill Error interrupt
27	FLC_wb_err_usr_int	WO	0x0	FLC write-back Error interrupt
26	FLC_evict_usr_int	WO	0x0	FLC Eviction occur interrupt
25	FLC_pm_usr_int	WO	0x0	FLC Performance Monitor interrupt
24:0	RSVD	RO	0x0	Reserved

6.21.146 FLC_TID_0_Control_0

Purpose: This register is used to define region of TID0.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_0	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_0	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.147 FLC_TID_0_Control_1

Purpose: This register is used to define region of TID0.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_0	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_0	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_0	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter

Bits	Field Name	Type	Reset	Description
21:20	allocation_scheme_0	RW	0x3	Allocation scheme for TID_0 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_0	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_0	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.148 FLC_TID_1_Control_0

Purpose: This register is used to define region of TID1.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
21:12	region_start_ca_1	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_1	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.149 FLC_TID_1_Control_1

Purpose: This register is used to define region of TID1.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_1	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_1	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_1	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_1	RW	0x3	Allocation scheme for TID_1 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_1	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_1	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.150 FLC_TID_2_Control_0

Purpose: This register is used to define region of TID2.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_2	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_2	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.151 FLC_TID_2_Control_1

Purpose: This register is used to define region of TID2.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_2	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_2	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_2	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_2	RW	0x3	Allocation scheme for TID_2 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_2	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_2	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.152 FLC_TID_3_Control_0

Purpose: This register is used to define region of TID3.

Usage constraints: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved

21:12	region_start_ca_3	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_3	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.153 FLC_TID_3_Control_1

Purpose: This register is used to define region of TID3.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_3	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_3	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_3	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_3	RW	0x3	Allocation scheme for TID_3 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_3	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_3	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.154 FLC_TID_4_Control_0

Purpose: This register is used to define region of TID4.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_4	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_4	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.155 FLC_TID_4_Control_1

Purpose: This register is used to define region of TID4.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_4	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_4	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_4	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_4	RW	0x3	Allocation scheme for TID_4 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_4	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_4	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.156 FLC_TID_5_Control_0

Purpose: This register is used to define region of TID5.

Usage constrains

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_5	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_5	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.157 FLC_TID_5_Control_1

Purpose: This register is used to define region of TID5.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_5	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_5	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_5	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_5	RW	0x3	Allocation scheme for TID_5 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_5	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines

Bits	Field Name	Type	Reset	Description
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_5	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.158 FLC_TID_6_Control_0

Purpose: This register is used to define region of TID6.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_6	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_6	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.159 FLC_TID_6_Control_1

Purpose: This register is used to define region of TID6.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
30:24	bf_access_cnt_threshold_6	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_6	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_6	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_6	RW	0x3	Allocation scheme for TID_6 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_6	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_6	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.160 FLC_TID_7_Control_0

Purpose: This register is used to define region of TID7.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_7	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_7	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.161 FLC_TID_7_Control_1

Purpose: This register is used to define region of TID7.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_7	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_7	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_7	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_7	RW	0x3	Allocation scheme for TID_7 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_7	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines

Bits	Field Name	Type	Reset	Description
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_7	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.162 FLC_TID_8_Control_0

Purpose: This register is used to define region of TID8.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_8	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_8	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.163 FLC_TID_8_Control_1

Purpose: This register is used to define region of TID8.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
30:24	bf_access_cnt_threshold _8	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_8	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_8	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_8	RW	0x3	Allocation scheme for TID_8 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_8	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_8	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.164 FLC_TID_9_Control_0

Purpose: This register is used to define region of TID9.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_9	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_9	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.165 FLC_TID_9_Control_1

Purpose: This register is used to define region of TID9.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_9	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_9	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_9	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_9	RW	0x3	Allocation scheme for TID_9 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_9	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines

Bits	Field Name	Type	Reset	Description
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_9	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.166 FLC_TID_10_Control_0

Purpose: This register is used to define region of TID10.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_10	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_10	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.167 FLC_TID_10_Control_1

Purpose: This register is used to define region of TID10.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
30:24	bf_access_cnt_threshold_ 10	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_10	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_10	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_10	RW	0x3	Allocation scheme for TID_10 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_10	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_10	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.168 FLC_TID_11_Control_0

Purpose: This register is used to define region of TID11.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_11	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_11	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.169 FLC_TID_11_Control_1

Purpose: This register is used to define region of TID11.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_11	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_11	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_11	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_11	RW	0x3	Allocation scheme for TID_11 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Type	Reset	Description
19:16	clean_threshold_11	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_11	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.170 FLC_TID_12_Control_0

Purpose: This register is used to define region of TID12.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_12	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Typ e	Res et	Description
11:8	region_size_12	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.171 FLC_TID_12_Control_1

Purpose: This register is used to define region of TID12.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_12	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_12	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_12	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_12	RW	0x3	Allocation scheme for TID_12 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_12	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_12	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.172 FLC_TID_13_Control_0

Purpose: This register is used to define region of TID13.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_13	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_13	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.173 FLC_TID_13_Control_1

Purpose: This register is used to define region of TID13.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_13	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_13	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_13	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_13	RW	0x3	Allocation scheme for TID_13 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_13	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_13	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.174 FLC_TID_14_Control_0

Purpose: This register is used to define region of TID14.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_14	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_14	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.175 FLC_TID_14_Control_1

Purpose: This register is used to define region of TID14.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_14	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_14	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_14	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_14	RW	0x3	Allocation scheme for TID_14 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Type	Reset	Description
19:16	clean_threshold_14	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_14	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.176 FLC_TID_15_Control_0

Purpose: This register is used to define region of TID15.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_15	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Type	Reset	Description
11:8	region_size_15	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.177 FLC_TID_15_Control_1

Purpose: This register is used to define region of TID15.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_15	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_15	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_15	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_15	RW	0x3	Allocation scheme for TID_15 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_15	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_15	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.178 FLC_TID_16_Control_0

Purpose: This register is used to define region of TID16.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_16	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_16	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.179 FLC_TID_16_Control_1

Purpose: This register is used to define region of TID16.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
30:24	bf_access_cnt_threshold_16	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_16	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_16	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_16	RW	0x3	Allocation scheme for TID_16 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_16	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_16	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.180 FLC_TID_17_Control_0

Purpose: This register is used to define region of TID17.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_17	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_17	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.181 FLC_TID_17_Control_1

Purpose: This register is used to define region of TID17.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_17	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_17	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_17	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_17	RW	0x3	Allocation scheme for TID_17 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Typ e	Rese t	Description
19:16	clean_threshold_17	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_17	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.182 FLC_TID_18_Control_0

Purpose: This register is used to define region of TID18.

Usage constrains: None

Bits	Field Name	Typ e	Res et	Description
31:22	RSVD	RO	0x18	Reserved
21:12	region_start_ca_1 8	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Typ e	Res et	Description
11:8	region_size_18	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.183 FLC_TID_18_Control_1

Purpose: This register is used to define region of TID18.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_ 18	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_18	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_18	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_18	RW	0x3	Allocation scheme for TID_18 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_18	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_18	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.184 FLC_TID_19_Control_0

Purpose: This register is used to define region of TID19.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_19	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_19	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.185 FLC_TID_19_Control_1

Purpose: This register is used to define region of TID19.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_19	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_19	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_19	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_19	RW	0x3	Allocation scheme for TID_19 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_19	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_19	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.186 FLC_TID_20_Control_0

Purpose: This register is used to define region of TID20.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_20	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_20	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.187 FLC_TID_20_Control_1

Purpose: This register is used to define region of TID20.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_20	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_20	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_20	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_20	RW	0x3	Allocation scheme for TID_20 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Typ e	Rese t	Description
19:16	clean_threshold_20	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_20	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.188 FLC_TID_21_Control_0

Purpose: This register is used to define region of TID21.

Usage constrains: None

Bits	Field Name	Typ e	Res et	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_2 1	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Typ e	Res et	Description
11:8	region_size_21	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.189 FLC_TID_21_Control_1

Purpose: This register is used to define region of TID21.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_21	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_21	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_21	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_21	RW	0x3	Allocation scheme for TID_21 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_21	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_21	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.190 FLC_TID_22_Control_0

Purpose: This register is used to define region of TID22.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_22	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_22	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.191 FLC_TID_22_Control_1

Purpose: This register is used to define region of TID22.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_22	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_22	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_22	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_22	RW	0x3	Allocation scheme for TID_22 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_22	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_22	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.192 FLC_TID_23_Control_0

Purpose: This register is used to define region of TID23.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_23	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_23	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.193 FLC_TID_23_Control_1

Purpose: This register is used to define region of TID23.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_23	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_23	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_23	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_23	RW	0x3	Allocation scheme for TID_23 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Type	Reset	Description
19:16	clean_threshold_23	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_23	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.194 FLC_TID_24_Control_0

Purpose: This register is used to define region of TID24.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_24	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Type	Reset	Description
11:8	region_size_24	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.195 FLC_TID_24_Control_1

Purpose: This register is used to define region of TID24.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_24	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_24	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_24	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_24	RW	0x3	Allocation scheme for TID_24 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_24	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_24	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.196 FLC_TID_25_Control_0

Purpose: This register is used to define region of TID25.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_25	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_25	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.197 FLC_TID_25_Control_1

Purpose: This register is used to define region of TID25.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_25	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_25	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_25	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_25	RW	0x3	Allocation scheme for TID_25 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_25	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_25	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.198 FLC_TID_26_Control_0

Purpose: This register is used to define region of TID26.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_26	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_26	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.199 FLC_TID_26_Control_1

Purpose: This register is used to define region of TID26.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_26	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_26	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_26	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_26	RW	0x3	Allocation scheme for TID_26 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Type	Reset	Description
19:16	clean_threshold_26	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_26	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.200 FLC_TID_27_Control_0

Purpose: This register is used to define region of TID27.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_27	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Type	Reset	Description
11:8	region_size_27	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.201 FLC_TID_27_Control_1

Purpose: This register is used to define region of TID27.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_27	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_27	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_27	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_27	RW	0x3	Allocation scheme for TID_27 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_27	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_27	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.202 FLC_TID_28_Control_0

Purpose: This register is used to define region of TID28.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_28	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_28	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.203 FLC_TID_28_Control_1

Purpose: This register is used to define region of TID28.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
30:24	bf_access_cnt_threshold_28	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_28	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_28	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_28	RW	0x3	Allocation scheme for TID_28 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_28	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_28	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.204 FLC_TID_29_Control_0

Purpose: This register is used to define region of TID29.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_29	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_29	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.205 FLC_TID_29_Control_1

Purpose: This register is used to define region of TID29.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_29	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_29	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_29	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_29	RW	0x3	Allocation scheme for TID_29 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write

Bits	Field Name	Type	Reset	Description
19:16	clean_threshold_29	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_29	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.206 FLC_TID_30_Control_0

Purpose: This register is used to define region of TID30.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_start_ca_30	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size

Bits	Field Name	Type	Reset	Description
11:8	region_size_30	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.207 FLC_TID_30_Control_1

Purpose: This register is used to define region of TID30.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved
30:24	bf_access_cnt_threshold_30	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_30	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_30	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_30	RW	0x3	Allocation scheme for TID_30 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_30	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	purge_depth_30	RW	0xA	Clean line purge depth after purge has been triggered. Write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.208 FLC_TID_31_Control_0

Purpose: This register is used to define region of TID31.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:22	RSVD	RO	0x0	Reserved
21:12	region_sart_ca_31	RW	0x0	Starting ca of the region(32 aligned) 0x0 0 0x1 32 0x2 64 ... Note: Must aligned to region_size
11:8	region_size_31	RW	0xA	Number of cache lines in the region 0x0 32 0x1 64 0x2 128 ... 0x9 16K 0xA 32K Note: when adjusting this field, clean_threshold needs to be adjusted also be less than the number of cacheline of this region
7:0	RSVD	RO	0x0	Reserved

6.21.209 FLC_TID_31_Control_1

Purpose: This register is used to define region of TID31.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
30:24	bf_access_cnt_threshold_31	RW	0x7F	Lines with access count smaller than the Access Count Threshold will be inserted in BF to prevent future allocation
23	rand_replace_en_31	RW	0x0	0x0: use pLRU replacement policy 0x1: use random replacement policy
22	bloom_filter_en_31	RW	0x0	0x0: don't use bloom filter 0x1: use bloom filter
21:20	allocation_scheme_31	RW	0x3	Allocation scheme for TID_31 0x0 No allocation 0x1 Allocate on Read only 0x2 Allocate on Write only 0x3 Allocate on Read and Write
19:16	clean_threshold_31	RW	0x7	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge_x000D_ 0x0 1 clean line_x000D_ 0x1 2 clean lines_x000D_ 0x2 4 clean lines_x000D_ 0x3 8 clean lines_x000D_ 0x4 16 clean lines_x000D_ 0x5 32 clean lines_x000D_ 0x6 64 clean lines_x000D_ 0x7 128 clean lines_x000D_ 0x8 256 clean lines_x000D_ 0x9 512 clean lines_x000D_ 0xA 1K clean lines_x000D_ 0xB 2K clean lines_x000D_ 0xC 4K clean lines_x000D_ 0xD 8K clean lines_x000D_ 0xE 16K clean lines_x000D_ 0xF 32K clean lines
15:4	RSVD	RO	0x0	Reserved
3:0	purge_depth_31	RW	0xA	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.21.210 FLC_PM_Control_0

Purpose: This register provides control settings and status information of FLC performance monitor. Refer to 4.5 Performance Monitor for details.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:25	RSVD	RO	0x0	Reserved
24:22	pm_done_status	RO	0x0	Performance Monitor Job Done Status: 0x0 Performance monitor job not done 0x1 Performance monitor job done without condition fulfilled or counter overflow 0x2 Performance monitor job done with condition fulfilled 0x4 Performance monitor job done with counter overflow All other values reserved
21	cmp_op	RW	0x0	0x0 Raise performance monitor interrupt when Event_A_Count is larger than threshold_value in mode 1, or when Event_A_Count/Event_B_Count is larger than threshold_ratio in mode 2. 0x1 Raise performance monitor interrupt when Event_A_Count is less than threshold_value in mode 1, or when Event_A_Count/Event_B_Count is less than threshold_ratio in mode 2.
20:16	tid_filter_target	RW	0x0	When TID filtering is enabled, only events from the selected TID is recorded by Performance Monitor.
15	tid_filter_enable	RW	0x0	0x0 Disable TID filtering for Performance Monitor 0x1 Enable TID filtering for Performance Monitor
14:10	event_b_sel	RW	0x0	0x0 FLC access count 0x1 FLC hit count 0x2 FLC L1 tag hit 0x3 FLC hash tag hit 0x4 FLC miss count 0x5 FLC read access count 0x6 FLC read hit count 0x7 FLC read miss count 0x8 FLC read L1 tag hit 0x9 FLC read hash tag hit 0xA FLC read hash_0 tag hit 0xB FLC read hash_1 tag hit 0xC FLC read hash_2 tag hit 0xD FLC write access count 0xE FLC write hit count 0xF FLC write miss count 0x10 FLC write L1 tag hit 0x11 FLC write hash tag hit 0x12 FLC write hash_0 tag hit 0x13 FLC write hash_1 tag hit 0x14 FLC write hash_2 tag hit 0x15 FLC eviction count 0x16 FLC Linefill count 0x17 FLC bloom filter insertion 0x18 FLC bloom filter hit

Bits	Field Name	Type	Reset	Description
9:5	event_a_sel	RW	0x0	0x0 FLC access count 0x1 FLC hit count 0x2 FLC L1 tag hit 0x3 FLC hash tag hit 0x4 FLC miss count 0x5 FLC read access count 0x6 FLC read hit count 0x7 FLC read miss count 0x8 FLC read L1 tag hit 0x9 FLC read hash tag hit 0xA FLC read hash_0 tag hit 0xB FLC read hash_1 tag hit 0xC FLC read hash_2 tag hit 0xD FLC write access count 0xE FLC write hit count 0xF FLC write miss count 0x10 FLC write L1 tag hit 0x11 FLC write hash tag hit 0x12 FLC write hash_0 tag hit 0x13 FLC write hash_1 tag hit 0x14 FLC write hash_2 tag hit 0x15 FLC eviction count 0x16 FLC Linefill count 0x17 FLC bloom filter insertion 0x18 FLC bloom filter hit
4	RSVD	RO	0x0	Reserved
3	overflow_int_trigger_dis	RW	0x0	0x1 Disable the interrupt triggering caused by counter overflow. Interrupt is triggered solely due to event condition fulfilled. When overflow, event counter will keep its value. 0x0 Enable the interrupt triggering caused by counter overflow. Refer to pm_int_trigger_reason to find PM interrupt triggered reason
2	pm_int_trigger_reason	RO	0x0	Performance Monitor interrupt triggered reason. 0x0 Event condition fulfilled 0x1 Counter overflow

Bits	Field Name	Type	Reset	Description
1:0	mode_select	RW	0x0	Mode_select field serves as Performance Monitor event counter and count down timer's start/stop switch. When Performance Monitor is enabled, event counters and count down timer keep running; When Performance Monitor is disabled, event counters and count down timer will reset to 0. It is recommended to disable Performance Monitor first before each run. 0x0 Disable Performance Monitor 0x1 Event_A_Count[63:0] is larger/less than the threshold value when count down counter expires. The threshold_value is defined in FLC_PM_Threshold and FLC_PM_Threshold_H. The comparison operation is defined in cmp_op. 0x2 Event_A_Count[63:0]/Event_B_Count[63:0] ratio is larger/less than the threshold_ratio when count down counter expires. The threshold_ratio is defined by event_b_weight / event_a_weight. The comparison operation is defined in cmp_op.

6.21.211 FLC_PM_Countdown

Purpose: This register is used to define the measurement window of Performance Monitor

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	countdown	RW	0x0	Countdown Value [31:0]. Performance Monitor will countdown from the specific value, defines the measurement window.

6.21.212 FLC_PM_Countdown_H

Purpose: This register is used to define the measurement window of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	Countdown_h	RW	0x0	Countdown Value [63:32]. Performance Monitor will countdown from the specific value, defines the measurement window.

6.21.213 FLC_PM_Threshold

Purpose: This register is used to set the threshold_value of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_threshold	RW	0x0	Performance Monitor Threshold Value [31:0] for mode 1 triggering defined by mode_select

6.21.214 FLC_PM_Threshold_H

Purpose: This register is used to set the threshold_value of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_threshold_h	RW	0x0	Performance Monitor Threshold Value [63:32] for mode 1 triggering defined by mode_select

6.21.215 FLC_PM_Counter_A

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_cnt_a	RO	0x0	Used in Performance Monitor mode 1 or mode 2 as Event_A_Count [31:0]

6.21.216 FLC_PM_Counter_A_H

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_cnt_a_h	RO	0x0	Used in Performance Monitor mode 1 or mode 2 as Event_A_Count [63:31]

6.21.217 FLC_PM_Counter_B

Purpose: This register is used to provide event B counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_cnt_b	RO	0x0	Used in Performance Monitor mode 2 as Event_B_Count [31:0]

6.21.218 FLC_PM_Counter_B_H

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_cnt_b_h	RO	0x0	Used in Performance Monitor mode 2 as Event_B_Count [63:32]

6.21.219 FLC_PM_Weight

Purpose This register is to define the threshold ratio of Performance Monitor.

Usage constrains None.

Bits	Field Name	Type	Reset	Description
31:16	event_b_weight	RW	0x0	Event B's weight for mode_2 triggering defined by mode_select. The threshold_ratio is defined by event_b_weight/event_a_weight.
15:0	event_a_weight	RW	0x0	Event A's weight for mode_2 triggering defined by mode_select. The threshold_ratio is defined by event_b_weight/event_a_weight.

6.21.220 FLC_PC_Control

Purpose: This register is used to configure countdown counter settings of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	pc_clk_div	RW	0x0	0x0 divide clock by 1 0x1 divide clock by 2 0x2 divide clock by 4 0x3 divide clock by 8 0x4 divide clock by 16 0x5 divide clock by 32 0x6 divide clock by 64 0x7 divide clock by 128
15:1	RSVD	RO	0x0	Reserved
0	pc_start_cond	RW	0x0	0x1 count down starts on first occurring event 0x0 count down starts immediately

6.21.221 FLC_LRU_Control

Purpose: This register is used to configure the sampling rate of pseudo LRU.

Usage constrains: If no down sampling is applied, pseudo LRU history tree will be updated by every foreground access.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1:0	down_sample_rate	RW	0x0	LRU update down sampling rate 0x0 no down sampling 0x1 1/2 down sampling 0x2 1/4 down sampling 0x3 1/8 down sampling

6.21.222 FLC_Clean_Line_Set

Purpose: This register is used to set the internal clean line counter.

Usage constrains: Register must be accessed in secure mode. Set the clean line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control.

Bits	Field Name	Type	Reset	Description
31:21	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
20:1	clean_line_num	RW	0x0	Indicate the number of remaining clean cache lines in FLC cacheable region
0	set_clean_line	WO	0x0	request to reset the FLC internal clean line counter to the value of clean_line_num. Note: reset the clean line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control

6.21.223 FLC_Unlock_Line_Set

Purpose: This register is used to set the internal unlock line counter.

Usage constrains: Register must be accessed in secure mode. Set the unlock line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control.

Bits	Field Name	Type	Reset	Description
31:21	RSVD	RO	0x0	Reserved
20:1	unlock_line_num	RW	0x0	Indicate the number of remaining unlock cache lines in FLC cacheable region
0	set_unlock_line	WO	0x0	request to reset the FLC internal unlock line counter to the value of unlock_line_num. Note: reset the unlock line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control

6.21.224 FLC_Bf_Refresh_Period

Purpose: This register is used to configure the refresh period of FLC bloom filter.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	refresh_period	RW	0xA00000	bloom filter refresh period

6.21.225 FLC_SRAM_Init

Purpose: This register is to initialize SRAMs of FLC during system initialization or reset.

Usage constrains: Register must be accessed in secure mode. Write 1 to sram_init_req to issue SRAM initialization request before any foreground traffic after system reset.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1	sram_init_req	WO	0x0	Request to initialize FLC SRAM.
0	sram_init_done	RW	0x0	FLC SRAM initialization done status.

6.21.226 4.1.225 FLC_Hash_Control

Purpose: This register is to enable/disable internal hash function upgrade feature.

Usage constrains: Do not toggle at run time.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved
0	hash_improve_en	RW	0x1	0x0 baseline hash function 0x1 upgraded hash function

6.21.227 FLC_TZ_Sel

Purpose: This register is to select the TrustZone region for configure.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:3	RSVD	RO	0x0	Reserved
2:0	tz_region_sel	RW	0x0	select the TrustZone region to config tz_region* register support region 0~7

6.21.228 tz_region_base_low_0

Purpose: TrustZone region_0 base address register.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:12	tz_base_address_low_0	RO	0x0	Control the base address bits[31:12] of Region_0. Base address bits[11:0] are always zero because tz doesn't not permit the region size to be less than 4KB
11:0	RSVD	RO	0x0	Reserved

6.21.229 tz_region_base_high_0

Purpose: TrustZone region_0 base address register.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	tz_base_address_high_0	RO	0x0	Control the base address bits[63:32] of Region_0.

6.21.230 tz_region_top_low_0

Purpose: TrustZone region_0 top address register.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:12	tz_top_address_low_0	RO	0xFFFFFFFF	Control the top address bits[31:12] of Region_0. Base address bits[11:0] are always 0xFFF because tz doesn't not permit the region size to be less than 4KB.
11:0	RSVD	RO	0x0	Reserved

6.21.231 tz_region_top_high_0

Purpose: TrustZone region_0 top address register.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	tz_top_address_high_0	RO	0xFFFFFFFF	Control the top address bits[63:32] of Region_0.

6.21.232 tz_region_attributes_0

Purpose: TrustZone region_0 attributes register, is used to control the permission of Region_0.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31	tz_s_wr_en_0	RW	0x1	Secure global write enable. This bit defines the permissions for the Secure writes in the region. 0: Secure writes to the region is not permitted. 1: Permits Secure write to the region.
30	tz_s_rd_en_0	RW	0x1	Secure global read enable. This bit defines the permissions for the Secure reads in the region. 0: Secure reads to the region is not permitted. 1: Permits Secure read to the region.
29:1	RSVD	RO	0x0	Reserved
0	tz_filter_en_0	RW	0x1	Region enable bit.

6.21.233 tz_region_id_access_0

Purpose: TrustZone region_0 NSAID control register.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:16	tz_nsaid_wr_en_0	RW	0xFFFF	NSAID write enables. This enables NSAID inputs to define Non-secure write access permission. Each bit of the tz_region0_nsaid_wr_en register field is associated with a value on the NSAID signal as follows: Bit 16 Associated with NSAID = 0. Bit 17 Associated with NSAID = 1. Bit 18 Associated with NSAID = 2. ... Bit 31 Associated with NSAID = 15.

Bits	Field Name	Type	Res et	Description
15:0	tz_nsaid_rd_en_0	RW	0xFF FF	NSAID read enables. This enables NSAID inputs to define Non-secure read access permission. Each bit of the tz_region0_nsaid_rd_en register field is associated with a value on the NSAID signal as follows: Bit 0 Associated with NSAID = 0. Bit 1 Associated with NSAID = 1. Bit 2 Associated with NSAID = 2. ... Bit 15 Associated with NSAID = 15.

6.21.234 tz_region_base_low_x

Purpose: TrustZone region_x base address register.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first

Bits	Field Name	Type	Res et	Description
31:12	tz_base_address_low_x	RW	0x0	Control the base address bits[31:12] of Region_x. Base address bits[11:0] are always zero because tz doesn't not permit the region size to be less than 4KB.
11:0	RSVD	RO	0x0	Reserved

6.21.235 tz_region_base_high_x

Purpose: TrustZone region_x base address register.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first

Bits	Field Name	Type	Rese t	Description
31:0	tz_base_address_high_x	RW	0x0	Control the base address bits[63:32] of Region_x.

6.21.236 tz_region_top_low_x

Purpose: TrustZone region_x top address register.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first

Bits	Field Name	Type	Res et	Description
31:12	tz_top_address_low_x	RW	0xFF FF	Control the top address bits[31:12] of Region_x. Base address bits[11:0] are always 0xFFF because tz doesn't not permit the region size to be less than 4KB
11:0	RSVD	RO	0x0	Reserved

6.21.237 tz_region_top_high_x

Purpose: TrustZone region_x top address register.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first.

Bits	Field Name	Type	Reset	Description
31:0	tz_top_address_high_x	RW	0xFF FFFF FF	Control the top address bits[63:32] of Region_x.

6.21.238 tz_region_attributes_x

Purpose: TrustZone region_x attributes register, is used to control the permission of Region_x.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first**.**

Bits	Field Name	Type	Reset	Description
31	tz_s_wr_en_x	RW	0x0	Secure global write enable. This bit defines the permissions for the Secure writes in the region. 0: Secure writes to the region is not permitted. 1: Permits Secure write to the region.
30	tz_s_rd_en_x	RW	0x0	Secure global read enable. This bit defines the permissions for the Secure reads in the region. 0: Secure reads to the region is not permitted. 1: Permits Secure read to the region.
29:1	RSVD	RO	0x0	Reserved
0	tz_filter_en_x	RW	0x0	Region enable bit.

6.21.239 tz_region_id_access_x

Purpose: TrustZone region_x NSAID control register.

Usage constrains: Register must be accessed in secure mode. This is an aliased register, need to select tz_region_sel in FLC_TZ_Sel first**.**

Bits	Field Name	Type	Reset	Description
31:16	tz_nsaid_wr_en_x	RW	0xFF FF	NSAID write enables. This enables NSAID inputs to define Non-secure write access permission. Each bit of the tz_region0_nsaid_wr_en register field is associated with a value on the NSAID signal as follows: Bit 16 Associated with NSAID = 0. Bit 17 Associated with NSAID = 1. Bit 18 Associated with NSAID = 2. ... Bit 31 Associated with NSAID = 15.

Bits	Field Name	Type	Reset	Description
15:0	tz_nsaid_rd_en_x	RW	0xFFFF	NSAID read enables. This enables NSAID inputs to define Non-secure read access permission. Each bit of the tz_region0_nsaid_rd_en register field is associated with a value on the NSAID signal as follows: Bit 0 Associated with NSAID = 0. Bit 1 Associated with NSAID = 1. Bit 2 Associated with NSAID = 2. ... Bit 15 Associated with NSAID = 15.

6.21.240 FLC_Hash_RAM_Sleep_Control

Purpose: This register is used to specify the FLC hash table and PA table SRAM SLP/DSLP mode timing parameters.

Usage constrains: Timing parameters must be programmed in cycle count; program the largest value if multiple SRAM exists.

Bits	Field Name	Type	Reset	Description
31:24	hash_ram_tslp	RW	0x0	Hash RAM waiting time Tslp for entering SLP mode
23:16	hash_ram_tslpwk2clk	RW	0x0	Hash RAM wake up time Tslpwk2clk for the first valid cycle from SLP mode
15:8	hash_ram_tdslp	RW	0x0	Hash RAM waiting time Tdslp for entering DSLP mode
7:0	hash_ram_tdslpwk2clk	RW	0x0	Hash RAM wake up time Tdslpwk2clk for the first valid cycle from DSLP mode

6.21.241 FLC_Dirty_RAM_Sleep_Control

Purpose: This register is used to specify the FLC dirty ram SRAM SLP/DSLP mode timing parameters.

Usage constrains: Timing parameters must be programmed in cycle count.

Bits	Field Name	Type	Reset	Description
31:24	dirty_ram_tslp	RW	0x0	Dirty RAM waiting time Tslp for entering SLP mode
23:16	dirty_ram_tslpwk2clk	RW	0x0	Dirty RAM wake up time Tslpwk2clk for the first valid cycle from SLP mode
15:8	dirty_ram_tdslp	RW	0x0	Dirty RAM waiting time Tdslp for entering DSLP mode
7:0	dirty_ram_tdslpwk2clk	RW	0x0	Dirty RAM wake up time Tdslpwk2clk for the first valid cycle from DSLP mode

6.21.242 FLC_Misc_SRAM_Sleep_Control

Purpose: This register is used to specify Miscellaneous FLC SRAM SLP/DSLP mode timing parameters.

Usage constrains: Timing parameters must be programmed in cycle count; program the largest value if multiple SRAM exists.

Bits	Field Name	Type	Reset	Description
31:24	misc_ram_tslp	RW	0x0	Maximum value of other FLC SRAM waiting time Tslp for entering SLP mode

Bits	Field Name	Type	Reset	Description
23:16	misc_ram_tslpwk2clk	RW	0x0	Maximum value of other FLC SRAM wake up time Tslpwk2clk for the first valid cycle from SLP mode
15:8	misc_ram_tdslp	RW	0x0	Maximum value of other FLC SRAM waiting time Tdslp for entering DSLP mode
7:0	misc_ram_tdslpwk2clk	RW	0x0	Maximum value of other FLC SRAM wake up time Tdslpwk2clk for the first valid cycle from DSLP mode

6.21.243 FLC_L1_Hit_Threshold

Purpose: This register is used to specify the consecutive L1 tag hit count threshold.

Usage constrains None

Bits	Field Name	Type	Reset	Description
31:17	reserved	RO	0x0	Reserved
16	l1_hit_threshold_en	RW	0x0	0x0 threshold disabled, hash ram will only be power gated when FLC is idle 0x1 threshold enabled, hash ram will be power gated when consecutive L1 tag hit number is larger than the threshold
15:0	l1_hit_threshold	RW	0x0	When consecutive L1 tag hit number is larger than this threshold, hash ram will be power gated

6.21.244 FLC_Power_Down_Control

Purpose: This register is used to request FLC to prepare for power down entrance.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:1	reserved	RO	0x0	Reserved
0	power_down_entrance_req	WO	0x0	Request FLC to prepare for powering down entrance. FLC will cut off background traffic, return bus credit and de-assert LINKACTIVEREQ. Once done, an interrupt will be raised and FLC is ready to be powered down.

6.22 FLC2 Registers

6.22.1 FLC_Debug_Control

Purpose: This register provides debug features to the FLC1.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:29	RSVD	RO	0x0	Reserved
28	precise_Interrupt_reporting_enable	RW	0x0	0x1 Enabled FLC will hold until Interrupt is cleared to make sure SW has capture all the interrupt information. 0x0 Disabled FLC will keep running and older data will get overwritten if subsequent interrupt comes before the current interrupt is cleared. Note: this bit is only used for default imprecise interrupts: Linefill Done, Eviction Occur and Hash-tag Collision Occur Notification.
27:24	RSVD	RO	0x0	Reserved
23	SRAM_SLP_Mode_Enable	RW	0x0	0x0 Disable SRAM Sleep Mode 0x1 Enable SRAM Sleep Mode Note: SLP and DSLP mode cannot be enabled at the same time
22	SRAM_DSLP_Mode_Enable	RW	0x0	0x0 Disable SRAM Deep Sleep Mode 0x1 Enable SRAM Deep Sleep Mode Note: SLP and DSLP mode cannot be enabled at the same time
21:17	RSVD	RO	0x0	Reserved
16	L1_tag_hit_disable	RW	0x0	0x0 Normal operation. 0x1 Mask L1 tag hit to zero for debug purpose.
15:3	RSVD	RO	0x0	Reserved
2:1	Forced_Cache_Entry	RW	0x0	Specify the number of cacheline when FLC_Debug is enabled. 0x1 256 0x2 512 0x3 1024
0	FLC_Cache_Debug	RW	0x0	0 = Normal operation. 1 = Use specified number of entries.

6.22.2 FLC_Mnt_Req

Purpose: This register provides interface for Cache maintenance operations. See section 2.9 for details of each operation.

Usage constrains: Program the register fields to issue the corresponding maintenance operations, only one operation can be requested at a time.

Bits	Field Name	Type	Reset	Description
31:27	RSVD	RO	0x0	Reserved
26	init_dram_hash_req	WO	0x0	requests to initialize dram hash tag before sending requests to FLC

Bits	Field Name	Type	Reset	Description
25:16	req_line_count	WO	0x0	Number of cacheline to perform for Allocate/Invalidate/Clean/Lock. 0 = 1 cache line, up to 1024 cache line.
15:14	RSVD	RO	0x0	Reserved
13	use_cache_index	WO	0x0	Use cache index instead of physical address for the supported operations.
12	unlock_enb	WO	0x0	Do unlock instead of lock for lock_request.
11	lock_req	WO	0x0	Request to lock/unlock an entry to the specified address/cache index.
10	clean_req	WO	0x0	Request to clean the dirty cacheline of the specified address.
9	clear_dirty_req	WO	0x0	Request to clear the dirty bit of the specified address, dirty granularity is 4KB.
8	chk_state_req	WO	0x0	Request to check if the specified address is hit/miss, locked/unlocked.
7	clean_all	WO	0x0	Request to clean all the dirty cacheline.
6	invld_all	WO	0x0	Invalid/clean all cache lines, fields use_tid, use_cache_index and req_line_count is ignored when this bit is set.
5	invld_clean	WO	0x0	Clean operation will be done before invalidation.
4	invld_req	WO	0x0	Request to invalidate the specified cache line.
3	allocate_fill_zero	WO	0x0	Fill cacheline content to zero after Allocation.
2	allocate_linefill	WO	0x0	Do LineFill after allocation.
1	allocate_lock	WO	0x0	Lock the cache line after allocation.
0	allocate_req	WO	0x0	Request to allocate a cache line for the specified address.

6.22.3 4.1.3 FLC_Mnt_Addr

Purpose: This register specifies the address bits [31:0] for maintenance operations.

Usage constrains: Use physical address if field use_cache_index in FLC_Mnt_Req is not set,   use cache index if field use_cache_index in FLC_Mnt_Req is asserted.

Bits	Field Name	Type	Reset	Description
31:0	mnt_addr	RW	0x0	FLC Maintenance Address [31:0]

6.22.4 4.1.4 FLC_Mnt_Addr_H

Purpose: This register specifies the address bits [63:32] for maintenance operations.

Usage constrains: Use physical address if field use_cache_index in FLC_Mnt_Req is not set, use cache index if field use_cache_index in FLC_Mnt_Req is asserted.

Bits	Field Name	Type	Reset	Description
31:0	mnt_addr_h	RW	0x0	FLC Maintenance Address [63:32]

6.22.5 4.1.5 FLC_Mnt_Status

Purpose: This register provides the execution result of maintenance operations.

Usage constrains: Check this register after maintenance done interrupt received.

Bits	Field Name	Type	Reset	Description
31:16	RSVD	RO	0x0	Reserved
15	err_lock	RO	0x0	Maintenance operation unsμCcessful due to unlock line threshold reached.
14:8	RSVD	RO	0x0	Reserved
7	lock_done	RO	0x0	Lock operation done.
6	clean_done	RO	0x0	Clean operation done.
5	clear_dirty_done	RO	0x0	Clear dirty bits operation done.
4	chk_state_hit	RO	0x0	Check state hit status 0x0 Miss 0x1 Hit
3	chk_state_lock	RO	0x0	Check state lock status 0x0 Unlocked 0x1 Locked
2	chk_state_done	RO	0x0	Check state operation done.
1	invld_done	RO	0x0	Invalidate operation done.
0	alloc_done	RO	0x0	Allocate operation done.

6.22.6 4.1.6 FLC_Mnt_Lock_Status

Purpose: This register indicates the number of unlocked cache lines in FLC.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:0	unlocked_line_count	RO	0x40000	Unlocked cache line count

6.22.7 4.1.7 FLC_Mnt_Unlock_Threshold

Purpose: This register configures the least unlocked cache lines FLC2 must maintain, if unlocked cache line number is smaller than the threshold, incoming lock request will be aborted.

Usage constrains None.

Bits	Field Name	Type	Reset	Description
31:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
3:0	unlock_threshold	RW	0x3	Unlock line threshold, if unlocked line number is less than threshold, incoming lock request will be failed. 0x0 8 unlock lines 0x1 16 unlock lines 0x2 32 unlock lines 0x3 64 unlock lines 0x4 128 unlock lines 0x5 256 unlock lines 0x6 512 unlock lines 0x7 1K unlock lines 0x8 2K unlock lines 0x9 4K unlock lines 0xA 8K unlock lines 0xB 16K unlock lines 0xC 32K unlock lines 0xD 64K unlock lines 0xE 128K unlock lines 0xF 256K unlock lines

6.22.8 4.1.8 FLC_Mnt_Chk_Status_Addr

Purpose: This register indicates bits [31:0] of the physical address or cache index of the cache line targeted by check state operation.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	chk_st_addr	RO	0x0	Return PA if check status via CA, or return CA if check status via PA

6.22.9 4.1.9 FLC_Mnt_Chk_Status_Addr_H

Purpose: This register indicates bits [63:32] of the physical address or cache index of the cache line targeted by check state operation.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	chk_st_addr_h	RO	0x0	Return PA if check status via CA, or return CA if check status via PA

6.22.10 4.1.10 FLC_LF_Err_Int_Addr

Purpose: This register specifies the address bits [31:0] of the cache line that triggers linefill error interrupt.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	lf_err_addr	RO	0x0	FLC linefill error interrupt address [31:0]

6.22.11 4.1.11 FLC_LF_Err_Int_Addr_H

Purpose: This register specifies the address bits [63:32] of the cache line that triggers linefill error interrupt.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:0	lf_err_addr_h	RO	0x0	FLC linefill error interrupt address [63:32]

6.22.12 4.1.12 FLC_WB_Err_Int_Addr

Purpose: This register specifies the address bits [31:0] of the cache line that triggers write back error interrupt.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:0	wb_err_addr	RO	0x0	FLC write back error interrupt address [31:0]

6.22.13 4.1.13 FLC_WB_Err_Int_Addr_H

Purpose: This register specifies the address bits [63:32] of the cache line that triggers write back error interrupt.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:0	wb_err_addr_h	RO	0x0	FLC write back error interrupt address [63:32]

6.22.14 4.1.14 FLC_NVMe_Status_Field

Purpose: This register provides the NVMe command completion status fields upon linefill/write-back error happens. Refer to the NVMe base specification 1.4 section 4.6.1 for field encoding.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:30	RSVD	RO	0x0	Reserved
29:15	nvme_rd_sf	RO	0x0	NVMe read command status field, check upon linefill error interrupt raised
14:0	nvme_wr_sf	RO	0x0	NVMe write command status field, check upon write-back error interrupt raised

6.22.15 4.1.15 FLC_LF_Done_Int_Addr

Purpose: This register specifies address bits [31:0] of the cache line that triggers linefill done interrupt.

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:0	lf_int_addr	RO	0x0	FLC Linefill Done Interrupt Address [31:0]

6.22.16 4.1.16 FLC_LF_Done_Int_Addr_H

Purpose: This register specifies address bits [63:32] of the cache line that triggers linefill done interrupt.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	lf_int_addr_h	RO	0x0	FLC Linefill Done Interrupt Address [63:32]

6.22.17 4.1.17 FLC_Evict_Int_Addr

Purpose: This register specifies address bits [31:0] of the cache line that triggers eviction done interrupt.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	evict_addr	RO	0x0	FLC Eviction Occurrence Interrupt Address [31:0]

6.22.18 4.1.18 FLC_Evict_Int_Addr_H

Purpose: This register specifies address bits [63:32] of the cache line that triggers linefill done interrupt.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:0	evict_addr_h	RO	0x0	FLC Eviction Occurrence Interrupt Address [63:32]

6.22.19 4.1.19 FLC_Misc_Status

Purpose: This register provides miscellaneous information about FLC2 status.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:6	RSVD	RO	0x0	Reserved
5	writeback_engine_idle	RO	0x0	write-back Engine is in idle state.
4	linefill_fifo_empty	RO	0x0	Linefill FIFO is in empty state.
3:1	RSVD	RO	0x0	Reserved
0	FLC_idle	RO	0x0	FLC is in idle state.

6.22.20 4.1.20 FLC_CHI_Node

Purpose: This register is used to define the AMBA CHI node ID for FLC.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:27	RSVD	RO	0x0	Reserved
26:16	downstream_nid	RW	0x3	Downstream node ID
15:11	RSVD	RO	0x0	Reserved
10:0	FLC_nid	RW	0x2	FLC node ID

6.22.21 4.1.21 FLC_Interleave_Control

Purpose: This register is used to program the address interleaving granularity.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1:0	interleave_granularity	RW	0x0	interleave granularity 0x0 interleave disabled 0x1 4KB interleave 0x2 8KB interleave 0x3 16KB interleave Note: if non-zero granularity is programmed, FLC decoders must be programmed to cover the address space of the whole FLC cluster. Otherwise FLC decoders should be programmed to cover the address space of this FLC slice only.

6.22.22 4.1.22 FLC_Addr_Map_0

Purpose: This register is used to define system address map for FLC2 cacheable region 0.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_0	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_0	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_0	RW	0x1	Address map valid

6.22.23 4.1.23 FLC_Addr_Map_0_H

Purpose: This register is used to define system address map for FLC2 cacheable region 0.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_0_h	RW	0x0	Start address [63:32]

6.22.24 4.1.24 FLC_Addr_Map_1

Purpose: This register is used to define system address map for FLC2 cacheable region 1.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_1	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_1	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_1	RW	0x0	Address map valid

6.22.25 4.1.25 FLC_Addr_Map_1_H

Purpose: This register is used to define system address map for FLC2 cacheable region 1.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_1_h	RW	0xff	Start address [63:32]

6.22.26 4.1.26 FLC_Addr_Map_2

Purpose: This register is used to define system address map for FLC2 cacheable region 2.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_2	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
5:1	area_length_2	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_2	RW	0x0	Address map valid

6.22.27 4.1.27 FLC_Addr_Map_2_H

Purpose: This register is used to define system address map for FLC2 cacheable region 2.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_2_h	RW	0xff	Start address [63:32]

6.22.28 4.1.28 FLC_Addr_Map_3

Purpose: This register is used to define system address map for FLC2 cacheable region 3.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_3	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_3	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_3	RW	0x0	Address map valid

6.22.29 4.1.29 FLC_Addr_Map_3_H

Purpose: This register is used to define system address map for FLC2 cacheable region 3.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_3_h	RW	0xff	Start address [63:32]

6.22.30 4.1.30 FLC_Addr_Map_4

Purpose: This register is used to define system address map for FLC2 cacheable region 4.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_4	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_4	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_4	RW	0x1	Address map valid

6.22.31 4.1.31 FLC_Addr_Map_4_H

Purpose: This register is used to define system address map for FLC2 cacheable region 4.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_4_h	RW	0x0	Start address [63:32]

6.22.32 4.1.32 FLC_Addr_Map_5

Purpose: This register is used to define system address map for FLC2 cacheable region 5.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region

region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_5	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_5	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_1	RW	0x0	Address map valid

6.22.33 4.1.33 FLC_Addr_Map_5_H

Purpose: This register is used to define system address map for FLC2 cacheable region 5.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_5_h	RW	0xff	Start address [63:32]

6.22.34 4.1.34 FLC_Addr_Map_6

Purpose: This register is used to define system address map for FLC2 cacheable region 6.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_6	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
5:1	area_length_6	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_6	RW	0x0	Address map valid

6.22.35 4.1.35 FLC_Addr_Map_6_H

Purpose: This register is used to define system address map for FLC2 cacheable region 6.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_6_h	RW	0xff	Start address [63:32]

6.22.36 4.1.36 FLC_Addr_Map_7

Purpose: This register is used to define system address map for FLC2 cacheable region 7.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:12	start_addr_7	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_7	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_7	RW	0x0	Address map valid

6.22.37 4.1.37 FLC_Addr_Map_7_H

Purpose: This register is used to define system address map for FLC2 cacheable region 7.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_7_h	RW	0xff	Start address [63:32]

6.22.38 4.1.38 FLC_Addr_Map_8

Purpose: This register is used to define system address map for FLC2 cacheable region 8.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:12	start_addr_8	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_8	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_8	RW	0x1	Address map valid

6.22.39 4.1.39 FLC_Addr_Map_8_H

Purpose: This register is used to define system address map for FLC2 cacheable region 8.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_8_h	RW	0x0	Start address [63:32]

6.22.40 4.1.40 FLC_Addr_Map_9

Purpose: This register is used to define system address map for FLC2 cacheable region 9.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable

region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_9	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_9	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_1	RW	0x0	Address map valid

6.22.41 4.1.41 FLC_Addr_Map_9_H

Purpose: This register is used to define system address map for FLC2 cacheable region 9.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_9_h	RW	0xff	Start address [63:32]

6.22.42 4.1.42 FLC_Addr_Map_10

Purpose: This register is used to define system address map for FLC2 cacheable region 10.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_10	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
5:1	area_length_10	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_10	RW	0x0	Address map valid

6.22.43 4.1.43 FLC_Addr_Map_10_H

Purpose: This register is used to define system address map for FLC2 cacheable region 10.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_10_h	RW	0xff	Start address [63:32]

6.22.44 4.1.44 FLC_Addr_Map_11

Purpose: This register is used to define system address map for FLC2 cacheable region 11.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:12	start_addr_11	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_11	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_11	RW	0x0	Address map valid

6.22.45 4.1.45 FLC_Addr_Map_11_H

Purpose: This register is used to define system address map for FLC2 cacheable region 11.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_11_h	RW	0xff	Start address [63:32]

6.22.46 4.1.46 FLC_Addr_Map_12

Purpose: This register is used to define system address map for FLC2 cacheable region 12.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_12	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_12	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_12	RW	0x1	Address map valid

6.22.47 4.1.47 FLC_Addr_Map_12_H

Purpose: This register is used to define system address map for FLC2 cacheable region 12.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_12_h	RW	0x0	Start address [63:32]

6.22.48 4.1.48 FLC_Addr_Map_13

Purpose: This register is used to define system address map for FLC2 cacheable region 13.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable

region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_13	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_13	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_13	RW	0x0	Address map valid

6.22.49 4.1.49 FLC_Addr_Map_13_H

Purpose: This register is used to define system address map for FLC2 cacheable region 13.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_13_h	RW	0xff	Start address [63:32]

6.22.50 4.1.50 FLC_Addr_Map_14

Purpose: This register is used to define system address map for FLC2 cacheable region 14.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:12	start_addr_14	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
5:1	area_length_14	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_14	RW	0x0	Address map valid

6.22.51 4.1.51 FLC_Addr_Map_14_H

Purpose: This register is used to define system address map for FLC2 cacheable region 14.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:0	start_addr_14_h	RW	0xff	Start address [63:32]

6.22.52 4.1.52 FLC_Addr_Map_15

Purpose: This register is used to define system address map for FLC2 cacheable region 15.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Typ e	Rese t	Description
31:12	start_addr_15	RW	0x0	Start address [31:12] Note: 1. Start address must be aligned to area length. 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted.
11:6	RSVD	RO	0x0	Reserved
5:1	area_length_15	RW	0x12	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	addr_map_vld_15	RW	0x0	Address map valid

6.22.53 4.1.53 FLC_Addr_Map_15_H

Purpose: This register is used to define system address map for FLC2 cacheable region 15.

Usage constrains: Register must be accessed in secure mode. All foreground traffic not hitting FLC2 cacheable region will be bypassed to external DDR.

Bits	Field Name	Type	Reset	Description
31:0	start_addr_15_h	RW	0xff	Start address [63:32]

6.22.54 4.1.54 FLC_Shadow_Addr_Map

Purpose: This register is used to define system shadow address map for FLC2 cacheable region.

Usage constrains: Register must be accessed in secure mode. The shadow area length must be the same as size of external DDR, traffic targeting the shadow area will access the external DDR directly.

Bits	Field Name	Type	Reset	Description
31:12	shadow_start_addr	RW	0x0	Shadow Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	shadow_area_length	RW	0x12	Shadow address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB … Note: must be the same as External DDR size
0	shadow_map_vld	RW	0x1	Shadow address map valid

6.22.55 4.1.55 FLC_Shadow_Addr_Map_H

Purpose: This register is used to define system shadow address map for FLC2 cacheable region.

Usage constrains: Register must be accessed in secure mode. The shadow area length must be the same as size of external DDR, traffic targeting the shadow area will access the external DDR directly.

Bits	Field Name	Type	Reset	Description
31:0	shadow_start_addr_h	RW	0x0	Shadow start address [63:32]

6.22.56 4.1.56 FLC_NC_Addr_Map

Purpose: This register is used to define system address map for FLC2 non-cacheable region and where it resides in external DDR.

Usage constrains: Register must be accessed in secure mode. FLC2 non-cacheable region must be smaller than the external DDR, but large enough to cover all data that cannot reside in NAND.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr	RW	0x0	Non-Cacheable Region start address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:9	nc_area_ratio	RW	0x4	Non-Cacheable Memory Region Size's ratio relative to External DDR size All other values not described below are reserved. 0x0 Non-Cacheable Region is the same size as External DDR size 0x1 Non-Cacheable Region is 1/2 size as External DDR size 0x2 Non-Cacheable Region is 1/4 size as External DDR size 0x3 Non-Cacheable Region is 1/8 size as External DDR size 0x4 Non-Cacheable Region is 1/16 size as External DDR size 0x5 Non-Cacheable Region is 1/32 size as External DDR size
8	nc_region_config	RW	0x1	Non-Cacheable Memory Region Configuration 0x0 NC Region maps to higher portion of the External DDR memory 0x1 NC Region maps to lower portion of the External DDR memory
7:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length	RW	0xF	Address map area length, = (2^area_length) * 4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ...... 0x12 1GB ......
0	nc_addr_map_vld	RW	0x1	Non-Cacheable Memory Region valid

6.22.57 4.1.57 FLC_NC_Addr_Map_H

Purpose: This register is used to define system address map for FLC2 non-cacheable region and where it resides in external DDR.

Usage constrains: Register must be accessed in secure mode. FLC2 non-cacheable region must be large enough to cover all data that cannot reside in NAND.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_h	RW	0x0	Non-cacheable region start address [63:32]

6.22.58 4.1.58 FLC_NC_Addr_Map_0

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 0 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled.

This register programming must match FLC1's FLC_Addr_Map_0.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr_0	RW	0x0	Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length_0	RW	0x12	Address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB …
0	nc_map_vld_0	RW	0x0	Address map valid

6.22.59 4.1.59 FLC_NC_Addr_Map_0_H

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 0 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_0_H.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_0_h	RW	0x0	start address [63:32]

6.22.60 4.1.60 FLC_NC_Addr_Map_1

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 1 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_1.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr_1	RW	0x0	Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length_1	RW	0x12	Address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB …
0	nc_map_vld_1	RW	0x0	Address map valid

6.22.61 4.1.61 FLC_NC_Addr_Map_1_H

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 1 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_1_H.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_1_h	RW	0xFF	start address [63:32]

6.22.62 4.1.62 FLC_NC_Addr_Map_2

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 2 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_2.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr_2	RW	0x0	Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length_2	RW	0x12	Address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB …
0	nc_map_vld_2	RW	0x0	Address map valid

6.22.63 4.1.63 FLC_NC_Addr_Map_2_H

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 2 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_2_H.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_2_h	RW	0xFF	start address [63:32]

6.22.64 4.1.64 FLC_NC_Addr_Map_3

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 3 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled.

This register programming must match FLC1's FLC_Addr_Map_3.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr_3	RW	0x0	Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length_3	RW	0x12	Address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB …
0	nc_map_vld_3	RW	0x0	Address map valid

6.22.65 4.1.65 FLC_NC_Addr_Map_3_H

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 3 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_3_H.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_3_h	RW	0xFF	start address [63:32]

6.22.66 4.1.66 FLC_NC_Addr_Map_4

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 4 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_4.

Bits	Field Name	Type	Reset	Description
31:12	nc_start_addr_4	RW	0x0	Start Address[31:12] Note: 1. Start address must be aligned to area length 2. Start address is aligned to 4KB, so the least significant 12 bits are omitted
11:6	RSVD	RO	0x0	Reserved
5:1	nc_area_length_4	RW	0x12	Address map area length, =(2^shadow_area_length)*4KB 0x0 4KB 0x1 8KB 0x2 16KB 0x3 32KB ... 0x12 1GB …
0	nc_map_vld_4	RW	0x0	Address map valid

6.22.67 4.1.67 FLC_NC_Addr_Map_4_H

Purpose: This register is used to define additional system address map for FLC2 non-cacheable region 4 when FLC2 is disabled.

Usage constrains: Register must be accessed in secure mode. This register is only used when FLC2 is disabled. This register programming must match FLC1's FLC_Addr_Map_4_H.

Bits	Field Name	Type	Reset	Description
31:0	nc_start_addr_4_h	RW	0xFF	start address [63:32]

6.22.68 4.1.68 FLC_Disable_Control

Purpose: This register is used to disable FLC2.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved
0	flc2_disable	RW	0x0	0x0 FLC2 enabled 0x1 FLC2 disabled

6.22.69 4.1.69 FLC_Trfc_Rglt_QoS_Control_fg2ext_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_0	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_0	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_0	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_0	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_0	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_0	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.70 4.1.70 FLC_Trfc_Rglt_QoS_Control_bp2ext_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_1	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_1	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_1	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_1	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_1	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_1	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.71 4.1.71 FLC_Trfc_Rglt_QoS_Control_rtb2ext_rd

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - RTB to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_2	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_2	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_2	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_2	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_2	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_2	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.72 4.1.72 FLC_Trfc_Rglt_QoS_Control_fg2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Foregroud to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_3	RW	0x0	QoS override value

Bits	Field Name	Type	Reset	Description
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_3	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_3	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_3	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_3	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_3	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.73 4.1.73 FLC_Trfc_Rglt_QoS_Control_wtb2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_4	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_4	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_4	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_4	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_4	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_4	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.74 4.1.74 FLC_Trfc_Rglt_QoS_Control_bp2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_5	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_5	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
6	qos_control_enable_5	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_5	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_5	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_5	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.75 4.1.75 FLC_Trfc_Rglt_QoS_Control_fz2ext_wr

Purpose: This register is used to define FLC Traffic Regulator QoS Control Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:20	RSVD	RO	0x0	Reserved
19:16	qos_override_6	RW	0x0	QoS override value
15:9	RSVD	RO	0x0	Reserved
8	reg_mode_6	RW	0x0	QoS regulator mode; 0=latency mode, 1=period mode
7	RSVD	RO	0x0	Reserved
6	qos_control_enable_6	RW	0x0	0 bypass input QoS 1 enable QoS control
5	RSVD	RO	0x0	Reserved
4	ot_en_6	RW	0x0	Enable outstanding
3	RSVD	RO	0x0	Reserved
2	rate_en_6	RW	0x0	Enable rate
1	RSVD	RO	0x0	Reserved
0	qos_control_mode_6	RW	0x0	0 enable QoS override mode 1 enable QoS regulation mode

6.22.76 4.1.76 FLC_QoS_Lat_fg2ext_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_0	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_0	RW	0x0	Transaction Target Latency / Period

6.22.77 4.1.77 FLC_QoS_Lat_bp2ext_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Bypass to External Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_1	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_1	RW	0x0	Transaction Target Latency / Period

6.22.78 4.1.78 FLC_QoS_Lat_rtb2ext_rd

Purpose: This register is used to define FLC QoS Latency Target Register - Backgound to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_2	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_2	RW	0x0	Transaction Target Latency / Period

6.22.79 4.1.79 FLC_QoS_Lat_fg2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Foreground to External Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_3	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_3	RW	0x0	Transaction Target Latency / Period

6.22.80 4.1.80 FLC_QoS_Lat_wtb2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_4	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_4	RW	0x0	Transaction Target Latency / Period

6.22.81 4.1.81 FLC_QoS_Lat_bp2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_5	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_5	RW	0x0	Transaction Target Latency / Period

6.22.82 4.1.82 FLC_QoS_Lat_fz2ext_wr

Purpose: This register is used to define FLC QoS Latency Target Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:19	RSVD	RO	0x0	Reserved
18:16	scale_factor_6	RW	0x0	scale factor. The smaller the scaling factor, the more slowly the QoS values change in response to changes in transaction latency. Power of 2 in the range 2^-3 to 2^-10
15:12	RSVD	RW	0x0	Reserved
11:0	lat_tgt_6	RW	0x0	Transaction Target Latency / Period

6.22.83 4.1.83 FLC_QoS_Lat_Range_fg2ext_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_0	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_0	RW	0x0	Minimum QoS Value

6.22.84 4.1.84 FLC_QoS_Lat_Range_bp2ext_rd

Purpose: This register is used to define FLC QoS Latency Range Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_1	RW	0x0	Maximum QoS Value

Bits	Field Name	Typ e	Rese t	Description
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_1	RW	0x0	Minimum QoS Value

6.22.85 4.1.85 FLC_QoS_Lat_Range_rtb2ext_rd

Purpose: This register is used to define FLC QoS Latency Range Register - RTB to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_2	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_2	RW	0x0	Minimum QoS Value

6.22.86 4.1.86 FLC_QoS_Lat_Range_fg2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_3	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_3	RW	0x0	Minimum QoS Value

6.22.87 4.1.87 FLC_QoS_Lat_Range_wtb2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_4	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_4	RW	0x0	Minimum QoS Value

6.22.88 4.1.88 FLC_QoS_Lat_Range_bp2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_5	RW	0x0	Maximum QoS Value

Bits	Field Name	Typ e	Rese t	Description
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_5	RW	0x0	Minimum QoS Value

6.22.89 4.1.89 FLC_QoS_Lat_Range_fz2ext_wr

Purpose: This register is used to define FLC QoS Latency Range Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:12	RSVD	RO	0x0	Reserved
11:8	max_qos_6	RW	0x0	Maximum QoS Value
7:4	RSVD	RO	0x0	Reserved
3:0	min_qos_6	RW	0x0	Minimum QoS Value

6.22.90 4.1.90 FLC_Max_Ot_fg2ext_rd

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_0	RW	0x0	Max outstanding transaction

6.22.91 4.1.91 FLC_Max_Ot_bp2ext_rd

Purpose: This register is used to define FLC Max Outstanding Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_1	RW	0x0	Max outstanding transaction

6.22.92 4.1.92 FLC_Max_Ot_rtb2ext_rd

Purpose: This register is used to define FLC Max Outstanding Register - RTB to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_2	RW	0x0	Max outstanding transaction

6.22.93 4.1.93 FLC_Max_Ot_fg2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - Foreground to External Memory Write

Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_3	RW	0x0	Max outstanding transaction

6.22.94 4.1.94 FLC_Max_Ot_wtb2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_4	RW	0x0	Max outstanding transaction

6.22.95 4.1.95 FLC_Max_Ot_bp2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_5	RW	0x0	Max outstanding transaction

6.22.96 4.1.96 FLC_Max_Ot_fz2ext_wr

Purpose: This register is used to define FLC Max Outstanding Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	max_ot_6	RW	0x0	Max outstanding transaction

6.22.97 4.1.97 FLC_Peak_Rate_fg2ext_rd

Purpose: This register is used to define FLC Peak Rate Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Typ e	Rese t	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_0	RW	0x0	Peak rate

6.22.98 4.1.98 FLC_Peak_Rate_bp2ext_rd

Purpose: This register is used to define FLC Peak Rate Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_1	RW	0x0	Peak rate

6.22.99 4.1.99 FLC_Peak_Rate_rtb2ext_rd

Purpose This register is used to define FLC Peak Rate Register - RTB to External Memory Read Traffic

Usage constrains

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_2	RW	0x0	Peak rate

6.22.100 4.1.100 FLC_Peak_Rate_fg2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_3	RW	0x0	Peak rate

6.22.101 4.1.101 FLC_Peak_Rate_wtb2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_4	RW	0x0	Peak rate

6.22.102 4.1.102 FLC_Peak_Rate_bp2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_5	RW	0x0	Peak rate

6.22.103 4.1.103 FLC_Peak_Rate_fz2ext_wr

Purpose: This register is used to define FLC Peak Rate Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	peak_rate_6	RW	0x0	Peak rate

6.22.104 4.1.104 FLC_Burst_Allow_fg2ext_rd

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_0	RW	0x0	Burstiness allowance

6.22.105 4.1.105 FLC_Burst_Allow_bp2ext_rd

Purpose: This register is used to define FLC Burst Allowance Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_1	RW	0x0	Burstiness allowance

6.22.106 4.1.106 FLC_Burst_Allow_rtb2ext_rd

Purpose: This register is used to define FLC Burst Allowance Register - RTB to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_2	RW	0x0	Burstiness allowance

6.22.107 4.1.107 FLC_Burst_Allow_fg2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_3	RW	0x0	Burstiness allowance

6.22.108 4.1.108 FLC_Burst_Allow_wtb2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_4	RW	0x0	Burstiness allowance

6.22.109 4.1.109 FLC_Burst_Allow_bp2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_5	RW	0x0	Burstiness allowance

6.22.110 4.1.110 FLC_Burst_Allow_fz2ext_wr

Purpose: This register is used to define FLC Burst Allowance Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	burst_allow_6	RW	0x0	Burstiness allowance

6.22.111 4.1.111 FLC_Avg_Rate_fg2ext_rd

Purpose: This register is used to define FLC Average Rate Register - Foreground to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_0	RW	0x0	Average rate

6.22.112 4.1.112 FLC_Avg_Rate_bp2ext_rd

Purpose: This register is used to define FLC Average Rate Register - Bypass to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_1	RW	0x0	Average rate

6.22.113 4.1.113 FLC_Avg_Rate_rtb2ext_rd

Purpose: This register is used to define FLC Average Rate Register - RTB to External Memory Read Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_2	RW	0x0	Average rate

6.22.114 4.1.114 FLC_Avg_Rate_fg2ext_wr

Purpose: This register is used to define FLC Average Rate Register - Foreground to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_3	RW	0x0	Average rate

6.22.115 4.1.115 FLC_Avg_Rate_wtb2ext_wr

Purpose: This register is used to define FLC Average Rate Register - WTB to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_4	RW	0x0	Average rate

6.22.116 4.1.116 FLC_Avg_Rate_bp2ext_wr

Purpose: This register is used to define FLC Average Rate Register - Bypass to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_5	RW	0x0	Average rate

6.22.117 4.1.117 FLC_Avg_Rate_fz2ext_wr

Purpose: This register is used to define FLC Average Rate Register - Fillzero to External Memory Write Traffic

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	avg_rate_6	RW	0x0	Average rate

6.22.118 4.1.119 FLC_ISR

Purpose: This register is used to present the status of FLC2 interrupts. Refer to 4.4 Interrupts/Exceptions for details.

Usage constrains: Check this register for specific events that calling interrupt. Write 1 to clear the status bit after corresponding interrupt is handled.

Bits	Field Name	Type	Reset	Description
31	FLC_lf_done_int	RW	0x0	FLC Linefill done interrupt status 0x1 FLC has finished the linefill and the data is ready to be used
30	FLC_mnt_done_int	RW	0x0	FLC Maintenance done interrupt status 0x1 FLC has finished the maintenance instrμCtion sent from register FLC_Mnt_Req
29	RSVD	RO	0x0	Reserved
28	FLC_lf_err_int	RW	0x0	FLC Linefill Error interrupt status 0x1 FLC receives error response during linefill
27	FLC_wb_err_int	RW	0x0	FLC write-back Error interrupt status 0x1 FLC receives error response during write-back
26	FLC_evict_int	RW	0x0	FLC Eviction occur interrupt status 0x1 FLC has encountered a cache line eviction
25	FLC_pm_int	RW	0x0	FLC Performance Monitor interrupt status 0x1 FLC Performance Monitor has triggered an interrupt
24:0	RSVD	RO	0x0	Reserved

6.22.119 4.1.120 FLC_IER

Purpose: This register is used to enable FLC2 interrupts.

Usage constrains: Write 1 to specific bit to enable the interrupt; un-enabled events will not call interrupt upon happen.

Bits	Field Name	Type	Reset	Description
31	FLC_lf_done_int_en	RW	0x0	FLC Linefill done interrupt enable
30	FLC_mnt_done_int_en	RW	0x1	FLC Maintenance done interrupt enable
29	RSVD	RO	0x0	Reserved
28	FLC_lf_err_int_en	RW	0x0	FLC Linefill Error interrupt enable
27	FLC_wb_err_int_en	RW	0x0	FLC write-back Error interrupt enable
26	FLC_evict_int_en	RW	0x0	FLC Eviction occur interrupt enable
25	FLC_pm_int_en	RW	0x0	FLC Performance Monitor interrupt enable
24:0	RSVD	RO	0x0	Reserved

6.22.120 4.1.121 FLC_User_Trigger_IR

Purpose: This register is used to manually trigger interrupt for debug purpose.

Usage constrains: Register must be accessed in secure mode. Write 1 to specific bit to trigger the corresponding interrupt.

Bits	Field Name	Type	Reset	Description
31	FLC_lf_done_usr_int	WO	0x0	FLC Linefill done interrupt
30	FLC_mnt_done_usr_int	WO	0x0	FLC Maintenance done interrupt
29	RSVD	RO	0x0	Reserved
28	FLC_lf_err_usr_int	WO	0x0	FLC Linefill Error interrupt
27	FLC_wb_err_usr_int	WO	0x0	FLC write-back Error interrupt
26	FLC_evict_usr_int	WO	0x0	FLC Eviction occur interrupt
25	FLC_pm_usr_int	WO	0x0	FLC Performance Monitor interrupt
24:0	RSVD	RO	0x0	Reserved

6.22.121 4.1.122 FLC_TID_Control_1

Purpose: This register provides replacement policy and dirty line purge control in FLC2.

Usage constrains: None

Bits	Field Name	Type	Reset	Description
31:24	RSVD	RO	0x0	Reserved
23	rand_replace_en_0	RW	0x0	0x0 Use pseudo LRU replacement policy 0x1 Use random replacement policy
22:20	RSVD	RO	0x0	Reserved
19:16	clean_threshold_0	RW	0xA	Clean line threshold to trigger background purge. If clean line within the region is less than the threshold, write-back engine will start to purge 0x0 1 clean lines 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines 0x10 64K clean lines 0x11 128K clean lines 0x12 256K clean lines All other values reserved.
15:4	RSVD	RO	0x0	Reserved

Bits	Field Name	Typ e	Rese t	Description
3:0	purge_depth_0	RW	0xD	Clean line purge depth after purge has been triggered. write-back will stop after purge depth has been reached 0x0 1 clean line 0x1 2 clean lines 0x2 4 clean lines 0x3 8 clean lines 0x4 16 clean lines 0x5 32 clean lines 0x6 64 clean lines 0x7 128 clean lines 0x8 256 clean lines 0x9 512 clean lines 0xA 1K clean lines 0xB 2K clean lines 0xC 4K clean lines 0xD 8K clean lines 0xE 16K clean lines 0xF 32K clean lines 0x10 64K clean lines 0x11 128K clean lines 0x12 256K clean lines All other values reserved. Note: clean_threshold + purge_depth must not be larger than region_size + 1Note: clean_threshold + purge_depth must not be larger than region_size + 1

6.22.122 4.1.123 FLC_PM_Control_0

Purpose: This register provides control settings and status information of FLC performance monitor. Refer to 4.5 Performance Monitor for details.

Usage constrains None.

Bits	Field Name	Type	Res et	Description
31:25	RSVD	RO	0x0	Reserved
24:22	pm_done_status	RO	0x0	Performance Monitor Job Done Status: 0x0 Performance monitor job not done 0x1 Performance monitor job done without condition fulfilled or counter overflow 0x2 Performance monitor job done with condition fulfilled 0x4 Performance monitor job done with counter overflow All other values reserved

Bits	Field Name	Type	Reset	Description
21	cmp_op	RW	0x0	0x0 Raise performance monitor interrupt when Event_A_Count is larger than threshold_value in mode 1, or when Event_A_Count/Event_B_Count is larger than threshold_ratio in mode 2. 0x1 Raise performance monitor interrupt when Event_A_Count is less than threshold_value in mode 1, or when Event_A_Count/Event_B_Count is less than threshold_ratio in mode 2.
20:15	RSVD	RO	0x0	Reserved
14:10	event_b_sel	RW	0x0	0x0 FLC access count 0x1 FLC hit count 0x2 FLC L1 tag hit 0x3 FLC hash tag hit 0x4 FLC miss count 0x5 FLC read access count 0x6 FLC read hit count 0x7 FLC read miss count 0x8 FLC read L1 tag hit 0x9 FLC read hash tag hit 0xA FLC write access count 0xB FLC write hit count 0xC FLC write_miss_count 0xD FLC write L1 tag hit 0xE FLC write hash tag hit 0xF FLC eviction count 0x10 FLC Linefill count

Bits	Field Name	Type	Reset	Description
9:5	event_a_sel	RW	0x0	0x0 FLC access count 0x1 FLC hit count 0x2 FLC L1 tag hit 0x3 FLC hash tag hit 0x4 FLC miss count 0x5 FLC read access count 0x6 FLC read hit count 0x7 FLC read miss count 0x8 FLC read L1 tag hit 0x9 FLC read hash tag hit 0xA FLC write access count 0xB FLC write hit count 0xC FLC write_miss_count 0xD FLC write L1 tag hit 0xE FLC write hash tag hit 0xF FLC eviction count 0x10 FLC Linefill count
4	RSVD	RO	0x0	Reserved
3	overflow_int_trigger_dis	RW	0x0	0x1 Disable the interrupt triggering caused by counter overflow. Interrupt is triggered solely due to event condition fulfilled. When overflow, event counter will keep its value. 0x0 Enable the interrupt triggering caused by counter overflow. Refer to pm_int_trigger_reason to find PM interrupt triggered reason
2	pm_int_trigger_reason	RO	0x0	Performance Monitor interrupt triggered reason. 0x0 Event condition fulfilled 0x1 Counter overflow
1:0	mode_select	RW	0x0	Mode_select field serves as Performance Monitor event counter and count down timer's start/stop switch. When Performance Monitor is enabled, event counters and count down timer keep running; When Performance Monitor is disabled, event counters and count down timer will reset to 0. It is recommended to disable Performance Monitor first before each run. 0x0 Disable Performance Monitor 0x1 Event_A_Count[63:0] is larger/less than the threshold value when count down counter expires. The threshold_value is defined in FLC_PM_Threshold and FLC_PM_Threshold_H The comparison operation is defined in cmp_op. 0x2 Event_A_Count[63:0]/Event_B_Count[63:0] ratio is larger/less than the threshold_ratio when count down counter expires. The threshold_ratio is defined by event_b_weight/event_a_weight. The comparison operation is defined in cmp_op.

6.22.123 4.1.124 FLC_PM_Countdown

Purpose: This register is used to define the measurement window of Performance Monitor

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	countdown	RW	0x0	Countdown Value [31:0]. Performance Monitor will countdown from the specific value, defines the measurement window.

6.22.124 4.1.125 FLC_PM_Countdown_H

Purpose: This register is used to define the measurement window of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	Countdown_h	RW	0x0	Countdown Value [63:32]. Performance Monitor will countdown from the specific value, defines the measurement window.

6.22.125 4.1.126 FLC_PM_Threshold

Purpose: This register is used to set the threshold_value of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_threshold	RW	0x0	Performance Monitor Threshold Value [31:0] for mode 1 triggering defined by mode_select

6.22.126 4.1.127 FLC_PM_Threshold_H

Purpose: This register is used to set the threshold_value of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_threshold_h	RW	0x0	Performance Monitor Threshold Value [63:32] for mode 1 triggering defined by mode_select

6.22.127 4.1.128 FLC_PM_Counter_A

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	pm_cnt_a	RO	0x0	Used in Performance Monitor mode 1 or mode 2 as Event_A_Count [31:0]

6.22.128 4.1.129 FLC_PM_Counter_A_H

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Typ e	Rese t	Description
31:0	pm_cnt_a_h	RO	0x0	Used in Performance Monitor mode 1 or mode 2 as Event_A_Count [63:31]

6.22.129 4.1.130 FLC_PM_Counter_B

Purpose: This register is used to provide event B counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Typ e	Rese t	Description
31:0	pm_cnt_b	RO	0x0	Used in Performance Monitor mode 2 as Event_B_Count [31:0]

6.22.130 4.1.131 FLC_PM_Counter_B_H

Purpose: This register is used to provide event A counts during the measurement window.

Usage constrains: None.

Bits	Field Name	Typ e	Rese t	Description
31:0	pm_cnt_b_h	RO	0x0	Used in Performance Monitor mode 2 as Event_B_Count [63:32]

6.22.131 4.1.132 FLC_PM_Weight

Purpose: This register is to define the threshold ratio of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Rese t	Description
31:16	event_b_weight	RW	0x0	Event B's weight for mode_2 triggering defined by mode_select. The threshold_ratio is defined by event_b_weight/event_a_weight.
15:0	event_a_weight	RW	0x0	Event A's weight for mode_2 triggering defined by mode_select. The threshold_ratio is defined by event_b_weight/event_a_weight.

6.22.132 4.1.133 FLC_PC_Control

Purpose: This register is used to configure countdown counter settings of Performance Monitor.

Usage constrains: None.

Bits	Field Name	Type	Res et	Description
31:19	RSVD	RO	0x0	Reserved
18:16	pc_clk_div	RW	0x0	0x0 divide clock by 1 0x1 divide clock by 2 0x2 divide clock by 4 0x3 divide clock by 8 0x4 divide clock by 16 0x5 divide clock by 32 0x6 divide clock by 64 0x7 divide clock by 128

Bits	Field Name	Type	Reset	Description
15:1	RSVD	RO	0x0	Reserved
0	pc_start_cond	RW	0x0	0x1 count down starts on first occurring event 0x0 count down starts immediately

6.22.133 4.1.134 FLC_LRU_Control

Purpose: This register is used to configure the sampling rate of pseudo LRU.

Usage constrains: If no down sampling is applied, pseudo LRU history tree will be updated by every foreground access.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1:0	down_sample_rate	RW	0x0	LRU update down sampling rate 0x0 no down sampling 0x1 1/2 down sampling 0x2 1/4 down sampling 0x3 1/8 down sampling

6.22.134 4.1.135 FLC_Clean_Line_Set

Purpose: This register is used to set the internal clean line counter.

Usage constrains: Register must be accessed in secure mode. Set the clean line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control.

Bits	Field Name	Type	Reset	Description
31:21	RSVD	RO	0x0	Reserved
20:1	clean_line_num	RW	0x0	Indicate the number of remaining clean cache lines in FLC cacheable region
0	set_clean_line	WO	0x0	request to reset the FLC internal clean line counter to the value of clean_line_num. Note: reset the clean line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control

6.22.135 4.1.136 FLC_Unlock_Line_Set

Purpose: This register is used to set the internal unlock line counter.

Usage constrains: Register must be accessed in secure mode. Set the unlock line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control.

Bits	Field Name	Type	Reset	Description
31:21	RSVD	RO	0x0	Reserved
20:1	unlock_line_num	RW	0x0	Indicate the number of remaining unlock cache lines in FLC cacheable region
0	set_unlock_line	WO	0x0	request to reset the FLC internal unlock line counter to the value of unlock_line_num. Note: reset the unlock line counter to correct value after programming FLC_NC_Addr_Map or FLC_Debug_Control

6.22.136 4.1.137 FLC_SRAM_Init

Purpose: This register is to initialize SRAMs of FLC during system initialization or reset.

Usage constrains: Register must be accessed in secure mode. Write 1 to sram_init_req to issue SRAM initialization request before any foreground traffic after system reset.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved
1	sram_init_req	WO	0x0	Request to initialize FLC SRAM.
0	sram_init_done	RW	0x0	FLC SRAM initialization done status.

6.22.137 4.1.138 FLC_NVMe_Init_Done

Purpose: This register is used to indicate NVMe has been sμCcessfully initialized.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved
0	nvme_init_done	RW	0x0	0x0 NVMe has not been initialized, FLC2 won't trigger traffic to NVMe 0x1 NVMe initialization completes, FLC2 can trigger traffic to NVMe

6.22.138 4.1.139 FLC_SQID

Purpose: This register is used to specify SQID of the NVMe submission queue of FLC2

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:16	RSVD	RO	0x0	Reserved
15:0	SQID	RW	0x0	I/O submission queue ID

6.22.139 4.1.140 FLC_SLBA_Bar

Purpose: This register is used to specify the starting logic block address of NVMe.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	slba_bar	RW	0x0	Starting LBA Base Address [31:0]

6.22.140 4.1.141 FLC_SLBA_Bar_H

Purpose: This register is used to specify the starting logic block address of NVMe.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	slba_bar_h	RW	0x0	Starting LBA Base Address [63:32]

6.22.141 4.1.142 FLC_LBA_Size

Purpose: This register is used to specify the size of NVMe logic block unit.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:8	RSVD	RO	0x0	Reserved
7:0	lbads	RW	0x9	This field indicates the LBA data size supported. The value is reported in terms of a power of two (2^n). A value smaller than 9 (i.e. 512 bytes) is not supported. If the value reported is 0h then the LBA format is not supported / used or is not currently available.

6.22.142 4.1.143 FLC_Upper_CID

Purpose: This register is used to specify the unique identifier of each FLC2 when a NVMe submission queue is shared by multiple FLC2.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:3	RSVD	RO	0x0	Reserved
2:0	upper_cid	RW	0x0	Upper NVMe Command ID

6.22.143 4.1.144 FLC_Mgr_dbl_bar

Purpose: This register is used to specify the request doorbell address of the NVMe queue manager.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	dbl_bar	RW	0x0	NVMe Queue Manager Doorbell Base Address [31:0]

6.22.144 4.1.145 FLC_NVMe_CMD_0

Purpose: This register is used by FLC2 to provide NVMe I/O command DWord 0.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:16	command_id	RO	0x0	Command Identifier
15:14	psdt	RO	0x0	PRP or SGL for Data Transfer 0x00 PRPs are used for this transfer else Reserved
13:10	RSVD	RO	0x0	Reserved
9:8	fuse	RO	0x0	Fused Operation 0x00 Normal Operation else Reserved
7:0	opcode	RO	0x0	NVMe command opcode

6.22.145 4.1.146 FLC_NVMe_CMD_1

Purpose: This register is used to provide NVMe namespace ID for FLC2.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	nsid	RW	0x0	Namespace Identifier

6.22.146 4.1.147 FLC_NVMe_CMD_6

Purpose: This register is used by FLC2 to provide NVMe I/O command DWord 6.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	PRP_1	RO	0x0	PRP Entry 1 [31:0]

6.22.147 4.1.148 FLC_NVMe_CMD_7

Purpose: This register is used by FLC2 to provide NVMe I/O command DWord 7.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	PRP_1_h	RO	0x0	PRP Entry 1 [63:32]

6.22.148 4.1.149 FLC_NVMe_CMD_10

Purpose: This register is used by FLC2 to provide NVMe I/O command DWord 10.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	SLBA	RO	0x0	Starting LBA[31:0]

6.22.149 4.1.150 FLC_NVMe_CMD_11

Purpose: This register is used by FLC2 to provide NVMe I/O command DWord 11.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	SLBA_h	RO	0x0	Starting LBA[63:32]

6.22.150 4.1.151 FLC_NVMe_CMD_PENDING

Purpose: This register is used by FLC2 as the NVMe command pending flag to the queue manager.

Usage constrains: As long as this flag is set, FLC2 will not issue a new NVMe I/O command.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
0	cmd_pending	RW	0x0	0x0 No NVMe command is pending fetch by Queue Manager, new command can be issued 0x1 A NVMe command is pending fetch by Queue Manager, don't send a new one Note: NVMe Queue Manager must write 1 to clear this bit after fetching all command dwords

6.22.151 4.1.152 FLC_NVMe_COMP

Purpose: This register is used to provide NVMe I/O command completion information to FLC2.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31	comp_vld	WO	0x0	NVMe command completion valid
30:15	comp_cid	WO	0x0	NVMe command completion cid
14:0	comp_sf	WO	0x0	NVMe command completion status field

6.22.152 4.1.153 FLC_AXI_DMA_bar

Purpose: This register is used to specify the starting logic block address of AXI DMA.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	Axi_dma_bar	RW	0x0	Starting AXI DMA Base Address[31:0]

6.22.153 4.1.154 FLC_AXI_DMA_bar_h

Purpose: This register is used to specify the starting logic block address of AXI DMA..

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	axi_dma_bar_h	RW	0x0	Starting AXI DMA Base Address[63:32]

6.22.154 4.1.155 FLC_AXI_DMA_dbl_bar

Purpose: This register is used to specify the request doorbell address of FLC AXI DMA Doorbell Base Address Register

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	axi_dma_dbl_bar	RW	0x0	AXI DMA Doorbell Base Address[31:0]

6.22.155 4.1.156 FLC_AXI_DMA_CMD_PENDING

Purpose: This register is used by FLC2 as the FLC AXI DMA Command Pending Register

Usage constrains: As long as this flag is set, FLC2 will not issue a new AXI DMA command.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved
0	axi_dma_cmd_pending	RW	0x0	0x0 No AXI DMA command is pending fetch by Queue Manager, new command can be issued 0x1 An AXI DMA command is pending fetch by Queue Manager, don't send a new one Note: AXI DMA master must write 1 to clear this bit after fetching all command dwords

6.22.156 4.1.157 FLC_AXI_DMA_CMD_0

Purpose: This register is used by FLC2 to provide FLC AXI DMA Command DWORD 0 Register.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	tgt_phy_addr	RO	0x0	Target Physical Address[31:0]

6.22.157 4.1.158 FLC_AXI_DMA_CMD_1

Purpose: This register is used to provide NVMe namespace ID for FLC2.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:16	RSVD	RO	0x0	Reserved
15:0	tgt_phy_addr_h	RO	0x0	Target Physical Address[47:32]

6.22.158 4.1.159 FLC_AXI_DMA_CMD_2

Purpose: This register is used by FLC2 to provide FLC AXI DMA Command DWORD 2 Register.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:0	Src_shadow_addr	RO	0x0	Source Shadow Address[31:0]

6.22.159 4.1.160 FLC_AXI_DMA_CMD_3

Purpose: This register is used by FLC2 to FLC AXI DMA Command DWORD 3 Register.

Usage constrains: None.

Bits	Field Name	Type	Reset	Description
31:16	RSVD	RO	0x0	Reserved
15:0	src_shadow_addr_h	RO	0x0	Source Shadow Address[47:32]

6.22.160 4.1.161 FLC_AXI_DMA_CMD_4

Purpose: This register is used by FLC2 to FLC AXI DMA Command DWORD 4 Register.

Usage constrains: None.

Bits	Field Name	Typ e	Rese t	Description
31:16	axi_dma_cid	RO	0x0	AXI DMA COMMAND ID
15:0	RSVD	RO	0x0	Reserved

6.22.161 4.1.162 FLC_AXI_DMA_COMP

Purpose: This register is used to provide NVMe I/O command completion information to FLC2.

Usage constrains: None.

Bits	Field Name	Typ e	Rese t	Description
31	Axi_dma_comp_vld	WO	0x0	AXI DMA command completion valid
30:15	Axi_dma_comp_cid	WO	0x0	AXI DMA command completion cid
14:0	Axi_dma_comp_sf	WO	0x0	AXI DMA command completion status field

6.22.162 4.1.163 FLC_DRAM_Hash_Starting_Addr

Purpose This register is used to specify the starting address of DRAM hash table for FLC2.

Usage constrains Register must be accessed in secure mode.

Bits	Field Name	Typ e	Rese t	Description
31:0	dram_hash_start_addr	RW	0x0	DRAM Hash Table Starting Address[31:0] Note: 1. DRAM Hash table range is 8MB, starting address must align to the size. 2. DRAM Hash table must be within Non-Cacheable Memory Region defined by FLC_NC_Addr_Map. 3. This register is ignored upon FLC2 is disabled.

6.22.163 4.1.164 FLC_DRAM_Hash_Starting_Addr_H

Purpose: This register is used to specify the starting address of DRAM hash table for FLC2.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Typ e	Rese t	Description
31:0	dram_hash_start_addr_h	RW	0x0	DRAM Hash Table Starting Address[63:32] Note: 1. DRAM Hash table range is 4MB, starting address must align to the size. 2. DRAM Hash table must be within Non-Cacheable Memory Region defined by FLC_NC_Addr_Map. 3. This register is ignored upon FLC2 is disabled.

6.22.164 4.1.165 FLC_DRAM_PAtable_Starting_Addr

Purpose: This register is used to specify the starting address of DRAM PA table for FLC2.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	dram_patable_start_addr	RW	0x0	DRAM PA Table Starting Address[31:0] Note: 1. DRAM PAtable range is 1MB, starting address must align to the size. 2. DRAM PAtable must be within Non-Cacheable Memory Region defined by FLC_NC_Addr_Map. 3. This register is ignored upon FLC2 is disabled.

6.22.165 4.1.166 FLC_DRAM_PAtable_Starting_Addr_H

Purpose: This register is used to specify the starting address of DRAM PA table for FLC2.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:0	dram_patable_start_addr_h	RW	0x0	DRAM PA Table Starting Address[63:32] Note: 1. DRAM PAtable range is 1MB, starting address must align to the size. 2. DRAM PAtable must be within Non-Cacheable Memory Region defined by FLC_NC_Addr_Map. 3. This register is ignored upon FLC2 is disabled.

6.22.166 4.1.167 FLC_AES_Control

Purpose: This register is used to enable AES encryption for traffic to NAND.

Usage constrains: Register must be accessed in secure mode. Do not toggle in run time.

Bits	Field Name	Type	Reset	Description
31:1	RSVD	RO	0x0	Reserved
0	AES_en	RW	0x0	0x1 Enable AES encryption for traffic to NAND 0x0 Disable AES encryption for traffic to NAND

6.22.167 4.1.168 FLC_Device_Config

Purpose: This register is used to specify the external DDR size

Usage constrains: Register must be accessed in secure mode. Do not toggle in run time.

Bits	Field Name	Type	Reset	Description
31:2	RSVD	RO	0x0	Reserved

Bits	Field Name	Type	Reset	Description
1:0	device_size	RW	0x2	0x0 External DDR size = 4GB 0x1 External DDR size = 2GB 0x2 External DDR size = 1GB All other values reserved. Note: if external DDR size is larger than 1GB, FLC2 is suggested to be disabled. Because FLC2 cacheable region only supports up to 1GB.

6.22.168 4.1.169 FLC_SRAM_Sleep_Control

Purpose: This register is used to specify FLC2 SRAM SLP/DSLP mode timing parameters.

Usage constrains: Timing parameters must be programmed in cycle count; program the largest value if multiple SRAM exists.

Bits	Field Name	Type	Reset	Description
31:24	sram_tslp	RW	0x0	Maximum value of FLC2 SRAM waiting time Tslp for entering SLP mode
23:16	sram_tslpwk2clk	RW	0x0	Maximum value of FLC2 SRAM wake up time Tslpwk2clk for the first valid cycle from SLP mode
15:8	sram_tdslp	RW	0x0	Maximum value of FLC2 SRAM waiting time Tdslp for entering DSLP mode
7:0	sram_tdslpwk2clk	RW	0x0	Maximum value of FLC2 SRAM wake up time Tdslpwk2clk for the first valid cycle from DSLP mode

6.22.169 4.1.170 FLC_Power_Down_Control

Purpose: This register is used to request FLC to prepare for power down entrance.

Usage constrains: Register must be accessed in secure mode.

Bits	Field Name	Type	Reset	Description
31:1	reserved	RO	0x0	Reserved
0	power_down_entrance_req	WO	0x0	Request FLC to prepare for powering down entrance. FLC will cut off background traffic, return bus credit and de-assert LINKACTIVEREQ. Once done, an interrupt will be raised and FLC is ready to be powered down.

6.23 ETM Registers

6.23.1 ETM_INT_RAW, 0x0000 0000

This register shows the raw pending interrupts yet to be serviced/cleared. Note that these bits are sticky and continue to remain set even after the underlying condition clears.

FW must manually clear these bits using **CSR.ETM_INT_CLR**.

Bits	Field	Defaul t	Type	Description
0	ETM_FIFO0_TR	0	RO	ETM FIFO 0 Threshold Reached.
1	ETM_FIFO1_TR	0	RO	ETM FIFO 1 Threshold Reached.
2	ETM_FIFO0_FULL	0	RO	ETM FIFO 0 Full.
3	ETM_FIFO1_FULL	0	RO	ETM FIFO 1 Full.
4 + X	PTRACE_LOSS_X	0	RO	P-Trace loss on packets of Type Code = X + 1

6.23.2 ETM_INT_ENA (0x0000 0004 – RW, 32-Bit)

Write 1 to the appropriate bit to enable that interrupt source to generate a masked interrupt status.

Bits	Field	Default	Type	Description
0	ETM_FIFO0_TR	0	R/W	ETM FIFO 0 Threshold Reached.
1	ETM_FIFO1_TR	0	R/W	ETM FIFO 1 Threshold Reached.
2	ETM_FIFO0_FULL	0	R/W	ETM FIFO 0 Full.
3	ETM_FIFO1_FULL	0	R/W	ETM FIFO 1 Full.
4 + X	PTRACE_LOSS_X	0	R/W	P-Trace loss on packets of Type Code = X + 1

6.23.3 ETM_INT_MASK, 0x0000 0008

This register is generated using the expression **CSR.ETM_INT_MASK** = **CSR.ETM_INT_RAW** & **CSR.ETM_INT_ENA**. The interrupt towards the CPU is generated by performing a bitwise OR of this register.

Bits	Field	Defaul t	Type	Description
0	ETM_FIFO0_TR	0	RO	ETM FIFO 0 Threshold Reached.
1	ETM_FIFO1_TR	0	RO	ETM FIFO 1 Threshold Reached.
2	ETM_FIFO0_FULL	0	RO	ETM FIFO 0 Full.
3	ETM_FIFO1_FULL	0	RO	ETM FIFO 1 Full.
4 + X	PTRACE_LOSS_X	0	RO	P-Trace loss on packets of Type Code = X + 1

6.23.4 ETM_INT_CLR, 0x0000 000C

Write 1 to clear interrupt pending status on the appropriate bit.

Bits	Field	Default	Type	Description
0	ETM_FIFO0_TR	0	WO	ETM FIFO 0 Threshold Reached.
1	ETM_FIFO1_TR	0	WO	ETM FIFO 1 Threshold Reached.
2	ETM_FIFO0_FULL	0	WO	ETM FIFO 0 Full.
3	ETM_FIFO1_FULL	0	WO	ETM FIFO 1 Full.
4 + X	PTRACE_LOSS_X	0	WO	P-Trace loss on packets of Type Code = X + 1

6.23.5 ETM_INT_FORCE, 0x0000 0010

Write 1 to force an interrupt on the appropriate bit. Intended to be used during FW development.

Bits	Field	Default	Type	Description
0	ETM_FIFO0_TR	0	WO	ETM FIFO 0 Threshold Reached.
1	ETM_FIFO1_TR	0	WO	ETM FIFO 1 Threshold Reached.
2	ETM_FIFO0_FULL	0	WO	ETM FIFO 0 Full.
3	ETM_FIFO1_FULL	0	WO	ETM FIFO 1 Full.
4 + X	PTRACE_LOSS_X	0	WO	P-Trace loss on packets of Type Code = X + 1

6.23.6 ETM_TIM_CURR_RD, 0x0000 0014

Bits	Field	Default	Type	Description
63:0	GTC_LOR	0	RO	Read this register to get the GTC.

6.23.7 ETM_FIFO_LEN, 0x0000 001C

Bits	Field	Default	Type	Description
[Clog2(ETM_FIFO_DEPTH+1)-1:0]	F0_DPT	32	RW	Specify the non-zero length of ETM FIFO 0 that is less than CSR.ETM_FIFO_LEN.

6.23.8 ETM_TIM_SCALER, 0x0000 0020

This sets the GTC frequency scaler. For example, if the timestamp clock is running at 1GHz, setting this to 1 will cause the GTC to tick at a rate of 500MHz. Setting this to 0 will have a tick rate of 1GHz.

Bits	Field	Default	Type	Description
31:0	TIM_SCALER	0	RW	(1 + Value) becomes GTC scaler.

6.23.9 ETM_CFG, 0x0000 0024

Bits	Field	Default	Type	Description
X	TRACE_ENABLE_X	0	RW	Enable input FIFO channel index X.
ETM_TRACE_PORTS	TSTAMP_PRE	1	RW	When 0, the timestamp of a group is updated on every hit. When 1, the timestamp of a group is recorded only when it is created and not updated further.
ETM_TRACE_PORTS + 1	CACHE_REPL_DIS	1	RW	Write 1 to disable the function of evicting the oldest entry when a bµCket has no more space.

6.23.10 ETM_FIFO_WATERMARK, 0x0000 0028

Bits	Field	Default	Type	Description
CLog2(ETM_FIFO_DEPTH):0	F0_FILL	0	RW	Watermark value for ETM FIFO 0. This is the fill level required to fire a threshold reached interrupt
16 + CLog2(ETM_FIFO_DEPTH):16	F1_FILL	0	RW	Watermark value for ETM FIFO 1. This is the fill level required to fire a threshold reached interrupt

6.23.11 ETM_CFG2, 0x0000 002C – RW

Bits	Field	Default	Type	Description
0	TRACE_PRE	1	RW	When 0, the P-Trace field in the hash entry is updated on every hit. When 1, the P-Trace field is not updated on every hit and so will always show the original p-trace that created the group.

6.23.12 ETM_FIFO_PENDING, 0x0000 0030

Bits	Field	Default	Type	Description
CLog2(ETM_FIFO_DEPTH):0	F0_FL	0	RO	Fill level of ETM FIFO 0.
16 + CLog2(ETM_FIFO_DEPTH):16	F1_FL	0	RO	Fill level of ETM FIFO 1.

6.23.13 ETM_ADDR_MTCH_TYPE_CFG, 0x0000 0034

Bits	Field	Default	Type	Description
Clog2(ETM_TRACE_PORTS)-1:0	CHIDX	0	RW	Specify the input channel whose “address match” registers you wish to modify/read.
Clog2(ETM_MAX_FILTER_COUNT) -1 + Clog2(ETM_TRACE_PORTS) : Clog2(ETM_TRACE_PORTS)	FILTERNUM	0	RW	Specify the filter on the PGF whose counters you wish to modify/read. The setting here will influence what is seen on 0x900 to 0x9FF address CSRs.

6.23.14 ETM_SPARE_REG_RW, 0x0000 0038

Bits	Field	Default	Type	Description
31:0	SRW	0	RW	RW spare.

6.23.15 ETM_SPARE_REG_RO, 0x0000 003C

Bits	Field	Default	Type	Description
31:0	SRO	0	RO	RO spare.

6.23.16 ETM_SFC_VAL, 0x0000 0054

Bits	Field	Default	Type	Description
ETM_MAX_EXPIRY_WDT-1:0	QCF	1	RW	Configure the qualification time for packets of Type Code CSR.ETM_TRACE_MAP.SSD_RD_IDX + 1. This should be less than the regular expiry time for p-traces of Type code CSR.ETM_TRACE_MAP.SSD_RD_IDX + 1.

6.23.17 ETM_TRACE_MAP, 0x0000 005C

Bits	Field	Default	Type	Description
2:0	SSD_RD_IDX	2	RW	Specify the input FIFO index on which SSD read traces are expected.
4:3	FLC1_RD_IDX	0	RW	Specify the input FIFO index on which FLC1 read traces are expected.

6.23.18 ETM_TIM_CURR, 0x0000 0074

Bits	Field	Default	Type	Description
63:0	GTC_NEW	N/A	WO	Write new value of GTC here. Always write lower 32-bit first before upper 32-bit.

6.23.19 ETM_FIFO_MUX, 0x0000 0094

Bits	Field	Default	Type	Description
X	EFM_X	1 if X = 2. 1 if X= 3. Else, 0	RW	0: P-Traces originating from input FIFO index X will be routed to ETM FIFO 0. 1: P-Traces originating from input FIFO index X will be routed to ETM FIFO 1

6.23.20 ETM_TSTAMP_CACHE_STATUS, 0x0000 009C

Bits	Field	Default	Type	Description
Y	ETCS_Y	0	RO	0: Timestamp buffer is empty. 1: Timestamp buffer is occupied (i.e., reservation count > 0).

6.23.21 ETM_TRACE_EXPIRED_COUNT[X], 0x0000 0100 + 4*X

Bits	Field	Default	Type	Description
ETM_MAX_EXPIRY_WDT-1:0	ETC_X	128	RW	Specify the expiry count for ETM p-trace from input FIFO channel index X (Type Code = X + 1).

6.23.22 ETM_TRACE_DROPPED[X], 0x0000 0200 + 8*X

Bits	Field	Default	Type	Description
ETM_MAX_EXPIRY_WDT-1:0	ETD_X	0	RO	A 64-bit register that holds a count of the expired p-traces so far, classified by the input FIFO channel index X they came from (Type Code = X + 1). Even though the register is 64-bit, the counter rolls over at $2^{ETM\_TRACE\_DROP\_CTR\_WDT}$. This means that the upper 64 – ETM_TRACE_DROP_CTR_WDT bits are always ZERO. NOTE: This counter will not account for traces dropped due to input filtering (due to PGF).

6.23.23 ETM_RS_PAC_THRESHOLD[X][Z], 0x0000 0300 + Z + 20*X

Bits	Field	Default	Type	Description
ETM_HIT_COUNT_WDT-1:0	EHC_X_Z	0	RW	Specify the PAC threshold for region size that has encoding of Z and Type Code=X + 1. Set to 0 to disable PAC expiry.

6.23.24 ETM_RS_CAP[X], 0x0000 0400 + 4*X

Bits	Field	Default	Type	Description
2:0	ERC_X	HRS	RW	[2:0]: Specify the encoding of the region size you want groups to be capped at (<= HRS), for Type Code = X + 1. For example, capping a Type Code=1, a p-trace with [2:0] == 0 will prevent a region from growing beyond LRS, causing every new p-trace to get its own hash entry.

6.23.25 ETM_PAC_DISABLE[X], 0x0000 0500 + 4*X

The ETM allows fine grained control over enabling PAC expiry for a particular Type Code and Region Size value.

Bits	Field	Default	Type	Description
M	EPD_X_M	0	RW	Write 1 to disable PAC expiry for 4^M * LRS sized groups with Type Code = X + 1 where M is the value in this register. This is like a bitmap.

6.23.26 ETM_RS_FILTER[X], 0x0000 0600 + 4*X

Bits	Field	Default	Type	Description
M	ERF_X_M	0	RW	[M]: Set to 1 to drop packets with region size = 4^M * LRS when writing to ETM FIFO where M is the value in this register. This is like a bitmap.

6.23.27 ETM_FILTER_BYPASS, 0x0000 0700

Bits	Field	Default	Type	Description
31:0	EFB	0xFFFFFFFF	RW	Bypass input p-trace Generation filters. Each bit corresponds to an input p-trace Generation filter. For trace input channel X (Type Code = X + 1), set bit X * ETM_MAX_FILTER_COUNT + T to 1 in this register to bypass the p–trace generation filter number (i.e., stage – can start from 0) T, else set the bit to 0.

6.23.28 ETM_FILTER_MASK, 0x0000 0800 + 8*X

Bits	Field	Default	Type	Description
63:0	EFM	0xFFFFFFFFFFFFFFFF	RW	Provide a filter mask for all input p-trace generation filters on input index X (i.e., Type Code = X + 1). Note that this mask is common for all the p-trace generation filters on a channel path. It operates on pre-upsized traces to save gate count. Each filter mask occupies 64-bit since it is expected that ETM_INPUT_ADDR_WDT [X - 1] will exceed 32-bit. • [0]: When 1, bit 0 of incoming trace is not compared, and is assumed to be equal. • [1]: When 1, bit 1 of incoming trace is not compared, and is assumed to be equal. • … • [63]: When 1, bit 63 of incoming trace is not compared, and is assumed to be equal. Filter mask vector bits at positions greater than the filter's input address width are IGNORED by the design.

6.23.29 ETM_ADDR_MTCH_RANGE, 0x0000 0900 + T*16

Bits	Field	Default	Type	Description
128	RW	0xFFFFFFFFFFFFFFFF	RW	Each match register is 128-bit wide. The set of registers visible is influenced by the setting in register CSR.ETM_ADDR_MTCH_TYPE_CFG. The 128-bit register at 0x900 + (T * 16) corresponds to the match register number T of filter stage number CSR.ETM_ADDR_MTCH_TYPE_CFG.FILTERNUM, on input channel number CSR.ETM_ADDR_MTCH_TYPE_CFG.CHIDX. Every filter has a set of match registers, see ETM_ADDR_MTCH_RANGES.Each 128-bit register is divided into two 64-bit lower and upper ranges as follows: Lower Range: • [31:0]: Match Address lower range, lower 32-bit. • [63:32]: Match Address lower range, upper 32-bit. Upper Range: • [95:64]: Match Address upper range, lower 32-bit. • [128:96]: Match Address upper range, upper 32-bit. Per filter stage, if the incoming trace >= address match lower range and <= address match upper range, the incoming trace is dropped. Match register bits at positions greater than the filter's input address width are IGNORED by the design.

6.24 AXI MASTER Registers

TBD

6.25 PCIe MAC Registers

TBD

6.26 PCIe PHY Registers

TBD

7.0 Acronyms

The chapter provides the definitions for the acronyms and terms used in this document.

Table 51 Acronyms

Acronym or Term	Definition
AES	Advanced Encryption Standard
AHB	Advanced High-Performance Bus
AIA	Advanced Interrupt Architecture
APB	Advanced Peripheral Bus
AXI	Advanced eXtensible Interface
AXI4TG	AXI4 Traffic Generator
CLIC	Core Local Interrupt Controller
CRC	Cyclic Redundancy Check
CSR	Control and Status Register
CXL	Compute Express Link
DBI	Direct Bond Interconnect
DDR	Double Data Rate
DFC	Dynamic Frequency Change
DIMM	Dual In-line Memory Module
DMA	Direct Memory Access
DRAM	Dynamic Random-Access Memory
D2D	Die-To-Die
DSTPH	Destination Path
ECC	Error correction codes
ELA	Embedded Logic Analyzer
ELBI	External Local Bus interface
ETM	Embedded Trace Module
EP	End Point
FLC	Final Level Cache
GFH	Goldfinch (FLC's DDR4 Chiplet)
hPPR	Hard Post-Package Repair
IEM	Industry Economic Modeling
IMSIC	Incoming MSI Controller
IPM	In-Package Memory
IRQ	Interrupt Request
MC	Memory Controller
NOC	Network On Chip
OTP	One-Time Programmable
PLIC	Platform-Level Interrupt Controller

Table 51 Acronyms (Continued)

Acronym or Term	Definition
PMA	Physical Media Attachment
PMU	Power Management Unit
QoS	Quality of Service
QM	Queue Manager
SEC	Security Engine
SECDED	Single-error-correct, double-error-detect
SHA	Secure Hash Algorithm
SLC	System Level Cache
sPPR	Soft Post-Package Repair
SRCPH	Source Path
TCIM	Terminal Count Interrupt Mask
TCM	Transmission Control Module

Contact Information

FLC Technology Group
4655 Old Ironsides Drive
Suite 490, Santa Clara, California 95054

For additional product and ordering information:

www.flctechgroup.com

sales@flctechgroup.com

To provide comments on this document:

techpub@flctechgroup.com

~ End of Document ~