Rockchip PX30 Technical Reference Manual Part1

Revision 1.1
Sep. 2018

Revision History

Date	Revision	Description
2018-9-17	1.1	Cancel the superfluous description
2018-8-13	1.0	Initial Release

Table of Content

Figure Index

Table Index

Warranty Disclaimer

Copyright and Patent Right

Trademarks

Confidentiality

Reverse engineering or disassembly is prohibited.

Copyright © 2018 Rockchip Electronics Co., Ltd.

Chapter 1 System Overview

1.1 Address Mapping

PX30 supports boot from internal bootrom, which supports remap function by software programming. Remap is controlledbyPMU_SGRF_SOC_CON0[13]. When remap is set to 1, the bootrom is un-accessable and PMU_MEM is mapped to address 0Xffff0000.

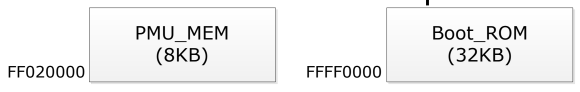
Addr	IP	Addr	IP	Addr	IP
FF1C0000		FF440000			
FF1B0000	I2C3 64K	FF430000	Reserved 64K		
FF1A0000	I2C2 64K	FF400000	GPU 192K		
FF190000	I2C1 64K	FF3C0000	Reserved 256K		
FF180000	I2C0 64K	FF3B0000	NANDC 64K	FFFF0000	
FF178000	UART5 32K	FF3A0000	SFC 64K	FF6C0000	Reserved 9408K
FF170000	UART4 32K	FF390000	EMMC 64K	FF680000	CA35_Debug 256K
FF168000	UART3 32K	FF380000	SDIO 64K	FF640000	Reserved 256K
FF160000	UART2 32K	FF370000	SDMMC 64K	FF638000	DDR_BUF 32K
FF158000	UART1 32K	FF360000	GMAC 64K	FF630000	DDR_GRF 32K
FF150000	Reserved 32K	FF350000	USB2_Host_OHCI 64K	FF620000	DDR_STDBY 64K
FF14C000	GPU_GRF 16K	FF340000	USB2_Host_EHCI 64K	FF610000	DDR_Monitor 64K
FF148000	CORE_GRF 16K	FF300000	USB2_OTG 256K	FF600000	DDR_uPCTL 64K
FF140000	GRF 32K	FF2F0000	CSI_PHY 64K	FF560000	Reserved 640K
FF130000	GIC400 64K	FF2E0000	DSI_PHY 64K	FF558000	Service_vpu 32K
FF120000	Reserved 64K	FF2D0000	OTP_FILTER 64K	FF550000	Service_vo 32K
FF11C000	SGRF 16K	FF2C0000	USB_GRF 64K	FF548000	Service_vi 32K
FF118000	KEY_READER 16K	FF2BC000	PMU_CRU 16K	FF540000	Service_usb 32K
FF110000	OTP_S 32K	FF2B8000	CPU_Boost 16K	FF538000	Service_mmc 32K

Addr	IP	Addr	IP	Addr	IP
	DMA_S		CRU		DDR_Firewall
FF100000	64K	FF2B0000	32K	FF534000	16K
FF0F0000	Reserved 64K	FF2A0000	DDR PHY 64K	FF530000	Service_msch 16K
FF0E0000	Int_MEM 64K	FF290000	OTP_NS 64K	FF52C000	Service sdcard 16K
FF0D0000	Reserved 64K	FF288000	SARADC 32K	FF526000	Service bus2peri 24K
FF0C0000	Reserved 64K	FF280000	TSADC 32K	FF524000	Service bus2msch 8K
FF0B0000	Crypto 64K	FF270000	GPIO3 64K	FF520000	Service gpu 16K
FF0A0000	PDM 64K	FF260000	GPIO2 64K	FF518000	Service_gmac 32K
FF090000	Reserved 64K	FF250000	GPIO1 64K	FF510000	Service crypto 32K
FF080000	I2S2_2CH 64K	FF240000	DMA_NS 64K	FF508000	Service cpu 32K
FF070000	I2S1_2CH 64K	FF230000	DCF 64K	FF500000	Service bus 32K
FF060000	I2S0/TDM_8CH 64K	FF220000	Timer_S 64K	FF4B0000	Reserved 64K
FF050000	PMU_SGRF 64K	FF210000	Timer_NS 64K	FF4A0000	ISP 64K
FF040000	GPIO0 64K	FF208000	PWM1 32K	FF490000	VIP 64K
FF030000	UART0 64K	FF200000	PWM0 32K	FF480000	RGA2-Lite 64K
FF020000	PMU_MEM 64K	FF1F0000	WDT_S 64K	FF470000	VOP_S 64K
FF010000	PMU_GRF 64K	FF1E0000	WDT_NS 64K	FF460000	VOP_M 64K
FF000000	PMU 64K	FF1D8000	SPI1 32K	FF450000	DSI_Host 64K
00000000	DDR 4GB-16MB	FF1D0000	SPI0 32K	FF440000	VPU 64K

Fig. 1-1PX30 Address Mapping

The following figure show the boot address when before remap and after remap

Before Remap



After Remap

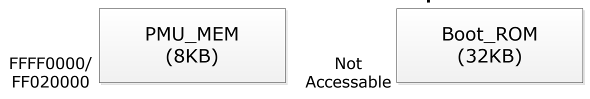


Fig. 1-2PX30 remap function

1.2 System Boot

PX30 provides system boot from off-chip devices such as SDMMC card, eMMC memory, serial nand or nor flash. When boot code is not ready in these devices, also provide system code download into them by USB OTGinterface. All of the boot code will be stored in internal bootrom. The following is the whole boot procedure for boot code, which will be stored in bootromin advance.

The following features are supports.

- Support system boot from the following device:
  - Serial Nor Flash, 1bit data width
  - eMMC Interface, 8bits data width
  - SDMMC Card, 4bits data width
  - Async Nand Flash, 8bit data width
  - 8bits toggle Nand Flash, 8bit data width
- Support system code download by USB OTG

Following figure shows PX30 boot procedure flow.

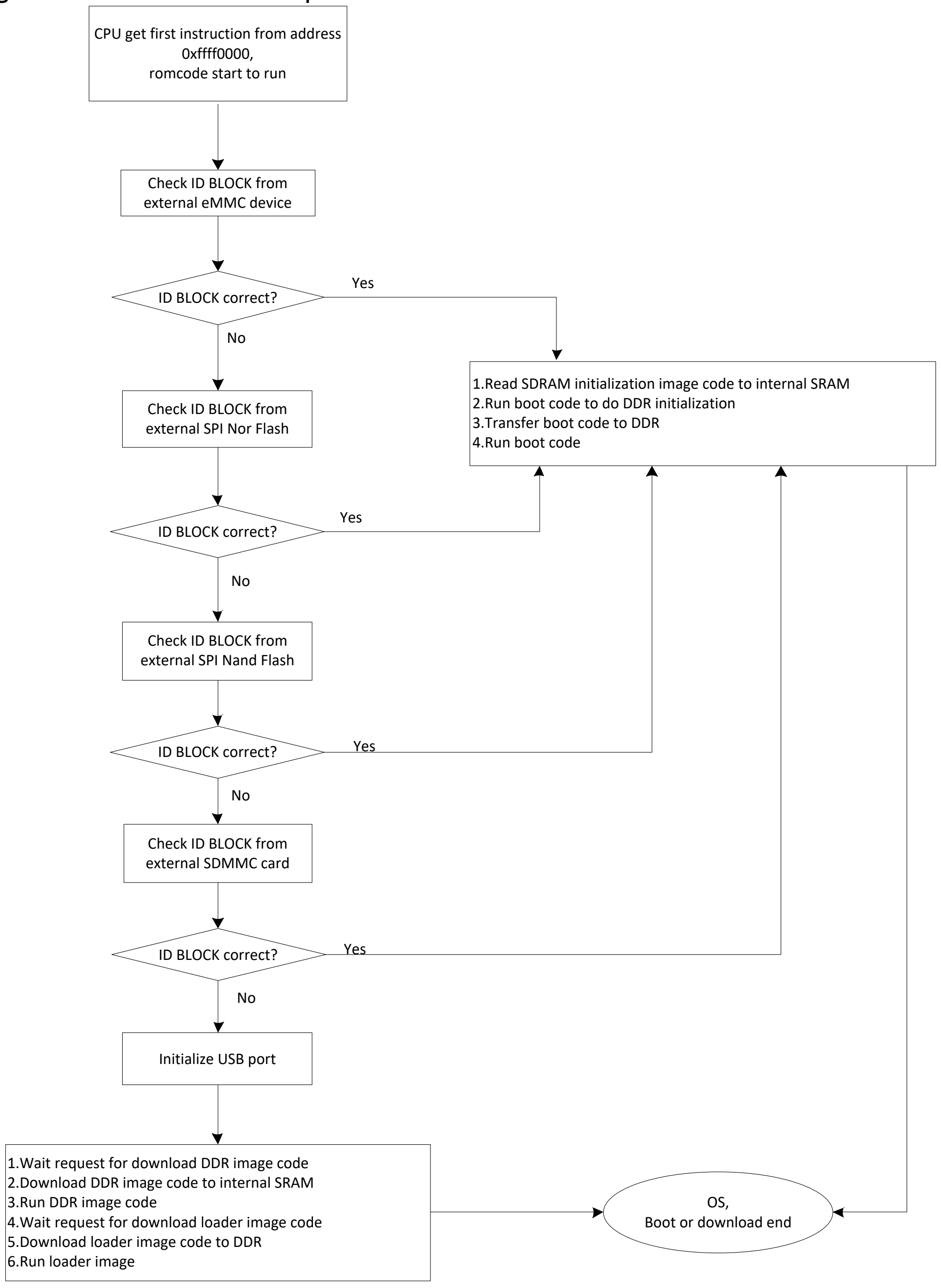

CPU get first instruction from address 0xffff0000, romcode start to run
Check ID BLOCK from external eMMC device
ID BLOCK correct?
Yes
No
1.Read SDRAM initialization image code to internal SRAM
2.Run boot code to do DDR initialization
3.Transfer boot code to DDR
4.Run boot code
Check ID BLOCK from external SPI Nor Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SPI Nand Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SDMMC card
ID BLOCK correct?
Yes
No
Initialize USB port
1.Wait request for download DDR image code
2.Download DDR image code to internal SRAM
3.Run DDR image code
4.Wait request for download loader image code
5.Download loader image code to DDR
6.Run loader image
OS, Boot or download end

Fig. 1-3 PX30 boot procedure flow

1.3 System Interrupt connection

PX30 provides an general interrupt controller(GIC) for CPU, which has 128 SPI (shared peripheral interrupts) interrupt sources and 3 PPI(Private peripheral interrupt) interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to Chapter 9.

Table 1-1 PX30 Interrupt connection list

IRQ Type	IRQ ID	Source(spi)	Polarity
SPI	32	dcf_int_dcf	High level
	33	|dmac_irq	High level
	34	dmac_irq_abort	High level
	35	gpio0_int	High level
	36	gpio1_int	High level
	37	gpio2_int	High level
	38	gpio3_int	High level
	39	i2c_irq_i2c0	High level
	40	i2c_irq_i2c1	High level
	41	i2c_irq_i2c2	High level
	42	i2c_irq_i2c3	High level
	43	i2c_irq_i2c4	High level
	44	i2s_intr_i2s0_8ch	High level
	45	i2s_intr_i2s1_2ch	High level
	46	i2s_intr_i2s2_2ch	High level
	47	UART0_intr	High level
	48	intr_UART1	High level
	49	intr_UART2dbg	High level
	50	intr_UART3	High level
	51	intr_UART4	High level
	52	intr_UART5	High level
	53	otpc_int_otpc_ns	High level
	54	otpc_int_otpc_s	High level
	55	pdm_irq	High level
	56	pwm_int_pwm0	High level
	57	pwm_int_pwm1	High level
	58	spi_intr_spi0	High level
	59	spi_intr_spi1	High level
	60	timer0_int_stimer	High level
	61	timer1_int_stimer	High level
	62	timer0_int_rktimer	High level
	63	timer1_int_rktimer	High level
	64	timer2_int_rktimer	High level
	65	timer3_int_rktimer	High level
	66	timer4_int_rktimer	High level
	67	timer5_int_rktimer	High level
	68	tsadc_int_tsadc	High level
	69	wdtns_irq	High level
	70	wdts_irq	High level
	71	upctl_arpoison_int	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	72	upctl_awpoison_int	High level
	73	upctl_alert_err_int	High level
	74	ddrmon_int	High level
	75	gmac_intr_gmac2io	High level
	76	pmt_intr_gmac2io	High level
	77	irq_gpu	High level
	78	irq_mmu_gpu	High level
	79	irq_job_gpu	High level
	80	irq_event_gpu	High level
	81	irq_dec_mmu	High level
	82	irq_hevc_mmu	High level
	83	Reserved	High level
	84	Reserved	High level
	85	sdmmc_int_emmc	High level
	86	sdmmc_int_sdmmc	High level
	87	sdmmc_int_sdio	High level
	88	sfc_int_sfc	High level
	89	nandc_int_flash	High level
	90	pmu_int	High level
	91	host_arb_int_usb2host	High level
	92	host_ehci_int_usb2host	High level
	93	host_ohci_int_usb2host	High level
	94	otg_int_usb2otg	High level
	95	usbphy_otg_disconnect_irq	High level
	96	usbphy_otg_linestate_irq	High level
	97	usbphy_otg_id_irq	High level
	98	usbphy_otg_bvalid_irq	High level
	99	usbphy_host_disconnect_irq	High level
	100	usbphy_host_linestate_irq	High level
	101	cif_int_out_cif	High level
	102	isp_irq_isp	High level
	103	jpeg_err_irq_isp	High level
	104	jpeg_stat_irq_isp	High level
	105	mi_irq_isp	High level
	106	mipi_irq_isp	High level
	107	mipi_dsi_host_irq_dsihost	High level
	108	rga_irq	High level
	109	vop_intr_vopm	High level
	110	vop_intr_vops	High level
	111	vpu_dec_irq	High level
	112	vpu_enc_irq	High level
	113	vpu_mmu_irq	High level
	114	crypto_irq	High level
	115	otp_mask_int_otpphy	High level
	116	saradc_irq	High level
	117	hwffc_int	High level
	118	irq_isp_mmu_0	High level
	119	irq_isp_mmu_1	High level
	120	irq_isp_mmu_2	High level
	121	pwm_int_pwr_pwm0	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	122	pwm_int_pwr_pwm1	High level
	123	sdmmc_detectn_irq_grf	High level
	124	key_reader_irq	High level
	125	vop_intr_post_lb_vopm	High level
	126	Reserved	High level
	127	Reserved	High level
	128	Reserved	High level
	129	Reserved	High level
	130	Reserved	High level
	131	Reserved	High level
	132	npmuirq[0]	High level
	133	npmuirq[1]	High level
	134	npmuirq[2]	High level
	135	npmuirq[3]	High level
	136	Reserved	High level
	137	Reserved	High level
	138	Reserved	High level
	139	Reserved	High level

1.4 System DMA hardware request connection

PX30 provides one DMA controller inside the system. The trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC, please refer to Chapter 8.

Table 1-2 PX30 DMAC Hardware request connection list

Req Number	Source	Polarity
0	UART0 tx	High level
1	UART0 rx	High level
2	UART1 tx	High level
3	UART1 rx	High level
4	UART2 tx	High level
5	UART2 rx	High level
6	UART3 tx	High level
7	UART3 rx	High level
8	UART4 tx	High level
9	UART4 rx	High level
10	UART5 tx	High level
11	UART5 rx	High level
12	SPI0 tx	High level
13	SPI0 rx	High level
14	SPI1 tx	High level
15	SPI1 rx	High level
16	I2S0_8ch tx	High level
17	I2S0_8ch rx	High level
18	I2S1_2ch_tx	High level
19	I2S1_2ch_rx	High level
20	I2S2_2ch_tx	High level
21	I2S2_2ch_rx	High level
22	pwm0	High level
23	pwm1	High level

Chapter 2 Clock & Reset Unit (CRU)

2.1 Overview

The CRU is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. CRU generates system clocks from PLL output clock or external clock source, and generates system reset from external power-on-reset, watchdog timer reset or software reset or temperature sensor.
CRU supports the following features:
- Compliance to the AMBA APB interface
- Embedded 5 PLLs
- Flexible selection of clock source
- Supports the respective divided clocks
- Supports the respective gating of all clocks
- Supports the respective software reset of all modules

2.2 Block Diagram

CRU comprises with:
- PLL
- Register configuration unit
- Clock generate unit
- Reset generate unit

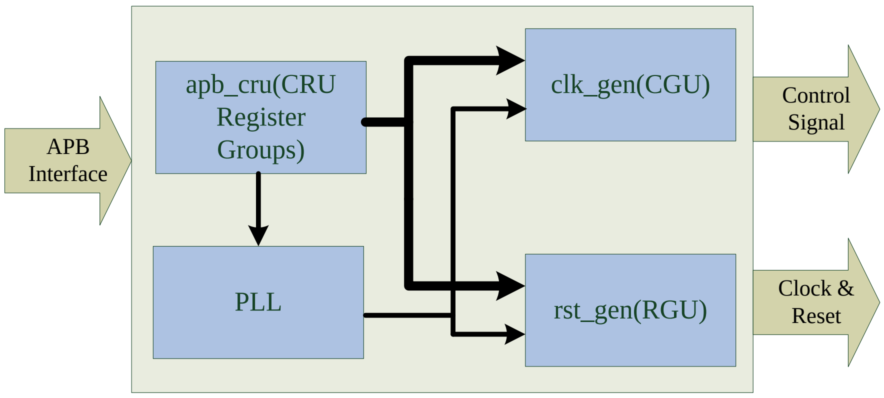
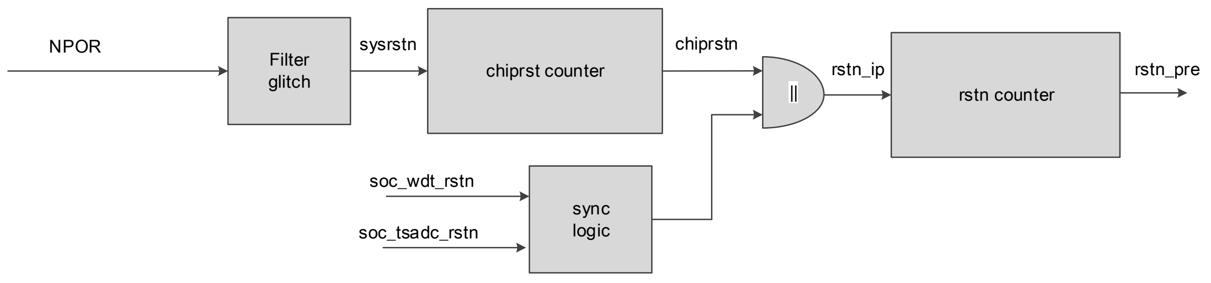


Fig. 2-1 CRU Block Diagram

2.3 System Reset Solution

The following diagram shows reset architecture.



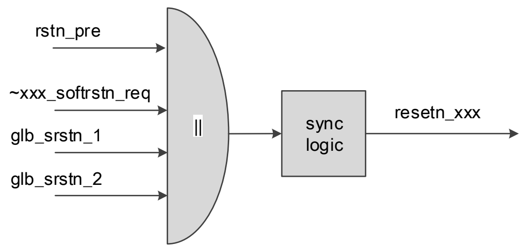


Fig. 2-2 Reset Architecture Diagram

Reset source of each reset signal includes hardware reset(NPOR), SoC watch dog reset(soc_wdt_rstn), SoC tsadc reset(soc_tsadc_rstn), software reset request(xxx_softrstn_req), global software first reset(glb_srstn_1), global software second reset(glb_srstn_2).
The 'xxx' of resetn_xxx and xxx_softrstn_req is the module name.
soc_wdt_rstn is the reset from watch-dog IP in the SoC.
glb_srstn_1 and glb_srstn_2 are the global software reset by programming CRU register. When writing register CRU_GLB_SRST_FST as 0xfdb9, glb_srstn_1 will be asserted, and when writing registerCRU_GLB_SRST_SND as 0xeca8, glb_srstn_2 will be asserted. The two software resets will be self-cleared by hardware. glb_srstn_1 will reset the all logic, and glb_srstn_2 will reset the all logic except GRF, SGRF and all GPIOs.

2.4 Function Description

There are 5 PLLs in the chip: ARM PLL, NEW PLL, DDR PLL, CODEC PLL and GENERAL PLL, and it supports only one crystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.
These 5 PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz. When power on or changing PLL setting, we must force PLL into slow mode or deep slow mode to ensure output stable clock.
To maximize the flexibility, some of clocks can select divider source from multiple PLLs.
To provide some specific frequency, another solution is integrated: fractional divider. In order to guarantee the performance for divided clock, there is some usage limit, we can only get low frequency and divider factor must be larger than 20. For some IP also provide N.5 divisor and duty cycle 50% divisor.
All clocks can be software gated and all resets can be software generated.

2.5 PLL Introduction

2.5.1 Overview

The chip uses up to 3.2GHz PLL for all the PLLs. The 3.2GHz PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments. By combining ultra-low jitter output clocks into a low power, low area, widely programmable design, we can greatly simplify a SoC by enabling a single macro to be used for all clocking applications in the system.

3.2GHz PLL supports the following features:

- Input frequency range:1MHz to 800MHz(Integer Mode) and 10MHz to 800MHz(Fractional Mode)
- Output Frequency Range:16MHz to 3.2GHz
- VCO output clock from 800MHz to 3.2GHz
- 24 bit fractional accuracy,and fractional mode jitter performance to nearly match integer mode performance.
- 4:1 VCO frequency range allows PLL to be optimized for minimum jitter or minimum power.
- Isolated analog supply(1.8V) allows for excellent supply rejection in noisy SoC applications.
- LockDetectSignal indicates when frequency lock has been achieved.

2.5.2 Blockdiagram

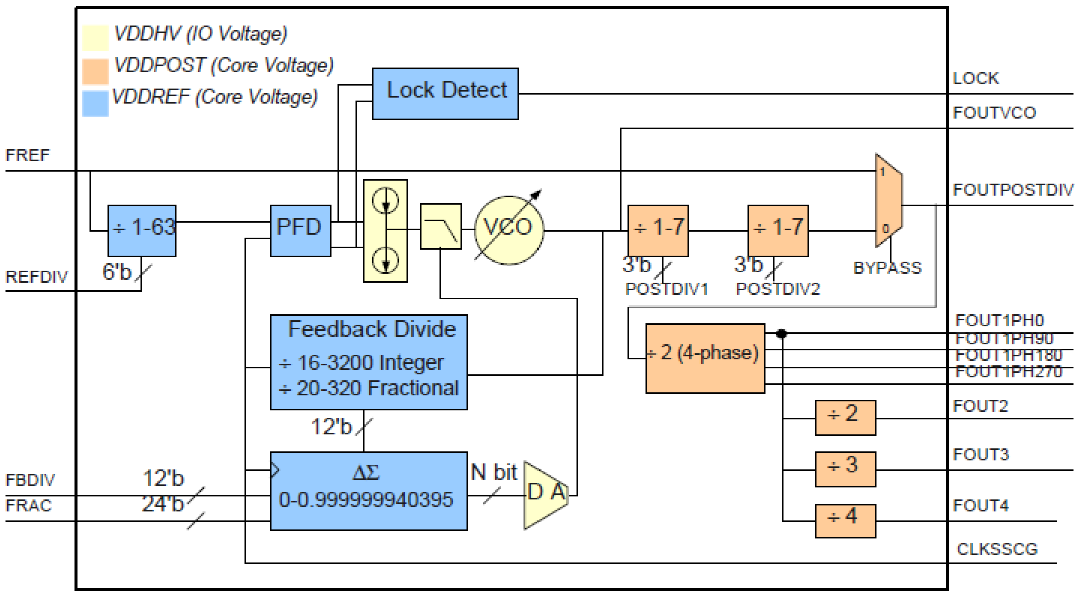


Fig. 2-3 PLLBlockDiagram

How to calculate the PLL

The Fractional PLL output frequency can be calculated using some simple formulas.These formulas also embedded with in the Fractional PLL Verilog model:

If DSMPD=1(DSM is disabled,"integer mode")

FOUTVCO=(FREF/REFDIV)*FBDIV

FOUTPOSTDIV=FOUTVCO/(POSTDIV1*POSTDIV2)

If DSMPD=0(DSM is enabled,"fractional mode")

FOUTVCO=(FREF/REFDIV)*(FBDIV+FRAC/(2^24))

FOUTPOSTDIV=FOUTVCO/(POSTDIV1*POSTDIV2)

Where:

FOUTVCO=Fractional PLL non-divided output frequency

FOUTPOSTDIV=Fractional PLL divided output frequency(output of second postdivider)

FREF=Fractional PLL input reference frequency

REFDIV=Fractional PLL input reference clock divider

FVCO=Frequency of internal VCO

FBDIV=Integer value programmed into feedback divide

FRAC=Fractional value programmed into DSM

Changing the PLL Programming

In most cases the PLL programming can be changed on-the-fly and the PLL will simply slew to the new frequency. However, certain changes have the potential to cause glitches on the PLL output clocks.These changes include:

- Switching into or out of BYPASS mode may cause a glitch on FOUTPOSTDIV
- Changing POSTDIV1 or POSTDIV2 may cause a short pulse with width equal to as little as one VCO period on FOUTPOSTDIV
- Changing POSTDIV could cause a shortened pulse on FOUT1PH*or FOUT2/3/4
- Asserting PD or FOUTPOSTDIVPD may cause a glitch on FOUTPOSTDIV

2.6 Register Description

2.6.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

2.6.2 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x00003064	APLL configuration register0
CRU_APLL_CON1	0x0004	W	0x00001041	APLL configuration register1
CRU_APLL_CON2	0x0008	W	0x00000001	APLL configuration register2
CRU_APLL_CON3	0x000c	W	0x00000007	APLL configuration register3
CRU_APLL_CON4	0x0010	W	0x00007f00	APLL configuration register4
CRU_DPLL_CON0	0x0020	W	0x000010c8	DPLL configuration register0
CRU_DPLL_CON1	0x0024	W	0x00001043	DPLL configuration register1
CRU_DPLL_CON2	0x0028	W	0x00000001	DPLL configuration register2
CRU_DPLL_CON3	0x002c	W	0x00000007	DPLL configuration register3
CRU_DPLL_CON4	0x0030	W	0x00007f00	DPLL configuration register4
CRU_CPLL_CON0	0x0040	W	0x00002063	CPLL configuration register0
CRU_CPLL_CON1	0x0044	W	0x00001041	CPLL configuration register1
CRU_CPLL_CON2	0x0048	W	0x00000001	CPLL configuration register2
CRU_CPLL_CON3	0x004c	W	0x00000007	CPLL configuration register3
CRU_CPLL_CON4	0x0050	W	0x00007f00	CPLL configuration register4
CRU_NPLL_CON0	0x0060	W	0x00002063	NPLL configuration register0
CRU_NPLL_CON1	0x0064	W	0x00001041	NPLL configuration register1
CRU_NPLL_CON2	0x0068	W	0x00000001	NPLL configuration register2
CRU_NPLL_CON3	0x006c	W	0x00000007	NPLL configuration register3
CRU_NPLL_CON4	0x0070	W	0x00007f00	NPLL configuration register4
CRU_MODE	0x00a0	W	0x00000000	MODE
CRU_MISC	0x00a4	W	0x00000000	MISC
CRU_GLB_CNT_TH	0x00b0	W	0x3a980064	GLB_CNT_TH
CRU_GLB_RST_ST	0x00b4	W	0x00000000	GLB_RST_ST
CRU_GLB_SRST_FST	0x00b8	W	0x00000000	GLB_SRST_FST
CRU_GLB_SRST_SND	0x00bc	W	0x00000000	GLB_SRST_SND
CRU_GLB_RST_CON	0x00c0	W	0x00000000	GLB_RST_CON
CRU_CLKSEL_CON0	0x0100	W	0x00001300	Clock select and divide register0
CRU_CLKSEL_CON1	0x0104	W	0x00000202	Clock select and divide register1
CRU_CLKSEL_CON2	0x0108	W	0x00000b00	Clock select and divide register2
CRU_CLKSEL_CON3	0x010c	W	0x00002103	Clock select and divide register3
CRU_CLKSEL_CON4	0x0110	W	0x00000003	Clock select and divide register4

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON5	0x0114	W	0x00000007	Clock select and divide register5
CRU_CLKSEL_CON6	0x0118	W	0x0bb8ea60	Clock select and divide register6
CRU_CLKSEL_CON7	0x011c	W	0x0000000b	Clock select and divide register7
CRU_CLKSEL_CON8	0x0120	W	0x00000007	Clock select and divide register8
CRU_CLKSEL_CON9	0x0124	W	0x0bb8ea60	Clock select and divide register9
CRU_CLKSEL_CON10	0x0128	W	0x00000103	Clock select and divide register10
CRU_CLKSEL_CON11	0x012c	W	0x00000103	Clock select and divide register11
CRU_CLKSEL_CON12	0x0130	W	0x00001702	Clock select and divide register12
CRU_CLKSEL_CON13	0x0134	W	0x00000600	Clock select and divide register13
CRU_CLKSEL_CON14	0x0138	W	0x00000705	Clock select and divide register14
CRU_CLKSEL_CON15	0x013c	W	0x00000707	Clock select and divide register15
CRU_CLKSEL_CON16	0x0140	W	0x00000003	Clock select and divide register16
CRU_CLKSEL_CON17	0x0144	W	0x00000003	Clock select and divide register17
CRU_CLKSEL_CON18	0x0148	W	0x00000002	Clock select and divide register18
CRU_CLKSEL_CON19	0x014c	W	0x00000002	Clock select and divide register19
CRU_CLKSEL_CON20	0x0150	W	0x00000002	Clock select and divide register20
CRU_CLKSEL_CON21	0x0154	W	0x00000002	Clock select and divide register21
CRU_CLKSEL_CON22	0x0158	W	0x0000170b	Clock select and divide register22
CRU_CLKSEL_CON23	0x015c	W	0x00000581	Clock select and divide register23
CRU_CLKSEL_CON24	0x0160	W	0x00000107	Clock select and divide register24
CRU_CLKSEL_CON25	0x0164	W	0x00000305	Clock select and divide register25
CRU_CLKSEL_CON26	0x0168	W	0x0000000b	Clock select and divide register26
CRU_CLKSEL_CON27	0x016c	W	0x0bb8ea60	Clock select and divide register27
CRU_CLKSEL_CON28	0x0170	W	0x0000000b	Clock select and divide register28
CRU_CLKSEL_CON29	0x0174	W	0x0bb8ea60	Clock select and divide register29
CRU_CLKSEL_CON30	0x0178	W	0x0000000b	Clock select and divide register30
CRU_CLKSEL_CON31	0x017c	W	0x0bb8ea60	Clock select and divide register31
CRU_CLKSEL_CON32	0x0180	W	0x0000000b	Clock select and divide register32
CRU_CLKSEL_CON33	0x0184	W	0x0bb8ea60	Clock select and divide register33
CRU_CLKSEL_CON34	0x0188	W	0x0000000b	Clock select and divide register34
CRU_CLKSEL_CON35	0x018c	W	0x0000000b	Clock select and divide register35
CRU_CLKSEL_CON36	0x0190	W	0x0bb8ea60	Clock select and divide register36
CRU_CLKSEL_CON37	0x0194	W	0x0000000b	Clock select and divide register37
CRU_CLKSEL_CON38	0x0198	W	0x0000000b	Clock select and divide register38
CRU_CLKSEL_CON39	0x019c	W	0x0bb8ea60	Clock select and divide register39
CRU_CLKSEL_CON40	0x01a0	W	0x0000000b	Clock select and divide register40
CRU_CLKSEL_CON41	0x01a4	W	0x0000000b	Clock select and divide register41
CRU_CLKSEL_CON42	0x01a8	W	0x0bb8ea60	Clock select and divide register42
CRU_CLKSEL_CON43	0x01ac	W	0x0000000b	Clock select and divide register43
CRU_CLKSEL_CON44	0x01b0	W	0x0000000b	Clock select and divide register44
CRU_CLKSEL_CON45	0x01b4	W	0x0bb8ea60	Clock select and divide register45
CRU_CLKSEL_CON46	0x01b8	W	0x0000000b	Clock select and divide register34

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON47	0x01bc	W	0x0000000b	Clock select and divide register47
CRU_CLKSEL_CON48	0x01c0	W	0x0bb8ea60	Clock select and divide register48
CRU_CLKSEL_CON49	0x01c4	W	0x00000b0b	Clock select and divide register49
CRU_CLKSEL_CON50	0x01c8	W	0x00000b0b	Clock select and divide register50
CRU_CLKSEL_CON52	0x01d0	W	0x00000b0b	Clock select and divide register52
CRU_CLKSEL_CON53	0x01d4	W	0x00000b0b	Clock select and divide register49
CRU_CLKSEL_CON54	0x01d8	W	0x00000001	Clock select and divide register43
CRU_CLKSEL_CON55	0x01dc	W	0x00000017	Clock select and divide register44
CRU_CLKSEL_CON56	0x01e0	W	0x00000010	Clock select and divide register45
CRU_CLKSEL_CON57	0x01e4	W	0x00001f00	Clock select and divide register57
CRU_CLKSEL_CON58	0x01e8	W	0x0000000b	Clock select and divide register58
CRU_CLKSEL_CON59	0x01ec	W	0x0bb8ea60	Clock select and divide register59
CRU_CLKGATE_CON0	0x0200	W	0x00000000	Clock gating register0
CRU_CLKGATE_CON1	0x0204	W	0x00000000	Clock gating register1
CRU_CLKGATE_CON2	0x0208	W	0x00000000	Clock gating register2
CRU_CLKGATE_CON3	0x020c	W	0x00000000	Clock gating register3
CRU_CLKGATE_CON4	0x0210	W	0x00000000	Clock gating register4
CRU_CLKGATE_CON5	0x0214	W	0x00000000	Clock gating register5
CRU_CLKGATE_CON6	0x0218	W	0x00000000	Clock gating register6
CRU_CLKGATE_CON7	0x021c	W	0x00000000	Clock gating register7
CRU_CLKGATE_CON8	0x0220	W	0x00000000	Clock gating register8
CRU_CLKGATE_CON9	0x0224	W	0x00000000	Clock gating register9
CRU_CLKGATE_CON10	0x0228	W	0x00000000	Clock gating register10
CRU_CLKGATE_CON11	0x022c	W	0x00000000	Clock gating register11
CRU_CLKGATE_CON12	0x0230	W	0x00000000	Clock gating register12
CRU_CLKGATE_CON13	0x0234	W	0x00000000	Clock gating register13
CRU_CLKGATE_CON14	0x0238	W	0x00000000	Clock gating register14
CRU_CLKGATE_CON15	0x023c	W	0x00000000	Clock gating register15
CRU_CLKGATE_CON16	0x0240	W	0x00000000	Clock gating register16
CRU_CLKGATE_CON17	0x0244	W	0x00000000	Clock gating register17
CRU_SSGTBL0_3	0x0280	W	0x00000000	External wave table register0
CRU_SSGTBL4_7	0x0284	W	0x00000000	External wave table register1
CRU_SSGTBL8_11	0x0288	W	0x00000000	External wave table register2
CRU_SSGTBL12_15	0x028c	W	0x00000000	External wave table register3
CRU_SSGTBL16_19	0x0290	W	0x00000000	External wave table register4
CRU_SSGTBL20_23	0x0294	W	0x00000000	External wave table register5
CRU_SSGTBL24_27	0x0298	W	0x00000000	External wave table register6
CRU_SSGTBL28_31	0x029c	W	0x00000000	External wave table register7
CRU_SSGTBL32_35	0x02a0	W	0x00000000	External wave table register8
CRU_SSGTBL36_39	0x02a4	W	0x00000000	External wave table register9
CRU_SSGTBL40_43	0x02a8	W	0x00000000	External wave table register10
CRU_SSGTBL44_47	0x02ac	W	0x00000000	External wave table register11

Name	Offset	Size	Reset Value	Description
CRU_SSGTBL48_51	0x02b0	W	0x00000000	External wave table register12
CRU_SSGTBL52_55	0x02b4	W	0x00000000	External wave table register13
CRU_SSGTBL56_59	0x02b8	W	0x00000000	External wave table register14
CRU_SSGTBL60_63	0x02bc	W	0x00000000	External wave table register15
CRU_SSGTBL64_67	0x02c0	W	0x00000000	External wave table register16
CRU_SSGTBL68_71	0x02c4	W	0x00000000	External wave table register17
CRU_SSGTBL72_75	0x02c8	W	0x00000000	External wave table register18
CRU_SSGTBL76_79	0x02cc	W	0x00000000	External wave table register19
CRU_SSGTBL80_83	0x02d0	W	0x00000000	External wave table register20
CRU_SSGTBL84_87	0x02d4	W	0x00000000	External wave table register21
CRU_SSGTBL88_91	0x02d8	W	0x00000000	External wave table register22
CRU_SSGTBL92_95	0x02dc	W	0x00000000	External wave table register23
CRU_SSGTBL96_99	0x02e0	W	0x00000000	External wave table register24
CRU_SSGTBL100_103	0x02e4	W	0x00000000	External wave table register25
CRU_SSGTBL104_107	0x02e8	W	0x00000000	External wave table register26
CRU_SSGTBL108_111	0x02ec	W	0x00000000	External wave table register27
CRU_SSGTBL112_115	0x02f0	W	0x00000000	External wave table register28
CRU_SSGTBL116_119	0x02f4	W	0x00000000	External wave table register29
CRU_SSGTBL120_123	0x02f8	W	0x00000000	External wave table register30
CRU_SSGTBL124_127	0x02fc	W	0x00000000	External wave table register31
CRU_SOFTRST_CON0	0x0300	W	0x00000000	Software reset control register0
CRU_SOFTRST_CON1	0x0304	W	0x00000000	Software reset control register1
CRU_SOFTRST_CON2	0x0308	W	0x00000000	Software reset control register2
CRU_SOFTRST_CON3	0x030c	W	0x00000000	Software reset control register3
CRU_SOFTRST_CON4	0x0310	W	0x00000000	Software reset control register4
CRU_SOFTRST_CON5	0x0314	W	0x00000000	Software reset control register5
CRU_SOFTRST_CON6	0x0318	W	0x00000000	Software reset control register6
CRU_SOFTRST_CON7	0x031c	W	0x00000000	Software reset control register7
CRU_SOFTRST_CON8	0x0320	W	0x00000000	Software reset control register8
CRU_SOFTRST_CON9	0x0324	W	0x00000000	Software reset control register9
CRU_SOFTRST_CON10	0x0328	W	0x00000000	Software reset control register10
CRU_SOFTRST_CON11	0x032c	W	0x00000000	Software reset control register11
CRU_SDMMC_CON0	0x0380	W	0x00000004	SDMMC control0
CRU_SDMMC_CON1	0x0384	W	0x00000000	SDMMC control1
CRU_SDIO_CON0	0x0388	W	0x00000004	SDIO control0
CRU_SDIO_CON1	0x038c	W	0x00000000	SDIO control1
CRU_EMMC_CON0	0x0390	W	0x00000004	EMMC control0
CRU_EMMC_CON1	0x0394	W	0x00000000	EMMC control1
CRU_GPLL_CON0	0xc000	W	0x00001032	GPLL configuration register0
CRU_GPLL_CON1	0xc004	W	0x00001041	GPLL configuration register1
CRU_GPLL_CON2	0xc008	W	0x00000001	GPLL configuration register2
CRU_GPLL_CON3	0xc00c	W	0x00000007	GPLL configuration register3

Name	Offset	Size	Reset Value	Description
CRU_GPLL_CON4	0xc010	W	0x00007f00	GPLL configuration register4
CRU_PMU_MODE	0xc020	W	0x00000000	PMU_MODE
CRU_PMU_CLKSEL_CON0	0xc040	W	0x0000000b	PMU Clock select and divide register0
CRU_PMU_CLKSEL_CON1	0xc044	W	0x0bb8ea60	PMU Clock select and divide register0
CRU_PMU_CLKSEL_CON2	0xc048	W	0x00003131	PMU Clock select and divide register2
CRU_PMU_CLKSEL_CON3	0xc04c	W	0x0000000b	PMU Clock select and divide register3
CRU_PMU_CLKSEL_CON4	0xc050	W	0x0000000b	PMU Clock select and divide register4
CRU_PMU_CLKSEL_CON5	0xc054	W	0x0bb8ea60	PMU Clock select and divide register5
CRU_PMU_CLKGATE_CON0	0xc080	W	0x00000000	PMU Clock gating register0
CRU_PMU_CLKGATE_CON1	0xc084	W	0x00000000	PMU Clock gating register1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

2.6.3 Detail Register Description

CRU_APLL_CON0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x3	postdiv1 First Post Divide Value, (1-7)
11:0	RW	0x064	fbdiv Feedback Divide Value, valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_APLL_CON1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value, (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value, (1-63)

CRU_APLL_CON2
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_APLL_CON3
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	WO	0x0	ssmod_divval Divider required to set the modulation frequency
3	WO	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_APLL_CON4
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_DPLL_CON0
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 First Post Divide Value, (1-7)
11:0	RW	0x0c8	fbdiv Feedback Divide Value, valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_DPLL_CON1
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted

Bit	Attr	Reset Value	Description
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x03	refdiv Reference Clock Divide Value (1-63)

CRU_DPLL_CON2
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
23:0	RW	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_DPLL_CON3
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	WO	0x0	ssmod_divval Divider required to set the modulation frequency
3	WO	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_DPLL_CON4
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr External wave table data inputs, (0-255)
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	WO	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_CPLL_CON0
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x2	postdiv1 First Post Divide Value, (1-7)
11:0	RW	0x063	fbdiv Feedback Divide Value, valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_CPLL_CON1
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
12	RW	0x1	dsmpd PLL delta sigma modulator enable 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_CPLL_CON2
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_CPLL_CON3
Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	WO	0x0	ssmod_divval Divider required to set the modulation frequency
3	WO	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_CPLL_CON4
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_NPLL_CON0
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x2	postdiv1 First Post Divide Value, (1-7)
11:0	RW	0x063	fbdiv Feedback Divide Value, valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_NPLL_CON1
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_NPLL_CON2
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_NPLL_CON3
Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	RW	0x0	ssmod_divval Divider required to set the modulation frequency

Bit	Attr	Reset Value	Description
3	RW	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread
2	RW	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	RW	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_NPLL_CON4
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_MODE
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	usbphy480m_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k

Bit	Attr	Reset Value	Description
7:6	RW	0x0	npll_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k
5:4	RW	0x0	dpll_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k
3:2	RW	0x0	cpll_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k
1:0	RW	0x0	apll_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k

CRU_MISC
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RW	0x0	core_high_freq_rst_en 1'b1:enable high frequency rst gate function 1'b0:disable high frequency rst gate function. Each bit for each core, eg. bit0 for core0
11:5	RO	0x0	reserved
4	RW	0x0	corepo_wrst_wfien 1'b1: enable core0/1/2/3 warm reset for cpu power on reset. 1'b0: disable core0/1/2/3 warm reset for cpu power on reset
3	RW	0x0	corepo_srst_wfien 1'b1: enable core0/1/2/3 wfi reset for cpu power on reset 1'b0: disable core0/1/2/3 wif reset for cpu power on reset
2	RW	0x0	core_wrst_wfien 1'b1: enable core0/1/2/3 warm reset for cpu reset. 1'b0: disable core0/1/2/3 warm reset for cpu reset
1	RW	0x0	core_srst_wfien 1'b1: enable core0/1/2/3 wfi reset for cpu reset 1'b0: disable core0/1/2/3 wif reset for cpu reset
0	RW	0x0	warmrst_en 1'b1: enable cpu warm reset. 1'b0: disable cpu warm reset

CRU_GLB_CNT_TH
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:16	RW	0x3a98	pll_lockperiod PLL lock filtered period time, measured in OSC clock cycles
15:0	RW	0x0064	global_reset_counter_threshold Global soft reset, wdt reset or tsadc_shut reset asserted time counter threshold. Measured in OSC clock cycles

CRU_GLB_RST_ST
Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:20	RO	0x0	resetn_corepo_src_st corepo resetn source status of core0~3. Each bit for each core
19:16	RO	0x0	resetn_core_src_st core resetn source status of core0~3. Each bit for each core
15:6	RO	0x0	reserved
5	W1C	0x0	snd_glb_tsadc_rst_st sencond global TSADC triggered reset flag 1'b0: last hot reset is not sencond global TSADC triggered reset 1'b1: last hot reset is sencond global TSADC triggered reset
4	W1C	0x0	fst_glb_tsadc_rst_st first global TSADC triggered reset flag 1'b0: last hot reset is not first global TSADC triggered reset 1'b1: last hot reset is first global TSADC triggered reset
3	W1C	0x0	snd_glb_wdt_rst_st sencond global WDT triggered reset flag 1'b0: last hot reset is not sencond global WDT triggered reset 1'b1: last hot reset is sencond global WDT triggered reset
2	W1C	0x0	fst_glb_wdt_rst_st first global WDT triggered reset flag 1'b0: last hot reset is not first global WDT triggered reset 1'b1: last hot reset is first global WDT triggered reset
1	W1C	0x0	snd_glb_rst_st second global rst flag 1'b0: last hot reset is not sencond global reset 1'b1: last hot reset is sencond global reset
0	W1C	0x0	fst_glb_rst_st first global rst flag 1'b0: last hot reset is not first global reset 1'b1: last hot reset is first global reset

CRU_GLB_SRST_FST
Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GLB_SRST_FST The first global software reset config value

CRU_GLB_SRST_SND
Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GLB_SRST_SND The second global software reset config value

CRU_GLB_RST_CON
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	wdt_reset_ext_en 1'b1: enable wdt reset extend, reset extend time depend on bit15~0 of GLB_CNT_TH 1'b0: disable wdt reset extend
6	RW	0x0	tsadc_shut_reset_ext_en 1'b1: enable tsadc_shut reset extend, reset extend time depend on bit15~0 of GLB_CNT_TH 1'b0: disable tsadc_shut reset extend
5	RO	0x0	reserved
4	RW	0x0	pmu_srst_wdt_en 1'b0: enable wdt reset as pmu reset source 1'b1: disable wdt reset as pmu reset source
3	RW	0x0	pmu_srst_glb_rst_en 1'b0: enable first or second global reset as pmu reset source 1'b1: disable first or second global reset as pmu reset source
2	RW	0x0	pmu_srst_ctrl 1'b1: second global reset trigger pmu reset 1'b0: first global reset trigger pmu reset
1	RW	0x0	wdt_glb_srst_ctrl 1'b0: wdt trigger second global reset 1'b1: wdt trigger first global reset
0	RW	0x0	tsadc_glb_srst_ctrl 1'b0: tsadc trigger second global reset 1'b1: tsadc trigger first global reset

CRU_CLKSEL_CON0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	aclk_core_div_con aclk_core=clk_core/(div_con+1)
11:8	RW	0x3	core_dbg_div_con pclk_dbg=clk_core/(div_con+1)
7	RW	0x0	core_clk_pll_sel 1'b0:APLL 1'b1:GPLL
6:4	RO	0x0	reserved
3:0	RW	0x0	clk_core_div_con clk_core=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_gpu_sel 1'b0: select clk_gpu_div 1'b1: select clk_gpu_np5
14:13	RW	0x0	aclk_gpu_div_con aclk_gpu=clk_gpu/(div_con+1)
12	RO	0x0	reserved
11:8	RW	0x2	clk_gpu_divnp5_con clk_gpu_np5=2*clk_gpu_div/(2*div_con+3)
7:6	RW	0x0	clk_gpu_pll_sel 2'h0:GPLL 2'h1:CPLL 2'h2:usbphy480M 2'h3:NPLL
5:4	RO	0x0	reserved
3:0	RW	0x2	clk_gpu_div_con clk_gpu_div=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x0b	pclk_ddr_div_con pclk_ddr=gpll_clk_src/(div_con+1)
7	RW	0x0	ddrphy4x_pll_clk_sel 1'b0: DPLL 1'b1: GPLL
6:5	RO	0x0	reserved
4	RW	0x0	clk_ddrstdby_sel 1'b0: select ddrphy1x as clk_ddrstdby clock 1'b1: select ddrphy4x/4 as clk_ddrstdby clock
3	RO	0x0	reserved
2:0	RW	0x0	ddrphy4x_div_con clk_ddrphy4x=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON3
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RW	0x2	pclk_vo_div_con pclk_vo=aclk_vo/(div_con+1)
11:8	RW	0x1	hclk_vo_div_con hclk_vo=aclk_vo/(div_con+1)
7:6	RW	0x0	aclk_vo_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x03	aclk_vo_div_con aclk_vo=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON4
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	clk_rga_core_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x03	clk_rga_core_div_con clk_rga_core=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON5
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	dclk_vopb_sel 2'b0: select dclk_vopb 2'b1: select dclk_vopb_frac_out 2'b2: select xin_osc0
13:12	RO	0x0	reserved
11	RW	0x0	dclk_vopb_pll_sel 1'b0:CPLL 1'b1:NPLL
10:8	RO	0x0	reserved
7:0	RW	0x07	dclk_vopb_div_con dclk_vopb=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON6
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	dclk_vopb_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is dclk_vopb

CRU_CLKSEL_CON7
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	clk_pwm_vopb_pll_sel 1'b0:GPLL 1'b1:xin_osc0
6:0	RW	0x0b	clk_pwm_vopb_div_con clk_pwm_vopb=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON8
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	dclk_vopl_sel 2'b0: select dclk_vopl 2'b1: select dclk_vopl_frac_out 2'b2: select xin_osc0
13:12	RO	0x0	reserved
11	RW	0x0	dclk_vopl_pll_sel 1'b0:NPLL 1'b1:CPLL
10:8	RO	0x0	reserved
7:0	RW	0x07	dclk_vopl_div_con dclk_vopl=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON9
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	dclk_vopl_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is dclk_vopl

CRU_CLKSEL_CON10
Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:8	RW	0x1	hclk_vpu_div_con hclk_vpu=aclk_vpu/(div_con+1)
7:6	RW	0x0	aclk_vpu_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x03	aclk_vpu_div_con aclk_vpu=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON11
Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:8	RW	0x1	hclk_vi_div_con hclk_vi=aclk_vi/(div_con+1)
7:6	RW	0x0	aclk_vi_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x03	aclk_vi_div_con aclk_vi=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON12
Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_gmac_out_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
13	RO	0x0	reserved
12:8	RW	0x17	clk_gmac_out_div_con clk_gmac_out=pll_clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	clk_isp_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x02	clk_isp_div_con clk_isp=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON13
Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_vpu_core_pll_clk_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
13	RO	0x0	reserved
12:8	RW	0x06	clk_vpu_core_div_con clk_vpu_core=pll_clk_src/(div_con+1)
7:6	RW	0x0	clk_cif_out_pll_sel 2'b0:xin_osc0 2'b1:CPLL 2'b2:NPLL 2'b3:usbphy480M
5:0	RW	0x00	clk_cif_out_div_con clk_cif_out=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON14
Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_hclk_peri_pll_sel 1'b0:GPLL 1'b1:CPLL
14:13	RO	0x0	reserved
12:8	RW	0x07	hclk_peri_div_con hclk_peri=pll_clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x05	aclk_peri_div_con aclk_peri=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON15
Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_nandc_sel 1'b0: select clk_nandc 1'b1: select clk_nandc_div50
14:13	RO	0x0	reserved
12:8	RW	0x07	clk_nandc_div50_div_con clk_nandc_div50=clk_nandc/(div_con+1), duty cycle is 50% for any value
7:6	RW	0x0	clk_nandc_pll 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL
5	RO	0x0	reserved
4:0	RW	0x07	clk_nandc_div_con clk_nandc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON16
Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_sdmmc_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL 2'b3:xin_osc0
13:8	RO	0x0	reserved
7:0	RW	0x03	clk_sdmmc_div_con clk_sdmmc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON17
Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_sdmmc_sel 1'b0:select clk_sdmmc 1'b1:select clk_sdmmc_div50
14:8	RO	0x0	reserved
7:0	RW	0x03	clk_sdmmc_div50_div_con clk_sdmmc_div50=clk_sdmmc/(div_con+1), duty cycle is 50% for any value

CRU_CLKSEL_CON18
Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_sdio_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL 2'b3:xin_osc0
13:8	RO	0x0	reserved
7:0	RW	0x02	clk_sdio_div_con clk_sdio=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON19
Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sdio_sel 1'b0:select clk_sdio 1'b1:select clk_sdio_div50
14:8	RO	0x0	reserved
7:0	RW	0x02	clk_sdio_div50_div_con clk_sdio_div50=clk_sdio/(div_con+1), duty cycle is 50% for any value

CRU_CLKSEL_CON20
Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	clk_emmc_pll_sel 2'b0:GPLL 2'b1:CPLL 2'b2:NPLL 2'b3:xin_osc0
13:8	RO	0x0	reserved
7:0	RW	0x02	clk_emmc_div_con clk_emmc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON21
Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_emmc_sel 1'b0:select clk_emmc 1'b1:select clk_emmc_div50
14:8	RO	0x0	reserved
7:0	RW	0x02	clk_emmc_div50_div_con clk_emmc_div50=clk_emmc/(div_con+1), duty cycle is 50% for any value

CRU_CLKSEL_CON22
Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_gmac_pll_sel 2'h0:GPLL 2'h1:CPLL 2'h2:NPLL
13	RO	0x0	reserved
12:8	RW	0x17	clk_gmac_div_con clk_gmac=pll_clk_src/(div_con+1)
7	RW	0x0	clk_sfc_pll_sel 1'b0:GPLL 1'b1:CPLL
6:0	RW	0x0b	clk_sfc_div_con clk_sfc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON23
Address: Operational Base + offset (0x015c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_hclk_pclk_bus_pll_sel 1'b0:GPLL 1'b1:CPLL
14:13	RO	0x0	reserved
12:8	RW	0x05	aclk_bus_div_con aclk_bus=pll_clk_src/(div_con+1)
7	RW	0x1	rmii_clk_sel 1'b0:10M 1'b1:100M
6	RW	0x0	rmii_extclksrc_sel 1'b0:select clk_gmac as clk_gmac 1'b1:select external phy clock as clk_gmac
5:4	RO	0x0	reserved
3:0	RW	0x1	pclk_gmac_div_con pclk_gmac=aclk_peri/(div_con+1)

CRU_CLKSEL_CON24
Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x1	pclk_bus_div_con pclk_bus=aclk_bus/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x07	hclk_bus_div_con hclk_bus=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON25
Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_crypto_apk_sel 2'h0:GPLL 2'h1:CPLL 2'h2:NPLL

Bit	Attr	Reset Value	Description
13	RO	0x0	reserved
12:8	RW	0x03	clk_crypto_apk_div_con clk_crypto_apk=pll_clk_src/(div_con+1)
7:6	RW	0x0	clk_crypto_pll_sel 2'h0:GPLL 2'h1:CPLL 2'h2:NPLL
5	RO	0x0	reserved
4:0	RW	0x05	clk_crypto_div_con clk_crypto=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON26
Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_pdm_sel 1'b0:select clk_pdm 1'b1:select clk_pdm_frac_out
14:10	RO	0x0	reserved
9:8	RW	0x0	clk_pdm_pll_sel 2'h0:GPLL 2'h1:xin_osc0 2'h2:NPLL
7	RO	0x0	reserved
6:0	RW	0x0b	clk_pdm_div_con clk_pdm=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON27
Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_pdm_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_pdm

CRU_CLKSEL_CON28
Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_i2s0_tx_out_mclk_sel 2'h0:select selected clock by clk_i2s0_tx_rx_clk_sel 2'h1:select xin_osc0_half 2'h2:select clk_i2s0_rx
13	RO	0x0	reserved
12	RW	0x0	clk_i2s0_tx_rx_clk_sel 1'b0: select clk_i2s0_tx_clk 1'b1: select clk_i2s0_rx_clk
11:10	RW	0x0	clk_i2s0_tx_sel 2'h0:select clk_i2s0_tx 2'h1:select clk_i2s0_tx_frac_out 2'h2:select mclk_i2s0_tx_in 2'h3:select xin_osc0_half
9	RO	0x0	reserved
8	RW	0x0	clk_i2s0_tx_pll_sel 1'b0:GPLL 1'b1:NPLL
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2s0_tx_div_con clk_i2s0_tx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON29
Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s0_tx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s0_tx

CRU_CLKSEL_CON30
Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s1_out_mclk_sel 1'b0:select selected clock by clk_i2s1_sel 1'b1:select xin_osc0_half
14:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x0	clk_i2s1_sel 2'h0:select clk_i2s1 2'h1:select clk_i2s1_frac_out 2'h2:select mclk_i2s1_in 2'h3:select xin_osc0_half
9	RO	0x0	reserved
8	RW	0x0	clk_i2s1_pll_sel 1'b0:GPLL 1'b1:NPLL
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2s1_div_con clk_i2s1=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON31
Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s1_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s1

CRU_CLKSEL_CON32
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s2_out_mclk_sel 1'b0:select selected clock by clk_i2s2_sel 1'b1:select xin_osc0_half
14:12	RO	0x0	reserved
11:10	RW	0x0	clk_i2s2_sel 2'h0:select clk_i2s2 2'h1:select clk_i2s2_frac_out 2'h2:select mclk_i2s2_in 2'h3:select xin_osc0_half
9	RO	0x0	reserved
8	RW	0x0	clk_i2s2_pll_sel 1'b0:GPLL 1'b1:NPLL
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2s2_div_con clk_i2s2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s2_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s2

CRU_CLKSEL_CON34
Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart1_pll_sel 2'b0:GPLL 2'b1:xin_osc0 2'b2:usbphy480M 2'b3:NPLL
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart1_div_con clk_uart1=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON35
Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart1_sel 2'b0:select clk_uart1 2'b1:select clk_uart1_np5 2'b2:select clk_uart1_frac_out
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart1_divnp5_div_con clk_uart1_np5=2*clk_uart1/(2*div_con+3)

CRU_CLKSEL_CON36
Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart1_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart1

CRU_CLKSEL_CON37
Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart2_pll_sel 2'b0:GPLL 2'b1:xin_osc0 2'b2:usbphy480M 2'b3:NPLL
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart2_div_con clk_uart2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON38
Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart2_sel 2'b0:select clk_uart2 2'b1:select clk_uart2_np5 2'b2:select clk_uart2_frac_out
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart2_divnp5_div_con clk_uart2_np5=2*clk_uart2/(2*div_con+3)

CRU_CLKSEL_CON39
Address: Operational Base + offset (0x019c)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart2_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart2

CRU_CLKSEL_CON40
Address: Operational Base + offset (0x01a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart3_pll_sel 2'b0:GPLL 2'b1:xin_osc0 2'b2:usbphy480M 2'b3:NPLL
13:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x0b	clk_uart3_div_con clk_uart3=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON41
Address: Operational Base + offset (0x01a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart3_sel 2'b0:select clk_uart3 2'b1:select clk_uart3_np5 2'b2:select clk_uart3_frac_out
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart3_divnp5_div_con clk_uart3_np5=2*clk_uart3/(2*div_con+3)

CRU_CLKSEL_CON42
Address: Operational Base + offset (0x01a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart3_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart3

CRU_CLKSEL_CON43
Address: Operational Base + offset (0x01ac)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart4_pll_sel 2'b0:GPLL 2'b1:xin_osc0 2'b2:usbphy480M 2'b3:NPLL
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart4_div_con clk_uart4=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON44
Address: Operational Base + offset (0x01b0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart4_sel 2'b0:select clk_uart4 2'b1:select clk_uart4_np5 2'b2:select clk_uart4_frac_out
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart4_divnp5_div_con clk_uart4_np5=2*clk_uart4/(2*div_con+3)

CRU_CLKSEL_CON45
Address: Operational Base + offset (0x01b4)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart4_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart4

CRU_CLKSEL_CON46
Address: Operational Base + offset (0x01b8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart5_pll_sel 2'b0:GPLL 2'b1:xin_osc0 2'b2:usbphy480M 2'b3:NPLL
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart5_div_con clk_uart5=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON47
Address: Operational Base + offset (0x01bc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart5_sel 2'b0:select clk_uart5 2'b1:select clk_uart5_np5 2'b2:select clk_uart5_frac_out
13:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x0b	clk_uart5_divnp5_div_con clk_uart5_np5=2*clk_uart5/(2*div_con+3)

CRU_CLKSEL_CON48
Address: Operational Base + offset (0x01c0)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart5_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart5

CRU_CLKSEL_CON49
Address: Operational Base + offset (0x01c4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c1_pll_sel 1'b0:GPLL 1'b1:xin_osc0
14:8	RW	0x0b	clk_i2c1_div_con clk_i2c1=pll_clk_src/(div_con+1)
7	RW	0x0	clk_i2c0_pll_sel 1'b0:GPLL 1'b1:xin_osc0
6:0	RW	0x0b	clk_i2c0_div_con clk_i2c0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON50
Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c3_pll_sel 1'b0:GPLL 1'b1:xin_osc0
14:8	RW	0x0b	clk_i2c3_div_con clk_i2c3=pll_clk_src/(div_con+1)
7	RW	0x0	clk_i2c2_pll_sel 1'b0:GPLL 1'b1:xin_osc0
6:0	RW	0x0b	clk_i2c2_div_con clk_i2c2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON52
Address: Operational Base + offset (0x01d0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_pwm1_pll_sel 1'b0:GPLL 1'b1:xin_osc0
14:8	RW	0x0b	clk_pwm1_div_con clk_pwm1=pll_clk_src/(div_con+1)
7	RW	0x0	clk_pwm0_pll_sel 1'b0:GPLL 1'b1:xin_osc0
6:0	RW	0x0b	clk_pwm0_div_con clk_pwm0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON53
Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spi1_pll_sel 1'b0:GPLL 1'b1:xin_osc0
14:8	RW	0x0b	clk_spi1_div_con clk_spi1=pll_clk_src/(div_con+1)
7	RW	0x0	clk_spi0_pll_sel 1'b0:GPLL 1'b1:xin_osc0
6:0	RW	0x0b	clk_spi0_div_con clk_spi0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON54
Address: Operational Base + offset (0x01d8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:0	RW	0x001	clk_tsadc_div_con clk_tsadc=xin_osc0/(div_con+1)

CRU_CLKSEL_CON55
Address: Operational Base + offset (0x01dc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10:0	RW	0x017	clk_saradc_div_con clk_saradc=xin_osc0/(div_con+1)

CRU_CLKSEL_CON56
Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5:4	RW	0x1	clk_otp_usr_div_con clk_otp_usr=clk_otp/(div_con+1)
3	RO	0x0	reserved
2:0	RW	0x0	clk_otp_div_con clk_otp=xin_osc0/(div_con+1)

CRU_CLKSEL_CON57
Address: Operational Base + offset (0x01e4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x1f	test_div_con clk_test_out=test_clk_src/(div_con+1)
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	testclk_sel 5'd00: clk_core 5'd01: aclk_gpu 5'd02: clk_ddrphy4x 5'd03: clk_i2c0 5'd04: aclk_vo 5'd05: clk_rga_core 5'd06: dclk_vopb 5'd07: dclk_vopl 5'd08: aclk_vpu 5'd09: aclk_vi 5'd10: clk_isp 5'd11: clk_rtc 5'd12: clk_ddrphy1x 5'd13: aclk_peri 5'd14: clk_nandc 5'd15: clk_sdmmc 5'd16: clk_sdio 5'd17: clk_emmc 5'd18: clk_pwm 5'd19: otp_ips_osc_out 5'd20: aclk_crypto 5'd21: clk_crypto_apk 5'd22: clk_24m 5'd23: aclk_gmac 5'd24: clk_gmac 5'd25: aclk_bus 5'd26: clk_pdm 5'd27: clk_i2s0 5'd28: clk_tsadc 5'd29: clk_uart1 5'd30: clk_saradc 5'd31: clk_otp

CRU_CLKSEL_CON58
Address: Operational Base + offset (0x01e8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_i2s0_rx_out_mclk_sel 2'b0:select selected clock by clk_i2s0_rx_tx_clk_sel 2'b1:select xin_osc0_half 2'b2:select clk_i2s0_tx
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	clk_i2s0_rx_tx_clk_sel 1'b0: select clk_i2s0_rx_clk 1'b1: select clk_i2s0_tx_clk
11:10	RW	0x0	clk_i2s0_rx_sel 2'b0:select clk_i2s0_rx 2'b1:select clk_i2s0_rx_frac_out 2'b2:select mclk_i2s0_rx_in 2'b3:select xin_osc0_half
9	RO	0x0	reserved
8	RW	0x0	clk_i2s0_rx_pll_sel 1'b0:GPLL 1'b1:NPLL
7	RO	0x0	reserved
6:0	RW	0x0b	clk_i2s0_rx_div_con clk_i2s0_rx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON59
Address: Operational Base + offset (0x01ec)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_i2s0_rx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s0_rx

CRU_CLKGATE_CON0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_ddrmon24m_clk_en When HIGH, disable clock
14	RW	0x0	clk_ddrphy4x_clk_en When HIGH, disable clock
13	RW	0x0	ddrphy_gpll_clk_en When HIGH, disable clock
12	RW	0x0	gpu_clk_div_clk_en When HIGH, disable clock
11	RW	0x0	aclk_gpu_niu_clk_en When HIGH, disable clock
10	RW	0x0	clk_gpu_clk_en When HIGH, disable clock
9	RW	0x0	gpu_clk_np5_src_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	gpu_pll_clk_en When HIGH, disable clock
7	RW	0x0	ddrphy_dpll_clk_en When HIGH, disable clock
6	RW	0x0	pclk_core_dbg_daplite_clk_en When HIGH, disable clock
5	RW	0x0	pclk_core_dbg_niu_clk_en When HIGH, disable clock
4	RW	0x0	aclk_core_niu_clk_en When HIGH, disable clock
3	RW	0x0	clk_jtag_core_clk_en When HIGH, disable clock
2	RW	0x0	pclk_core_dbg_src_clk_en When HIGH, disable clock
1	RW	0x0	aclk_core_src_clk_en When HIGH, disable clock
0	RW	0x0	core_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON1
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_axi_split_clk_en When HIGH, disable clock
14	RW	0x0	pclk_ddr_grf_clk_en When HIGH, disable clock
13	RW	0x0	clk_ddrstanby_clk_en When HIGH, disable clock
12	RW	0x0	pclk_ddrstdby_clk_en When HIGH, disable clock
11	RW	0x0	clk_ddrmon_clk_en When HIGH, disable clock
10	RW	0x0	pclk_ddrmon_clk_en When HIGH, disable clock
9	RW	0x0	pclk_msch_clk_en When HIGH, disable clock
8	RW	0x0	clk_msch_clk_en When HIGH, disable clock
7	RW	0x0	pclk_upctl2_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_ddrc_upctl2_clk_en When HIGH, disable clock
5	RW	0x0	aclk_upctl2_clk_en When HIGH, disable clock
4	RO	0x0	reserved
3	RW	0x0	pclk_axi_cmd_buffer_clk_en When HIGH, disable clock
2	RW	0x0	aclk_axi_cmd_buffer_clk_en When HIGH, disable clock
1	RW	0x0	ddr_pclk_pll_clk_en When HIGH, disable clock
0	RW	0x0	clk_stdby_src_clk_en When HIGH, disable clock

CRU_CLKGATE_CON2
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	pclk_vo_src_clk_en When HIGH, disable clock
12	RW	0x0	hclk_vo_src_clk_en When HIGH, disable clock
11:9	RO	0x0	reserved
8	RW	0x0	dclk_vopl_clk_en When HIGH, disable clock
7	RW	0x0	dclk_vopl_frac_src_clk_en When HIGH, disable clock
6	RW	0x0	dclk_vopl_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_pwm_vopb_pll_clk_en When HIGH, disable clock
4	RW	0x0	dclk_vopb_clk_en When HIGH, disable clock
3	RW	0x0	dclk_vopb_frac_src_clk_en When HIGH, disable clock
2	RW	0x0	dclk_vopb_pll_clk_en When HIGH, disable clock
1	RW	0x0	clk_rga_core_pll_clk_en When HIGH, disable clock
0	RW	0x0	aclk_vo_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON3

Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	pclk_mipi_dsi_host_clk_en When HIGH, disable clock
8	RW	0x0	hclk_rga_clk_en When HIGH, disable clock
7	RW	0x0	aclk_rga_clk_en When HIGH, disable clock
6	RW	0x0	hclk_vopl_clk_en When HIGH, disable clock
5	RW	0x0	aclk_vopl_clk_en When HIGH, disable clock
4	RW	0x0	hclk_vopb_clk_en When HIGH, disable clock
3	RW	0x0	aclk_vopb_clk_en When HIGH, disable clock
2	RW	0x0	pclk_vo_niu_clk_en When HIGH, disable clock
1	RW	0x0	hclk_vo_niu_clk_en When HIGH, disable clock
0	RW	0x0	aclk_vo_niu_clk_en When HIGH, disable clock

CRU_CLKGATE_CON4

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_vi_niu_clk_en When HIGH, disable clock
14	RW	0x0	pclkin_cif_clk_en When HIGH, disable clock
13	RW	0x0	pclkin_isp_clk_en When HIGH, disable clock
12	RW	0x0	hclk_vi_src_clk_en When HIGH, disable clock
11	RW	0x0	clk_cif_out_pll_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
10	RO	0x0	reserved
9	RW	0x0	clk_isp_pll_clk_en When HIGH, disable clock
8	RW	0x0	aclk_vi_pll_clk_en When HIGH, disable clock
7	RW	0x0	hclk_vpu_niu_clk_en When HIGH, disable clock
6	RW	0x0	hclk_vpu_clk_en When HIGH, disable clock
5	RW	0x0	aclk_vpu_niu_clk_en When HIGH, disable clock
4	RW	0x0	aclk_vpu_clk_en When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	hclk_vpu_src_clk_en When HIGH, disable clock
1	RW	0x0	clk_vpu_core_pll_clk_en When HIGH, disable clock
0	RW	0x0	aclk_vpu_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON5
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_nandc_clk_en When HIGH, disable clock
14	RO	0x0	reserved
13	RW	0x0	clk_nandc_clk_en When HIGH, disable clock
12	RW	0x0	clk_nandc_div50_clk_en When HIGH, disable clock
11	RW	0x0	clk_nandc_pll_clk_en When HIGH, disable clock
10	RO	0x0	reserved
9	RW	0x0	aclk_peri_niu_clk_en When HIGH, disable clock
8	RW	0x0	aclk_peri_clk_en When HIGH, disable clock
7	RW	0x0	aclk_hclk_pclk_peri_pll_clk_en When HIGH, disable clock
6:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	hclk_isp_clk_en When HIGH, disable clock
3	RW	0x0	aclk_isp_clk_en When HIGH, disable clock
2	RW	0x0	hclk_cif_clk_en When HIGH, disable clock
1	RW	0x0	aclk_cif_clk_en When HIGH, disable clock
0	RW	0x0	hclk_vi_niu_clk_en When HIGH, disable clock

CRU_CLKGATE_CON6
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sdmmc_clk_en When HIGH, disable clock
14	RW	0x0	clk_sdmmc_div50_clk_en When HIGH, disable clock
13	RW	0x0	clk_sdmmc_pll_clk_en When HIGH, disable clock
12	RW	0x0	hclk_pdsdcard_clk_en When HIGH, disable clock
11	RW	0x0	hclk_sfc_clk_en When HIGH, disable clock
10	RW	0x0	hclk_emmc_clk_en When HIGH, disable clock
9	RW	0x0	hclk_sdio_clk_en When HIGH, disable clock
8	RW	0x0	hclk_pdmmc_nand_niu_clk_en When HIGH, disable clock
7	RW	0x0	clk_sfc_pll_clk_en When HIGH, disable clock
6	RW	0x0	clk_emmc_clk_en When HIGH, disable clock
5	RW	0x0	clk_emmc_div50_clk_en When HIGH, disable clock
4	RW	0x0	clk_emmc_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_sdio_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	clk_sdio_div50_clk_en When HIGH, disable clock
1	RW	0x0	clk_sdio_pll_clk_en When HIGH, disable clock
0	RW	0x0	hclk_pdmmc_nand_clk_en When HIGH, disable clock

CRU_CLKGATE_CON7
Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_gmac_ref_clk_en When HIGH, disable clock
14	RO	0x0	reserved
13	RW	0x0	clk_gmac_tx_rx_clk_en When HIGH, disable clock
12	RW	0x0	pclk_gmac_src_clk_en When HIGH, disable clock
11	RW	0x0	clk_gmac_pll_clk_en When HIGH, disable clock
10	RW	0x0	aclk_pdgmac_clk_en When HIGH, disable clock
9	RO	0x0	reserved
8	RW	0x0	hclk_usb2host_arb_clk_en When HIGH, disable clock
7	RO	0x0	reserved
6	RW	0x0	hclk_usb2host_clk_en When HIGH, disable clock
5	RW	0x0	hclk_usb2otg_clk_en When HIGH, disable clock
4	RW	0x0	hclk_pdusb_niu_clk_en When HIGH, disable clock
3	RW	0x0	clk_otg_adp_clk_en When HIGH, disable clock
2	RW	0x0	hclk_pdusb_clk_en When HIGH, disable clock
1	RW	0x0	hclk_sdmmc_clk_en When HIGH, disable clock
0	RW	0x0	hclk_pdsdcard_niu_clk_en When HIGH, disable clock

CRU_CLKGATE_CON8
Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_crypto_apk_pll_clk_en When HIGH, disable clock
14	RW	0x0	clk_crypto_pll_clk_en When HIGH, disable clock
13	RW	0x0	hclk_pdcrypto_clk_en When HIGH, disable clock
12	RW	0x0	aclk_pdcrypto_clk_en When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	pclk_top_clk_en When HIGH, disable clock
9	RW	0x0	pclk_bus_clk_en When HIGH, disable clock
8	RW	0x0	hclk_bus_clk_en When HIGH, disable clock
7	RW	0x0	aclk_bus_clk_en When HIGH, disable clock
6	RW	0x0	pd_bus_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_gmac_out_pll_clk_en When HIGH, disable clock
4	RO	0x0	reserved
3	RW	0x0	pclk_gmac_clk_en When HIGH, disable clock
2	RW	0x0	aclk_gmac_clk_en When HIGH, disable clock
1	RW	0x0	pclk_gmac_niu_clk_en When HIGH, disable clock
0	RW	0x0	aclk_gmac_niu_clk_en When HIGH, disable clock

CRU_CLKGATE_CON9
Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s0_tx_out_mclk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
14	RW	0x0	clk_i2s0_tx_clk_en When HIGH, disable clock
13	RW	0x0	clk_i2s0_tx_frac_src_clk_en When HIGH, disable clock
12	RW	0x0	clk_i2s0_tx_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_pdm_clk_en When HIGH, disable clock
10	RW	0x0	clk_pdm_frac_src_clk_en When HIGH, disable clock
9	RW	0x0	clk_pdm_pll_clk_en When HIGH, disable clock
8:6	RO	0x0	reserved
5	RW	0x0	hclk_crypto_clk_en When HIGH, disable clock
4	RW	0x0	aclk_crypto_clk_en When HIGH, disable clock
3	RW	0x0	hclk_crypto_niu_clk_en When HIGH, disable clock
2	RW	0x0	aclk_crypto_niu_clk_en When HIGH, disable clock
1:0	RO	0x0	reserved

CRU_CLKGATE_CON10
Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart1_clk_en When HIGH, disable clock
14	RW	0x0	clk_uart1_frac_src_clk_en When HIGH, disable clock
13	RW	0x0	clk_uart1_divnp5_clk_en When HIGH, disable clock
12	RW	0x0	clk_uart1_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_i2s0_rx_out_mclk_oe 0:disable clk_i2s0_rx_out_mclk pad 1:enable clk_i2s0_rx_out_mclk pad
10	RW	0x0	clk_i2s2_out_mclk_oe 0:disable clk_i2s2_out_mclk pad 1:enable clk_i2s2_out_mclk pad

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_i2s1_out_mclk_oe 0:disable clk_i2s1_out_mclk pad 1:enable clk_i2s1_out_mclk pad
8	RW	0x0	clk_i2s0_tx_out_mclk_oe 0:disable clk_i2s0_tx_out_mclk pad 1:enable clk_i2s0_tx_out_mclk pad
7	RW	0x0	clk_i2s2_out_mclk_en When HIGH, disable clock
6	RW	0x0	clk_i2s2_clk_en When HIGH, disable clock
5	RW	0x0	clk_i2s2_frac_src_clk_en When HIGH, disable clock
4	RW	0x0	clk_i2s2_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_i2s1_out_mclk_en When HIGH, disable clock
2	RW	0x0	clk_i2s1_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2s1_frac_src_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2s1_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON11
Address: Operational Base + offset (0x022c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart5_clk_en When HIGH, disable clock
14	RW	0x0	clk_uart5_frac_src_clk_en When HIGH, disable clock
13	RW	0x0	clk_uart5_divnp5_clk_en When HIGH, disable clock
12	RW	0x0	clk_uart5_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_uart4_clk_en When HIGH, disable clock
10	RW	0x0	clk_uart4_frac_src_clk_en When HIGH, disable clock
9	RW	0x0	clk_uart4_divnp5_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	clk_uart4_pll_clk_en When HIGH, disable clock
7	RW	0x0	clk_uart3_clk_en When HIGH, disable clock
6	RW	0x0	clk_uart3_frac_src_clk_en When HIGH, disable clock
5	RW	0x0	clk_uart3_divnp5_clk_en When HIGH, disable clock
4	RW	0x0	clk_uart3_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_uart2_clk_en When HIGH, disable clock
2	RW	0x0	clk_uart2_frac_src_clk_en When HIGH, disable clock
1	RW	0x0	clk_uart2_divnp5_clk_en When HIGH, disable clock
0	RW	0x0	clk_uart2_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON12
Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_cpu_boost_clk_en When HIGH, disable clock
11	RW	0x0	clk_otp_pll_clk_en When HIGH, disable clock
10	RW	0x0	clk_saradc_pll_clk_en When HIGH, disable clock
9	RW	0x0	clk_tsadc_pll_clk_en When HIGH, disable clock
8	RW	0x0	clk_spi1_pll_clk_en When HIGH, disable clock
7	RW	0x0	clk_spi0_pll_clk_en When HIGH, disable clock
6	RW	0x0	clk_pwm1_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_pwm0_pll_clk_en When HIGH, disable clock
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	clk_i2c3_pll_clk_en When HIGH, disable clock
2	RW	0x0	clk_i2c2_pll_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2c1_pll_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2c0_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON13
Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_dfc_clk_en When HIGH, disable clock
14	RW	0x0	hclk_rom_clk_en When HIGH, disable clock
13	RO	0x0	reserved
12	RW	0x0	aclk_gic_clk_en When HIGH, disable clock
11	RW	0x0	aclk_intmem_clk_en When HIGH, disable clock
10	RW	0x0	pclk_bus_niu_clk_en When HIGH, disable clock
9	RW	0x0	hclk_bus_niu_clk_en When HIGH, disable clock
8	RW	0x0	aclk_bus_niu_clk_en When HIGH, disable clock
7	RO	0x0	reserved
6	RW	0x0	clk_otp_usr_clk_en When HIGH, disable clock
5	RW	0x0	clk_timer5_clk_en When HIGH, disable clock
4	RW	0x0	clk_timer4_clk_en When HIGH, disable clock
3	RW	0x0	clk_timer3_clk_en When HIGH, disable clock
2	RW	0x0	clk_timer2_clk_en When HIGH, disable clock
1	RW	0x0	clk_timer1_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	clk_timer0_clk_en When HIGH, disable clock

CRU_CLKGATE_CON14
Address: Operational Base + offset (0x0238)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_pwm0_clk_en When HIGH, disable clock
14	RO	0x0	reserved
13	RW	0x0	pclk_i2c3_clk_en When HIGH, disable clock
12	RW	0x0	pclk_i2c2_clk_en When HIGH, disable clock
11	RW	0x0	pclk_i2c1_clk_en When HIGH, disable clock
10	RW	0x0	pclk_i2c0_clk_en When HIGH, disable clock
9	RW	0x0	pclk_uart5_clk_en When HIGH, disable clock
8	RW	0x0	pclk_uart4_clk_en When HIGH, disable clock
7	RW	0x0	pclk_uart3_clk_en When HIGH, disable clock
6	RW	0x0	pclk_uart2_clk_en When HIGH, disable clock
5	RW	0x0	pclk_uart1_clk_en When HIGH, disable clock
4	RW	0x0	hclk_i2s2_clk_en When HIGH, disable clock
3	RW	0x0	hclk_i2s1_clk_en When HIGH, disable clock
2	RW	0x0	hclk_i2s0_clk_en When HIGH, disable clock
1	RW	0x0	hclk_pdm_clk_en When HIGH, disable clock
0	RW	0x0	pclk_dcf_clk_en When HIGH, disable clock

CRU_CLKGATE_CON15
Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	pclk_bus_sgrf_clk_en When HIGH, disable clock
11	RW	0x0	pclk_bus_grf_clk_en When HIGH, disable clock
10	RW	0x0	pclk_gpio3_clk_en When HIGH, disable clock
9	RW	0x0	pclk_gpio2_clk_en When HIGH, disable clock
8	RW	0x0	pclk_gpio1_clk_en When HIGH, disable clock
7	RW	0x0	pclk_wdt_ns_en When HIGH, disable clock
6	RW	0x0	pclk_otp_ns_clk_en When HIGH, disable clock
5	RW	0x0	pclk_timer_clk_en When HIGH, disable clock
4	RW	0x0	pclk_tsadc_clk_en When HIGH, disable clock
3	RW	0x0	pclk_saradc_clk_en When HIGH, disable clock
2	RW	0x0	pclk_spi1_clk_en When HIGH, disable clock
1	RW	0x0	pclk_spi0_clk_en When HIGH, disable clock
0	RW	0x0	pclk_pwm1_clk_en When HIGH, disable clock

CRU_CLKGATE_CON16
Address: Operational Base + offset (0x0240)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	testclk_clk_en When HIGH, disable clock
14:8	RO	0x0	reserved
7	RW	0x0	pclk_cpu_boost_clk_en When HIGH, disable clock
6	RW	0x0	pclk_usb_grf_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
5	RW	0x0	pclk_mipicsiphy_clk_en When HIGH, disable clock
4	RW	0x0	pclk_mipidsiphy_clk_en When HIGH, disable clock
3	RW	0x0	pclk_ddrphy_clk_en When HIGH, disable clock
2	RW	0x0	pclk_otp_phy_clk_en When HIGH, disable clock
1	RW	0x0	pclk_top_cru_clk_en When HIGH, disable clock
0	RW	0x0	pclk_top_niu_clk_en When HIGH, disable clock

CRU_CLKGATE_CON17
Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	aclk_gpu_div_clk_en When HIGH, disable clock
9	RW	0x0	pclk_gpu_grf_clk_en When HIGH, disable clock
8	RW	0x0	aclk_gpu_perf_clk_en When HIGH, disable clock
7	RO	0x0	reserved
6	RW	0x0	pclk_core_grf_clk_en When HIGH, disable clock
5	RW	0x0	aclk_core_perf_clk_en When HIGH, disable clock
4	RW	0x0	clk_core_pvtm_clk_en When HIGH, disable clock
3	RW	0x0	clk_i2s0_rx_out_mclk_en When HIGH, disable clock
2	RW	0x0	clk_i2s0_rx_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2s0_rx_divfrac_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2s0_rx_pll_clk_en When HIGH, disable clock

CRU_SSGTBL0_3
Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl0_3 Extern wave table 0-3 7-0: table0 15-8: table1 23-16: table2 31-24: table3

CRU_SSGTBL4_7
Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl4_7 Extern wave table 4-7 7-0: table4 15-8: table5 23-16: table6 31-24: table7

CRU_SSGTBL8_11
Address: Operational Base + offset (0x0288)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl8_11 Extern wave table 8-11 7-0: table8 15-8: table9 23-16: table10 31-24: table11

CRU_SSGTBL12_15
Address: Operational Base + offset (0x028c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl12_15 Extern wave table 12-15 7-0: table12 15-8: table13 23-16: table14 31-24: table15

CRU_SSGTBL16_19
Address: Operational Base + offset (0x0290)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl16_19 Extern wave table 16-19 7-0: table16 15-8: table17 23-16: table18 31-24: table19

CRU_SSGTBL20_23
Address: Operational Base + offset (0x0294)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl20_23 Extern wave table 20-23 7-0: table20 15-8: table21 23-16: table22 31-24: table23

CRU_SSGTBL24_27
Address: Operational Base + offset (0x0298)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl24_27 Extern wave table 24-27 7-0: table24 15-8: table25 23-16: table26 31-24: table27

CRU_SSGTBL28_31
Address: Operational Base + offset (0x029c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl28_31 Extern wave table 28-31 7-0: table28 15-8: table29 23-16: table30 31-24: table31

CRU_SSGTBL32_35
Address: Operational Base + offset (0x02a0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl32_35 Extern wave table 32-35 7-0: table32 15-8: table33 23-16: table34 31-24: table35

CRU_SSGTBL36_39
Address: Operational Base + offset (0x02a4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl36_39 Extern wave table 36-39 7-0: table36 15-8: table37 23-16: table38 31-24: table39

CRU_SSGTBL40_43
Address: Operational Base + offset (0x02a8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl40_43 Extern wave table 40-43 7-0: table40 15-8: table41 23-16: table42 31-24: table43

CRU_SSGTBL44_47
Address: Operational Base + offset (0x02ac)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl44_47 Extern wave table 44-47 7-0: table44 15-8: table45 23-16: table46 31-24: table47

CRU_SSGTBL48_51
Address: Operational Base + offset (0x02b0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl48_51 Extern wave table 48-51 7-0: table48 15-8: table49 23-16: table50 31-24: table51

CRU_SSGTBL52_55

Address: Operational Base + offset (0x02b4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl52_55 Extern wave table 52-55 7-0: table52 15-8: table53 23-16: table54 31-24: table55

CRU_SSGTBL56_59
Address: Operational Base + offset (0x02b8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl56_59 Extern wave table 56-59 7-0: table56 15-8: table57 23-16: table58 31-24: table59

CRU_SSGTBL60_63
Address: Operational Base + offset (0x02bc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl60_63 Extern wave table 60-63 7-0: table60 15-8: table61 23-16: table62 31-24: table63

CRU_SSGTBL64_67
Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl64_67 Extern wave table 64-67 7-0: table64 15-8: table65 23-16: table66 31-24: table67

CRU_SSGTBL68_71
Address: Operational Base + offset (0x02c4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl68_71 Extern wave table 68-71 7-0: table68 15-8: table69 23-16: table70 31-24: table71

CRU_SSGTBL72_75
Address: Operational Base + offset (0x02c8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl72_75 Extern wave table 72-75 7-0: table72 15-8: table73 23-16: table74 31-24: table75

CRU_SSGTBL76_79
Address: Operational Base + offset (0x02cc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl76_79 Extern wave table 76-79 7-0: table76 15-8: table77 23-16: table78 31-24: table79

CRU_SSGTBL80_83
Address: Operational Base + offset (0x02d0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl80_83 Extern wave table 76-79 7-0: table80 15-8: table81 23-16: table82 31-24: table83

CRU_SSGTBL84_87
Address: Operational Base + offset (0x02d4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl84_87 Extern wave table 84-87 7-0: table84 15-8: table85 23-16: table86 31-24: table87

CRU_SSGTBL88_91
Address: Operational Base + offset (0x02d8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl88_91 Extern wave table 88-91 7-0: table88 15-8: table89 23-16: table90 31-24: table91

CRU_SSGTBL92_95
Address: Operational Base + offset (0x02dc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl92_95 Extern wave table 92-95 7-0: table92 15-8: table93 23-16: table94 31-24: table95

CRU_SSGTBL96_99
Address: Operational Base + offset (0x02e0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl96_99 Extern wave table 96-99 7-0: table96 15-8: table97 23-16: table98 31-24: table99

CRU_SSGTBL100_103
Address: Operational Base + offset (0x02e4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl100_103 Extern wave table 100-103 7-0: table100 15-8: table101 23-16: table102 31-24: table103

CRU_SSGTBL104_107
Address: Operational Base + offset (0x02e8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl104_107 Extern wave table 104-107 7-0: table104 15-8: table105 23-16: table106 31-24: table107

CRU_SSGTBL108_111
Address: Operational Base + offset (0x02ec)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl108_111 Extern wave table 108-111 7-0: table108 15-8: table109 23-16: table110 31-24: table111

CRU_SSGTBL112_115
Address: Operational Base + offset (0x02f0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl112_115 Extern wave table 112-115 7-0: table112 15-8: table113 23-16: table114 31-24: table115

CRU_SSGTBL116_119
Address: Operational Base + offset (0x02f4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl116_119 Extern wave table 116-119 7-0: table116 15-8: table117 23-16: table118 31-24: table119

CRU_SSGTBL120_123
Address: Operational Base + offset (0x02f8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl120_123 Extern wave table 120-123 7-0: table120 15-8: table121 23-16: table122 31-24: table123

CRU_SSGTBL124_127
Address: Operational Base + offset (0x02fc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl124_127 Extern wave table 124-127 7-0: table124 15-8: table125 23-16: table126 31-24: table127

CRU_SOFTRST_CON0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	R/W SC	0x0	l2_srstn_req When HIGH, reset relative logic
14	R/W SC	0x0	strc_sys_asrstn_req When HIGH, reset relative logic
13	R/W SC	0x0	core_noc_srstn_req When HIGH, reset relative logic
12	RW	0x0	topdbg_srstn_req When HIGH, reset relative logic
11	RW	0x0	core3_dbg_srstn_req When HIGH, reset relative logic
10	RW	0x0	core2_dbg_srstn_req When HIGH, reset relative logic
9	RW	0x0	core1_dbg_srstn_req When HIGH, reset relative logic
8	RW	0x0	core0_dbg_srstn_req When HIGH, reset relative logic
7	RW	0x0	core3_srstn_req When HIGH, reset relative logic
6	RW	0x0	core2_srstn_req When HIGH, reset relative logic
5	RW	0x0	core1_srstn_req When HIGH, reset relative logic
4	R/W SC	0x0	core0_srstn_req When HIGH, reset relative logic
3	RW	0x0	corepo3_srstn_req When HIGH, reset relative logic
2	RW	0x0	corepo2_srstn_req When HIGH, reset relative logic
1	RW	0x0	corepo1_srstn_req When HIGH, reset relative logic
0	R/W SC	0x0	corepo0_srstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x0	axi_cmd_buffert_psrstn_req When HIGH, reset relative logic
14	RW	0x0	axi_cmd_buffer_asrstn_req When HIGH, reset relative logic
13	RW	0x0	axi_split_asrstn_req When HIGH, reset relative logic
12	RW	0x0	ddrgrf_psrstn_req When HIGH, reset relative logic
11	RW	0x0	ddrstdby_srstn_req When HIGH, reset relative logic
10	RW	0x0	ddrstdby_psrstn_req When HIGH, reset relative logic
9	RW	0x0	ddrmon_psrstn_req When HIGH, reset relative logic
8	RW	0x0	msch_psrstn_req When HIGH, reset relative logic
7	RW	0x0	msch_srstn_req When HIGH, reset relative logic
6	RW	0x0	upctl2_prstn_req When HIGH, reset relative logic
5	RW	0x0	upctl2_asrstn_req When HIGH, reset relative logic
4	RW	0x0	upctl2_srstn_req When HIGH, reset relative logic
3	RW	0x0	gpu_niu_srstn_req When HIGH, reset relative logic
2	RW	0x0	gpu_srstn_req When HIGH, reset relative logic
1	RW	0x0	core_pvtm_srstn_req When HIGH, reset relative logic
0	RW	0x0	dap_srstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON2
Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	mipicsiphy_psrstn_req When HIGH, reset relative logic
14	RW	0x0	cif_pclkin_srstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
13	RW	0x0	cif_hsrstn_req When HIGH, reset relative logic
12	RW	0x0	cif_asrstn_req When HIGH, reset relative logic
11	RW	0x0	isp_srstn_req When HIGH, reset relative logic
10	RW	0x0	isp_hsrstn_req When HIGH, reset relative logic
9	RW	0x0	vi_niu_hsrstn_req When HIGH, reset relative logic
8	RW	0x0	vi_niu_asrstn_req When HIGH, reset relative logic
7	RW	0x0	vpu_niu_hsrstn_req When HIGH, reset relative logic
6	RW	0x0	vpu_hsrstn_req When HIGH, reset relative logic
5	RW	0x0	vpu_niu_asrstn_req When HIGH, reset relative logic
4	RW	0x0	vpu_asrstn_req When HIGH, reset relative logic
3	RO	0x0	reserved
2	RW	0x0	ddrphy_psrstn_req When HIGH, reset relative logic
1	RW	0x0	ddrphydiv_srstn_req When HIGH, reset relative logic
0	RW	0x0	ddrphy_srstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON3
Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	vpu_core_srstn_req When HIGH, reset relative logic
14	RW	0x0	mipidsiphy_psrstn_req When HIGH, reset relative logic
13	RW	0x0	mipidsi_host_psrstn_req When HIGH, reset relative logic
12	RW	0x0	rga_srstn_req When HIGH, reset relative logic
11	RW	0x0	rga_hsrstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	rga_asrstn_req When HIGH, reset relative logic
9	RW	0x0	vopl_srstn_req When HIGH, reset relative logic
8	RW	0x0	vopl_hsrstn_req When HIGH, reset relative logic
7	RW	0x0	vopl_asrstn_req When HIGH, reset relative logic
6	RW	0x0	pwm_vopb_srstn_req When HIGH, reset relative logic
5	RW	0x0	vopb_srstn_req When HIGH, reset relative logic
4	RW	0x0	vopb_hsrstn_req When HIGH, reset relative logic
3	RW	0x0	vopb_asrstn_req When HIGH, reset relative logic
2	RW	0x0	vo_niu_psrstn_req When HIGH, reset relative logic
1	RW	0x0	vo_niu_hsrstn_req When HIGH, reset relative logic
0	RW	0x0	vo_niu_asrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON4
Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	cpu_boost_srstn_req When HIGH, reset relative logic
14	RW	0x0	cpu_boost_psrstn_req When HIGH, reset relative logic
13	RW	0x0	usbphy_grf_psrstn_req When HIGH, reset relative logic
12	RW	0x0	usbphy_host_port_srstn_req When HIGH, reset relative logic
11	RW	0x0	usbphy_otg_port_srstn_req When HIGH, reset relative logic
10	RW	0x0	usbphypor_srstn_req When HIGH, reset relative logic
9	RW	0x0	usb2host_srstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
8	RW	0x0	usb2host_ehci_srstn_req When HIGH, reset relative logic
7	RW	0x0	usb2host_aux_hsrstn_req When HIGH, reset relative logic
6	RW	0x0	usb2host_arb_hsrstn_req When HIGH, reset relative logic
5	RW	0x0	usb2host_hsrstn_req When HIGH, reset relative logic
4	RW	0x0	usb2otg_adp_srstn_req When HIGH, reset relative logic
3	RW	0x0	usb2otg_srstn_req When HIGH, reset relative logic
2	RW	0x0	usb2otg_hsrstn_req When HIGH, reset relative logic
1	RW	0x0	usb_niu_hsrstn_req When HIGH, reset relative logic
0	RW	0x0	peri_niu_asrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON5
Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	gmac_asrstn_req When HIGH, reset relative logic
13	RW	0x0	gmac_niu_psrstn_req When HIGH, reset relative logic
12	RW	0x0	gmac_niu_asrstn_req When HIGH, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	nandc_srstn_req When HIGH, reset relative logic
9	RW	0x0	nandc_hrstn_req When HIGH, reset relative logic
8:7	RO	0x0	reserved
6	RW	0x0	sdmmc_hsrstn_req When HIGH, reset relative logic
5	RW	0x0	pdsdcard_niu_hsrstn_req When HIGH, reset relative logic
4	RW	0x0	sfc_srstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	sfc_hsrstn_req When HIGH, reset relative logic
2	RW	0x0	emmc_hsrstn_req When HIGH, reset relative logic
1	RW	0x0	sdio_hsrstn_req When HIGH, reset relative logic
0	RW	0x0	pdmmc_nand_niu_hsrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON6
Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	gpu_grf_psrstn_req When HIGH, reset relative logic
14	RW	0x0	gpu_perf_asrstn_req When HIGH, reset relative logic
13	RW	0x0	core_grf_psrstn_req When HIGH, reset relative logic
12	RW	0x0	core_perf_asrstn_req When HIGH, reset relative logic
11	RW	0x0	pmu_ddr_fail_save_srstn_req When HIGH, reset relative logic
10	RW	0x0	pmu_niu_hrstn_req When HIGH, reset relative logic
9	RW	0x0	pmu_uart_srstn_req When HIGH, reset relative logic
8	RW	0x0	pmu_pvtm_srstn_req When HIGH, reset relative logic
7	RW	0x0	pmu_cru_psrstn_req When HIGH, reset relative logic
6	RW	0x0	pmu_uart0_psrstn_req When HIGH, reset relative logic
5	RW	0x0	pmu_gpio0_psrstn_req When HIGH, reset relative logic
4	RW	0x0	pmu_mem_psrstn_req When HIGH, reset relative logic
3	RW	0x0	pmu_pmu_srstn_req When HIGH, reset relative logic
2	RW	0x0	pmu_grf_psrstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
1	RW	0x0	pmu_sgrf_psrstn_req When HIGH, reset relative logic
0	RW	0x0	pmu_niu_psrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON7
Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	dcf_asrstn_req When HIGH, reset relative logic
14	RW	0x0	rom_hsrstn_req When HIGH, reset relative logic
13	RO	0x0	reserved
12	RW	0x0	gic_asrst_req When HIGH, reset relative logic
11	RW	0x0	intmem_asrst_req When HIGH, reset relative logic
10	RW	0x0	bus_top_niu_psrst_req When HIGH, reset relative logic
9	RW	0x0	bus_niu_psrst_req When HIGH, reset relative logic
8	RW	0x0	bus_niu_hsrstn_req When HIGH, reset relative logic
7:6	RO	0x0	reserved
5	RW	0x0	crypto_apk_srstn_req When HIGH, reset relative logic
4	RW	0x0	crypto_srstn_req When HIGH, reset relative logic
3	RW	0x0	crypto_hsrstn_req When HIGH, reset relative logic
2	RW	0x0	crypto_asrstn_req When HIGH, reset relative logic
1	RW	0x0	crypto_niu_hsrstn_req When HIGH, reset relative logic
0	RW	0x0	crypto_niu_asrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON8
Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	uart4_psrstn_req When HIGH, reset relative logic
14	RW	0x0	uart3_srstn_req When HIGH, reset relative logic
13	RW	0x0	uart3_psrstn_req When HIGH, reset relative logic
12	RW	0x0	uart2_srstn_req When HIGH, reset relative logic
11	RW	0x0	uart2_psrstn_req When HIGH, reset relative logic
10	RW	0x0	uart1_srstn_req When HIGH, reset relative logic
9	RW	0x0	uart1_psrstn_req When HIGH, reset relative logic
8	RW	0x0	i2s2_srstn_req When HIGH, reset relative logic
7	RW	0x0	i2s2_hsrstn_req When HIGH, reset relative logic
6	RW	0x0	i2s1_srstn_req When HIGH, reset relative logic
5	RW	0x0	i2s1_hsrstn_req When HIGH, reset relative logic
4	RW	0x0	i2s0_tx_srstn_req When HIGH, reset relative logic
3	RW	0x0	i2s0_hsrstn_req When HIGH, reset relative logic
2	RW	0x0	pdm_srstn_req When HIGH, reset relative logic
1	RW	0x0	pdm_hsrstn_req When HIGH, reset relative logic
0	RW	0x0	dcf_psrstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON9
Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pwm1_psrstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
14	RW	0x0	pwm0_srstn_req When HIGH, reset relative logic
13	RW	0x0	pwm0_psrstn_req When HIGH, reset relative logic
12:11	RO	0x0	reserved
10	RW	0x0	i2c3_srstn_req When HIGH, reset relative logic
9	RW	0x0	i2c3_psrstn_req When HIGH, reset relative logic
8	RW	0x0	i2c2_srstn_req When HIGH, reset relative logic
7	RW	0x0	i2c2_psrstn_req When HIGH, reset relative logic
6	RW	0x0	i2c1_srstn_req When HIGH, reset relative logic
5	RW	0x0	i2c1_psrstn_req When HIGH, reset relative logic
4	RW	0x0	i2c0_srstn_req When HIGH, reset relative logic
3	RW	0x0	i2c0_psrstn_req When HIGH, reset relative logic
2	RW	0x0	uart5_srstn_req When HIGH, reset relative logic
1	RW	0x0	uart5_psrstn_req When HIGH, reset relative logic
0	RW	0x0	uart4_srstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON10
Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	timer5_srstn_req When HIGH, reset relative logic
14	RW	0x0	timer4_srstn_req When HIGH, reset relative logic
13	RW	0x0	timer3_srstn_req When HIGH, reset relative logic
12	RW	0x0	timer2_srstn_req When HIGH, reset relative logic
11	RW	0x0	timer1_srstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	timer0_srstn_req When HIGH, reset relative logic
9	RW	0x0	timer_psrstn_req When HIGH, reset relative logic
8	RW	0x0	tsadc_srstn_req When HIGH, reset relative logic
7	RW	0x0	tsadc_psrstn_req When HIGH, reset relative logic
6	RW	0x0	saradc_srstn_req When HIGH, reset relative logic
5	RW	0x0	saradc_psrstn_req When HIGH, reset relative logic
4	RW	0x0	spi1_srstn_req When HIGH, reset relative logic
3	RW	0x0	spi1_psrstn_req When HIGH, reset relative logic
2	RW	0x0	spi0_srstn_req When HIGH, reset relative logic
1	RW	0x0	spi0_psrstn_req When HIGH, reset relative logic
0	RW	0x0	pwm1_srstn_req When HIGH, reset relative logic

CRU_SOFTRST_CON11
Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2s0_rx_srstn_req When HIGH, reset relative logic
14:11	RO	0x0	reserved
10	RW	0x0	grf_psrstn_req When HIGH, reset relative logic
9	RW	0x0	sgrf_psrstn_req When HIGH, reset relative logic
8	RW	0x0	gpio3_psrstn_req When HIGH, reset relative logic
7	RW	0x0	gpio2_psrstn_req When HIGH, reset relative logic
6	RW	0x0	gpio1_psrstn_req When HIGH, reset relative logic
5	RW	0x0	wdt_ns_psrstn_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
4	RW	0x0	otp_phy_srstn_req When HIGH, reset relative logic
3	RW	0x0	otp_phy_psrstn_req When HIGH, reset relative logic
2	RW	0x0	otp_ns_usr_srstn_req When HIGH, reset relative logic
1	RW	0x0	otp_ns_sbpi_srstn_req When HIGH, reset relative logic
0	RW	0x0	otp_ns_psrstn_req When HIGH, reset relative logic

CRU_SDMMC_CON0
Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel drv_sel
10:3	RW	0x00	drv_delaynum drv_delaynum
2:1	RW	0x2	drv_degree drv_degree
0	RW	0x0	init_state init_state

CRU_SDMMC_CON1
Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	sample_sel sample_sel
10:3	RW	0x00	sample_delaynum sample_delaynum
2:1	RW	0x0	sample_degree sample_degree
0	RO	0x0	reserved

CRU_SDIO_CON0
Address: Operational Base + offset (0x0388)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel drv_sel
10:3	RW	0x00	drv_delaynum drv_delaynum
2:1	RW	0x2	drv_degree drv_degree
0	RW	0x0	init_state init_state

CRU_SDIO_CON1
Address: Operational Base + offset (0x038c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	sample_sel sample_sel
10:3	RW	0x00	sample_delaynum sample_delaynum
2:1	RW	0x0	sample_degree sample_degree
0	RO	0x0	reserved

CRU_EMMC_CON0
Address: Operational Base + offset (0x0390)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	drv_sel drv_sel
10:3	RW	0x00	drv_delaynum drv_delaynum
2:1	RW	0x2	drv_degree drv_degree
0	RW	0x0	init_state init_state

CRU_EMMC_CON1
Address: Operational Base + offset (0x0394)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	sample_sel sample_sel
10:3	RW	0x00	sample_delaynum sample_delaynum
2:1	RW	0x0	sample_degree sample_degree
0	RO	0x0	reserved

CRU_GPLL_CON0
Address: Operational Base + offset (0xc000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 First Post Divide Value, (1-7)
11:0	RW	0x032	fbdiv Feedback Divide Value, valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_GPLL_CON1
Address: Operational Base + offset (0xc004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15	RW	0x0	pllpdsel PLL global power down source selection If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value, (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value, (1-63)

CRU_GPLL_CON2
Address: Operational Base + offset (0xc008)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down 4-phase clocks and 2X, 3X, 4X clocks 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd Power down buffered VCO clock 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Power down all outputs except for buffered VCO clock 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
24	RW	0x0	dacpd Power down quantization noise cancellation DAC 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_GPLL_CON3
Address: Operational Base + offset (0xc00c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	WO	0x0	ssmod_divval Divider required to set the modulation frequency
3	WO	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_GPLL_CON4
Address: Operational Base + offset (0xc010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:8	WO	0x7f	ssmod_ext_maxaddr External wave table data inputs, (0-255)
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave 1'b0: no select ext_wave 1'b1: select ext_wave

CRU_PMU_MODE
Address: Operational Base + offset (0xc020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:2	RO	0x0	reserved
1:0	RW	0x0	gpll_work_mode 2'h0:clock from xin_osc0_func_div 2'h1:clock from pll 2'h2:clock from clk_rtc_32k

CRU_PMU_CLKSEL_CON0
Address: Operational Base + offset (0xc040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_rtc32k_clk_sel 2'h0:select clk_32k_from_io as clk_rtc_32k 2'h1:select clk_32k_from_pvtm as clk_rtc_32k 2'h2:select clk_div32p768khz as clk_rtc_32k
13	RO	0x0	reserved
12:8	RW	0x00	xin_osc0_func_div_con xin_osc0_func_div=xin_osc0/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x0b	pclk_pdpmu_div_con pclk_pdpmu=gpll_clk_src/(div_con+1)

CRU_PMU_CLKSEL_CON1
Address: Operational Base + offset (0xc044)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_div32p768khz_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is xin_osc0

CRU_PMU_CLKSEL_CON2
Address: Operational Base + offset (0xc048)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_wifi_sel 1'b0:select xin_osc0 as clk_wifi_out 1'b1:select clk_wifi_div as clk_wifi_out
14	RO	0x0	reserved
13:8	RW	0x31	clk_wifi_div_con clk_wifi_div=gpll_clk_src/(div_con+1)
7	RW	0x0	mipidsiphy_ref_sel 1'b0:select xin_osc0 as mipidsi phy reference clock 1'b1:select clk_ref24m as mipidsi phy reference clock
6	RW	0x0	usbphy_ref_sel 1'b0:select xin_osc0 as usbphy reference clock 1'b1:select clk_ref24m as usbphy reference clock
5:0	RW	0x31	clk_ref24m_div_con clk_ref24m=gpll_clk_src/(div_con+1)

CRU_PMU_CLKSEL_CON3
Address: Operational Base + offset (0xc04c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart0_pll_sel 2'h0:GPLL 2'h1:xin_osc0 2'h2:usbphy480M 2'h3:NPLL
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart0_div_con clk_uart0=pll_clk_src/(div_con+1)

CRU_PMU_CLKSEL_CON4
Address: Operational Base + offset (0xc050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	clk_uart0_sel 2'h0:select clk_uart0 2'h1:select clk_uart0_np5 2'h2:select clk_uart0_frac_out
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart0_divnp5_div_con clk_uart0_np5=2*clk_uart0/(2*div_con+3)

CRU_PMU_CLKSEL_CON5
Address: Operational Base + offset (0xc054)

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	clk_uart0_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart0

CRU_PMU_CLKGATE_CON0
Address: Operational Base + offset (0xc080)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_wifi_clk_en When HIGH, disable clock
14	RW	0x0	clk_wifi_pll_clk_en When HIGH, disable clock
13	RW	0x0	clk_div32p768khz_src_clk_en When HIGH, disable clock
12	RW	0x0	xin_osc0_func_div_src_clk_en When HIGH, disable clock
11:9	RO	0x0	reserved
8	RW	0x0	pclk_pmu_cru_clk_en When HIGH, disable clock
7	RW	0x0	pclk_pmu_uart0_clk_en When HIGH, disable clock
6	RW	0x0	pclk_pmu_gpio0_clk_en When HIGH, disable clock
5	RW	0x0	pclk_pmu_mem_clk_en When HIGH, disable clock
4	RW	0x0	pclk_pmu_pmu_clk_en When HIGH, disable clock
3	RW	0x0	pclk_pmu_grf_clk_en When HIGH, disable clock
2	RW	0x0	pclk_pmu_sgrf_clk_en When HIGH, disable clock
1	RW	0x0	pclk_pmu_niu_clk_en When HIGH, disable clock
0	RW	0x0	pclk_pdpmu_pll_clk_en When HIGH, disable clock

CRU_PMU_CLKGATE_CON1
Address: Operational Base + offset (0xc084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit; when every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	mipidsiphy_ref_cclk_en When HIGH, disable clock
9	RW	0x0	usbphy_ref_clk_en When HIGH, disable clock
8	RW	0x0	clk_ref24m_pll_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
7:5	RO	0x0	reserved
4	RW	0x0	clk_pvtm_pmu_clk_en When HIGH, disable clock
3	RW	0x0	clk_uart0_pmu_clk_en When HIGH, disable clock
2	RW	0x0	clk_uart0_pmu_frac_clk_en When HIGH, disable clock
1	RW	0x0	clk_uart0_pmu_divnp5_clk_en When HIGH, disable clock
0	RW	0x0	clk_uart0_pmu_pll_clk_en When HIGH, disable clock

2.7 Timing Diagram

Power on reset timing is shown as follow:

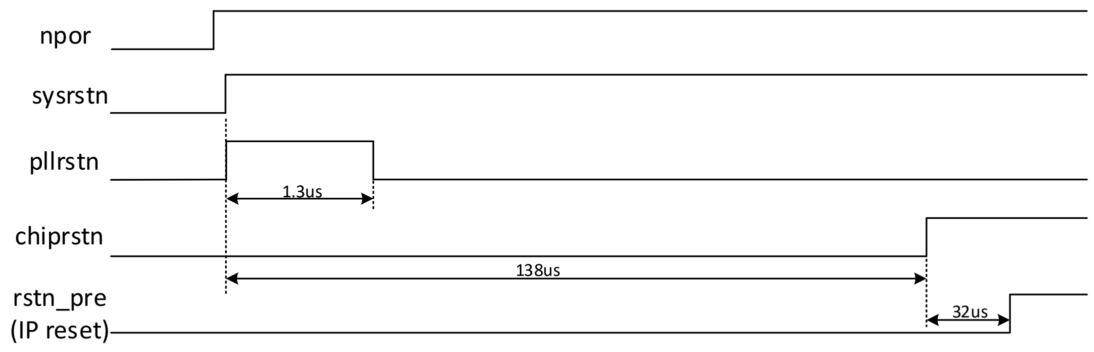


Fig. 2-4 Chip Power On Reset Timing Diagram

Npor is hardware reset signal from out-chip, which is filtered glitch to obtain signal sysrstn. To make PLLs work normally, the PLL reset signal (pllrstn) must maintain high for more than 1us, and PLLs start to lock when pllrstn de-assert, and the PLL max lock time is 1500 PLL REFCLK cycles. And then the system will wait about 138us, and then de-assert reset signal chiprstn. The signal chiprstn is used to generate output clocks in CRU. After CRU start output clocks, the system waits again for 768cycles (21.3us) to de-assert signal rstn_pre, which is used to generate power on reset of all IPs.

2.8 Application Notes

2.8.1 PLL usage

A. PLL output frequency configuration

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_xPLL_CON0.
DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_xPLL_CON1.
FRAC can be configured by programming CRU_xPLL_CON2.
If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = (FREF / REFDIV) * FBDIV
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 1

REFDIV = 6
FBDIV = 175
POSTDIV1=1
POSTDIV2=1
And then
FOUTVCO = (FREF / REFDIV) * FBDIV = 24/6*175=700
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)=700/1/1=700
If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = (FREF / REFDIV) * (FBDIV + FRAC / (2^24))
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:
DSMPD = 0
REFDIV = 1
FBDIV = 40
FRAC = 24’hf5c28f
POSTDIV1=2
POSTDIV2=1
And then
FOUTVCO = (FREF / REFDIV) * (FBDIV + FRAC / (2^24)) = 983.04
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)=983.04/(2*1)=491.52

B. PLL setting consideration

- If the POSTDIV value is changed during operation a short pulse (glitch) may occur on FOUTPOSTDIV. The minimum width of the short pulse will be equal to twice the period of the VCO. Therefore, if the circuitry clocked by the PLL is sensitive to short pulses, the new divide value should be re-timed so that it is synchronous with the rising edge of the output clock (FOUTPOSTDIV). Glitches cannot occur on any of the other outputs.
- For lowest power operation, the minimum VCO and FREF frequencies should be used. For minimum jitter operation, the highest VCO and FREF frequencies should be used. The normal operating range for the VCO is described above in.
- The supply rejection will be worse at the low end of the VCO range so care should be taken to keep the supply clean for low power applications.
- The feedback divider is not capable of dividing by all possible settings due to the use of a power-saving architecture. The following settings are valid for FBDIV:
- DSMPD=1 (Integer Mode)
- DSMPD=0 (Fractional Mode)
- The PD input places the PLL into the lowest power mode. In this case, all analog circuits are turned off and FREF will be "ignored". The FOUTPOSTDIV and FOUTVCO pins are forced to logic low (0V).
- The BYPASS pin controls a mux which selects FREF to be passed to the FOUTPOSTDIV when active high. However, the PLL continues to run as it normally would if bypass were low. This is a useful feature for PLL testing since the clock path can be verified without the PLL being required to work. Also, the effect that the PLL induced supply noise has on the output buffering can be evaluated. It is not recommended to switch between BYPASS mode and normal mode for regular chip operation since this may result in a glitch. Also, FOUTPOSTDIVPD should be set low if the PLL is to be used in BYPASS mode.

2.8.2 PLL frequency change and lock check

The PLL programming supports changed on-the-fly and the PLL will simply slew to the new frequency.
PLL lock state can be checked in CRU_APLL_CON1[10], CRU_DPLL_CON1[10], CRU_CPLL_CON1[10], CRU_GPLL_CON1[10] register. The lock state is high when both original hardware PLL lock and PLL counter lock are high. The PLL counter lock initial value is CRU_GLB_CNT_TH[31:16].
The max delay time is 500 REF_CLK.
PLL locking consists of three phases.

- Phase 1 is control voltage slewing. During this phase one of the clocks (reference or divide) is much faster than the other, and the PLL frequency adjusts almost continuously. When locking from power down, the divide clock is initially very slow and steadily increases frequency. It will take slightly longer for faster VCO settings when locking from power down, since the PLL must slew further.

- Phase 2 is small signal phase acquisition. During this phase, the internal up/down signals alternate semi-chaotically as the phase slowly adjusts until the two signals are aligned. The duration of this phase depends on the loop bandwidth and is faster with higher bandwidth. Bandwidth can be estimated as FREF / REFDIV / 20 for integer mode and FREF /REFDIV / 40 for fractional mode. The duration of small signal locking is about 1/Bandwidth.
- Phase 3 is the digital cycle count. After the last cycle slip is detected, an internal counter waits 256 FREF / REFDIV cycles before the lock signal goes high. This is frequently the dominant factor in lock time – especially for slower reference clock signals or large reference divide settings. This time can be calculated as 256*REFDIV/FREF.

2.8.3 Fractional divider usage

To get specific frequency, clocks of I2S, PDM, UART can be generated by fractional divider. Generally you must set that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock like I2S, UART and PDM. For implementation issue, the input source clocks of fractional divider also have the following limitation.

Table 2-1 Source Clock Limitation of Fractional Divider

Clock Name	Fractional divider source clock Limit
clk_pdm	600MHz
clk_i2s0_tx/rx	600MHz
clk_i2s1/2	600MHz
clk_uart0~5	600MHz
vopb_dclk	600MHz
vopl_dclk	600MHz

2.8.4 Divfree50 divider usage

Some IPs, such as NAND, EMMC, SDIO and SDMMC need clock of 50% duty cycle, divfree50 can generate clock of 50% duty cycle even in odd value divisor.

2.8.5 DivFreeNP5 divider usage

Some IPs, such as GPU and UART need some special frequency can use this divider. Frequency of this divider=clk_src/((2*n+1)/2). Eg, UART with baud rate of 4Mbps need use this divider to generate 64MHz clock from 480MHz of usbphypll.

2.8.6 Global software reset

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0xfdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0xeca8 to assert the second global software reset glb_srstn_2. These two software resets are self-de-asserted by hardware. Resetting hold timing of global software reset (glb_srstn_1, glb_srstn_2, soc_wdt_rstn, soc_tsadc_rstn) can be programmable up to 1ms.
Glb_srstn_1 resets almost all logic.
Glb_srstn_2 resets almost all logic except GRF and GPIOs.

Chapter 3 General Register Files (GRF)

3.1 Overview

The general register file will be used to do static set by software, which is composed of many registers for system control. The GRF is located at several addresses.

- GRF, used for general non-secure system,
- PMUGRF, used for always on system,
- CORE_GRF, used for core pvtm and performance monitor
- GPU_GRF, used for gpu configuration and performance monitor
- USBPHY_GRF, used for usbphy configuration
- DDR_GRF, used for controlling ip in PD_DDR

3.2 Function Description

The function of general register file is：

- IOMUX control
- Control the state of GPIO in power-down mode
- GPIO PAD pull down and pull up control
- Used for common system control
- Used to record the system state

Table 3-1GRF Adress Mapping Table

Name	Address Base
PMUGRF	0xFF010000
GRF	0xFF140000
CORE_GRF	0xFF148000
GPU_GRF	0xFF14C000
USBPHY_GRF	0xFF2C0000
DDR_GRF	0xFF630000

3.3 GRF Register Description

3.3.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.3.2 Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO1A_IOMUX_L	0x0000	W	0x00000000	GPIO1A iomux control low bits
GRF_GPIO1A_IOMUX_H	0x0004	W	0x00000000	GPIO1A iomux control high bits
GRF_GPIO1B_IOMUX_L	0x0008	W	0x00000000	GPIO1B iomux control low bits
GRF_GPIO1B_IOMUX_H	0x000c	W	0x00000000	GPIO1B iomux control high bits
GRF_GPIO1C_IOMUX_L	0x0010	W	0x00000000	GPIO1C iomux control low bits
GRF_GPIO1C_IOMUX_H	0x0014	W	0x00000000	GPIO1C iomux control high bits
GRF_GPIO1D_IOMUX_L	0x0018	W	0x00002200	GPIO1D iomux control low bits
GRF_GPIO1D_IOMUX_H	0x001c	W	0x00000033	GPIO1D iomux control high bits
GRF_GPIO2A_IOMUX_L	0x0020	W	0x00000000	GPIO2A iomux control low bits
GRF_GPIO2A_IOMUX_H	0x0024	W	0x00000000	GPIO2A iomux control high bits

Name	Offset	Size	Reset Value	Description
GRF_GPIO2B_IOMUX_L	0x0028	W	0x00000000	GPIO2B iomux control low bits
GRF_GPIO2B_IOMUX_H	0x002c	W	0x00000000	GPIO2B iomux control high bits
GRF_GPIO2C_IOMUX_L	0x0030	W	0x00000000	GPIO2C iomux control low bits
GRF_GPIO2C_IOMUX_H	0x0034	W	0x00000000	GPIO2C iomux control high bits
GRF_GPIO3A_IOMUX_L	0x0040	W	0x00000000	GPIO3A iomux control low bits
GRF_GPIO3A_IOMUX_H	0x0044	W	0x00000000	GPIO3A iomux control high bits
GRF_GPIO3B_IOMUX_L	0x0048	W	0x00000000	GPIO3B iomux control low bits
GRF_GPIO3B_IOMUX_H	0x004c	W	0x00000000	GPIO3B iomux control high bits
GRF_GPIO3C_IOMUX_L	0x0050	W	0x00000000	GPIO3C iomux control low bits
GRF_GPIO3C_IOMUX_H	0x0054	W	0x00000000	GPIO3C iomux control high bits
GRF_GPIO3D_IOMUX_L	0x0058	W	0x00000000	GPIO3D iomux control low bits
GRF_GPIO3D_IOMUX_H	0x005c	W	0x00000000	GPIO3D iomux control high bits
GRF_GPIO1A_P	0x0060	W	0x00005555	GPIO1A PU/PD control
GRF_GPIO1B_P	0x0064	W	0x00005695	GPIO1B PU/PD control
GRF_GPIO1C_P	0x0068	W	0x00005955	GPIO1C PU/PD control
GRF_GPIO1D_P	0x006c	W	0x00006555	GPIO1D PU/PD control
GRF_GPIO2A_P	0x0070	W	0x0000aaaa	GPIO2A PU/PD control
GRF_GPIO2B_P	0x0074	W	0x00006aaa	GPIO2B PU/PD control
GRF_GPIO2C_P	0x0078	W	0x00002aa9	GPIO2C PU/PD control
GRF_GPIO3A_P	0x0080	W	0x0000aaaa	GPIO3A PU/PD control
GRF_GPIO3B_P	0x0084	W	0x0000aaaa	GPIO3B PU/PD control
GRF_GPIO3C_P	0x0088	W	0x0000aaaa	GPIO3C PU/PD control
GRF_GPIO3D_P	0x008c	W	0x000000aa	GPIO3D PU/PD control
GRF_GPIO1A_SR	0x0090	W	0x00000000	GPIO1A slow rate control
GRF_GPIO1B_SR	0x0094	W	0x00000000	GPIO1B slow rate control
GRF_GPIO1C_SR	0x0098	W	0x00000000	GPIO1C slow rate control
GRF_GPIO1D_SR	0x009c	W	0x00000000	GPIO1D slow rate control
GRF_GPIO2A_SR	0x00a0	W	0x00000000	GPIO2A slow rate control
GRF_GPIO2B_SR	0x00a4	W	0x00000000	GPIO2B slow rate control
GRF_GPIO2C_SR	0x00a8	W	0x00000000	GPIO2C slow rate control
GRF_GPIO3A_SR	0x00b0	W	0x00000000	GPIO3A slow rate control
GRF_GPIO3B_SR	0x00b4	W	0x00000000	GPIO3B slow rate control
GRF_GPIO3C_SR	0x00b8	W	0x00000000	GPIO3C slow rate control
GRF_GPIO3D_SR	0x00bc	W	0x00000000	GPIO3D slow rate control
GRF_GPIO1A_SMT	0x00c0	W	0x00000000	GPIO1A smitter control
GRF_GPIO1B_SMT	0x00c4	W	0x00000000	GPIO1B smitter control
GRF_GPIO1C_SMT	0x00c8	W	0x00000000	GPIO1C smitter control
GRF_GPIO1D_SMT	0x00cc	W	0x00000000	GPIO1D smitter control
GRF_GPIO2A_SMT	0x00d0	W	0x00000000	GPIO2A smitter control
GRF_GPIO2B_SMT	0x00d4	W	0x00000000	GPIO2B smitter control
GRF_GPIO2C_SMT	0x00d8	W	0x00000000	GPIO2C smitter control
GRF_GPIO3A_SMT	0x00e0	W	0x00000000	GPIO3A smitter control

Name	Offset	Size	Reset Value	Description
GRF_GPIO3B_SMT	0x00e4	W	0x00000000	GPIO3B smitter control
GRF_GPIO3C_SMT	0x00e8	W	0x00000000	GPIO3C smitter control
GRF_GPIO3D_SMT	0x00ec	W	0x00000000	GPIO3D smitter control
GRF_GPIO1A_E	0x00f0	W	0x0000aaaa	GPIO1A driver strengh control
GRF_GPIO1B_E	0x00f4	W	0x0000aaaa	GPIO1B driver strengh control
GRF_GPIO1C_E	0x00f8	W	0x0000aa55	GPIO1C driver strengh control
GRF_GPIO1D_E	0x00fc	W	0x0000aaaa	GPIO1D driver strengh control
GRF_GPIO2A_E	0x0100	W	0x00005555	GPIO2A driver strengh control
GRF_GPIO2B_E	0x0104	W	0x00000000	GPIO2B driver strengh control
GRF_GPIO2C_E	0x0108	W	0x00001554	GPIO2C driver strengh control
GRF_GPIO3A_E	0x0110	W	0x00005555	GPIO3A driver strengh control
GRF_GPIO3B_E	0x0114	W	0x00005555	GPIO3B driver strengh control
GRF_GPIO3C_E	0x0118	W	0x00005555	GPIO3C driver strengh control
GRF_GPIO3D_E	0x011c	W	0x00000055	GPIO3D driver strengh control
GRF_IO_VSEL	0x0180	W	0x00000000	IO Voltage Seletion register
GRF_IOFUNC_CON0	0x0184	W	0x00000000	io function control register0
GRF_SOC_CON0	0x0400	W	0x00000000	SOC control register0
GRF_SOC_CON1	0x0404	W	0x00000000	SOC control register1
GRF_SOC_CON2	0x0408	W	0x00001000	SOC control register2
GRF_SOC_CON3	0x040c	W	0x00000000	SOC control register3
GRF_SOC_CON4	0x0410	W	0x00000000	SOC control register4
GRF_SOC_CON5	0x0414	W	0x00000000	SOC control register5
GRF_PD_VI_CON	0x0430	W	0x00000000	PD_VI control register
GRF_PD_VO_CON0	0x0434	W	0x00000000	PD_VO control register0
GRF_PD_VO_CON1	0x0438	W	0x00000000	PD_VO control register1
GRF_SOC_STATUS0	0x0480	W	0x00000000	SOC status register0
GRF_CPU_CON0	0x0500	W	0x00000060	CPU control register0
GRF_CPU_CON1	0x0504	W	0x0000008c	CPU control register1
GRF_CPU_CON2	0x0508	W	0x00000021	CPU control register2
GRF_CPU_STATUS0	0x0520	W	0x00000000	CPU status register0
GRF_CPU_STATUS1	0x0524	W	0x00000000	CPU status register1
GRF_SOC_NOC_CON0	0x0530	W	0x00000000	NOC control register0
GRF_SOC_NOC_CON1	0x0534	W	0x00000000	NOC control register1
GRF_DDR_BANKHASH_CTRL	0x0550	W	0x00000000	DDR BANK HASH control register0
GRF_DDR_BANK_MASK0	0x0554	W	0x00000000	The MSB mask for the first bank bit
GRF_DDR_BANK_MASK1	0x0558	W	0x00000000	The MSB mask for the second bank bit
GRF_DDR_BANK_MASK2	0x055c	W	0x00000000	The MSB mask for the third bank bit
GRF_HOST0_CON0	0x0700	W	0x00000820	USB host control register0
GRF_HOST0_CON1	0x0704	W	0x000004bc	USB host control register1

Name	Offset	Size	Reset Value	Description
GRF_OTG_CON3	0x0880	W	0x00000000	OTG control register
GRF_HOST0_STATUS4	0x0890	W	0x00000000	USB host status register
GRF_MAC_CON1	0x0904	W	0x00000000	MAC control register1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.3.3 Detail Register Description

GRF_GPIO1A_IOMUX_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1a3_sel 4'h0: gpio 4'h1: flash_d3 4'h2: emmc_d3 4'h3: sfc_sio3
11:8	RW	0x0	gpio1a2_sel 4'h0: gpio 4'h1: flash_d2 4'h2: emmc_d2 4'h3: sfc_sio2
7:4	RW	0x0	gpio1a1_sel 4'h0: gpio 4'h1: flash_d1 4'h2: emmc_d1 4'h3: sfc_sio1
3:0	RW	0x0	gpio1a0_sel 4'h0: gpio 4'h1: flash_d0 4'h2: emmc_d0 4'h3: sfc_sio0

GRF_GPIO1A_IOMUX_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1a7_sel 4'h0: gpio 4'h1: flash_d7 4'h2: emmc_d7
11:8	RW	0x0	gpio1a6_sel 4'h0: gpio 4'h1: flash_d6 4'h2: emmc_d6
7:4	RW	0x0	gpio1a5_sel 4'h0: gpio 4'h1: flash_d5 4'h2: emmc_d5
3:0	RW	0x0	gpio1a4_sel 4'h0: gpio 4'h1: flash_d4 4'h2: emmc_d4 4'h3: sfc_csn0

GRF_GPIO1B_IOMUX_L
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1b3_sel 4'h0: gpio 4'h1: flash_ale 4'h2: emmc_rstn
11:8	RW	0x0	gpio1b2_sel 4'h0: gpio 4'h1: flash_dqs 4'h2: emmc_cmd

Bit	Attr	Reset Value	Description
7:4	RW	0x0	gpio1b1_sel 4'h0: gpio 4'h1: flash_rdy 4'h2: emmc_clkout 4'h3: sfc_clk
3:0	RW	0x0	gpio1b0_sel 4'h0: gpio 4'h1: flash_cs0 4'h2: emmc_pwren

GRF_GPIO1B_IOMUX_H
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1b7_sel 4'h0: gpio 4'h1: flash_rdn 4'h2: uart3_rxm1 4'h3: spi0_clk
11:8	RW	0x0	gpio1b6_sel 4'h0: gpio 4'h1: flash_cs1 4'h2: uart3_txm1 4'h3: spi0_csn
7:4	RW	0x0	gpio1b5_sel 4'h0: gpio 4'h1: flash_wrn 4'h2: uart3_rtsm1 4'h3: spi0_miso 4'h4: i2c3_scl
3:0	RW	0x0	gpio1b4_sel 4'h0: gpio 4'h1: flash_cle 4'h2: uart3_ctsm1 4'h3: spi0_mosi 4'h4: i2c3_sda

GRF_GPIO1C_IOMUX_L
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1c3_sel 4'h0: gpio 4'h1: uart1_rts
11:8	RW	0x0	gpio1c2_sel 4'h0: gpio 4'h1: uart1_cts
7:4	RW	0x0	gpio1c1_sel 4'h0: gpio 4'h1: uart1_tx
3:0	RW	0x0	gpio1c0_sel 4'h0: gpio 4'h1: uart1_rx

GRF_GPIO1C_IOMUX_H
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio1c7_sel 4'h0: gpio 4'h1: sdio_d1
11:8	RW	0x0	gpio1c6_sel 4'h0: gpio 4'h1: sdio_d0
7:4	RW	0x0	gpio1c5_sel 4'h0: gpio 4'h1: sdio_clk

Bit	Attr	Reset Value	Description
3:0	RW	0x0	gpio1c4_sel 4'h0: gpio 4'h1: sdio_cmd

GRF_GPIO1D_IOMUX_L
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x2	gpio1d3_sel 4'h0: gpio 4'h1: sdmmc_d1 4'h2: uart2dbg_rxm0
11:8	RW	0x2	gpio1d2_sel 4'h0: gpio 4'h1: sdmmc_d0 4'h2: uart2dbg_txm0
7:4	RW	0x0	gpio1d1_sel 4'h0: gpio 4'h1: sdio_d3
3:0	RW	0x0	gpio1d0_sel 4'h0: gpio 4'h1: sdio_d2

GRF_GPIO1D_IOMUX_H
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:12	RW	0x0	gpio1d7_sel 4'h0: gpio 4'h1: sdmmc_cmd 4'h2: uart4_rts
11:8	RW	0x0	gpio1d6_sel 4'h0: gpio 4'h1: sdmmc_clkout 4'h2: uart4_cts 4'h3: test_clk0
7:4	RW	0x3	gpio1d5_sel 4'h0: gpio 4'h1: sdmmc_d3 4'h2: uart4_tx 4'h3: jtag_tms
3:0	RW	0x3	gpio1d4_sel 4'h0: gpio 4'h1: sdmmc_d2 4'h2: uart4_rx 4'h3: jtag_tck

GRF_GPIO2A_IOMUX_L
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio2a3_sel 4'h0: gpio 4'h1: cif_d5m0 4'h2: rmii_rxd0
11:8	RW	0x0	gpio2a2_sel 4'h0: gpio 4'h1: cif_d4m0 4'h2: rmii_txd0
7:4	RW	0x0	gpio2a1_sel 4'h0: gpio 4'h1: cif_d3m0 4'h2: rmii_txd1

Bit	Attr	Reset Value	Description
3:0	RW	0x0	gpio2a0_sel 4'h0: gpio 4'h1: cif_d2m0 4'h2: rmii_txen

GRF_GPIO2A_IOMUX_H
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio2a7_sel 4'h0: gpio 4'h1: cif_d9m0 4'h2: rmii_mdio
11:8	RW	0x0	gpio2a6_sel 4'h0: gpio 4'h1: cif_d8m0 4'h2: rmii_rxdv
7:4	RW	0x0	gpio2a5_sel 4'h0: gpio 4'h1: cif_d7m0 4'h2: rmii_rxer
3:0	RW	0x0	gpio2a4_sel 4'h0: gpio 4'h1: cif_d6m0 4'h2: rmii_rxd1

GRF_GPIO2B_IOMUX_L
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:12	RW	0x0	gpio2b3_sel 4'h0: gpio 4'h1: cif_clkoutm0 4'h2: clk_out_ethernet
11:8	RW	0x0	gpio2b2_sel 4'h0: gpio 4'h1: cif_clkinm0 4'h2: rmii_clk
7:4	RW	0x0	gpio2b1_sel 4'h0: gpio 4'h1: cif_hrefm0 4'h2: rmii_mdc
3:0	RW	0x0	gpio2b0_sel 4'h0: gpio 4'h1: cif_vsyncm0

GRF_GPIO2B_IOMUX_H
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio2b7_sel 4'h0: gpio 4'h1: cif_d10m0 4'h2: i2c2_scl
11:8	RW	0x0	gpio2b6_sel 4'h0: gpio 4'h1: cif_d1m0 4'h2: uart2_rxm1
7:4	RW	0x0	gpio2b5_sel 4'h0: gpio 4'h1: pwm2
3:0	RW	0x0	gpio2b4_sel 4'h0: gpio 4'h1: cif_d0m0 4'h2: uart2_txm1

GRF_GPIO2C_IOMUX_L

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio2c3_sel 4'h0: gpio 4'h1: i2s1_2ch_mclk
11:8	RW	0x0	gpio2c2_sel 4'h0: gpio 4'h1: i2s1_2ch_sclk
7:4	RW	0x0	gpio2c1_sel 4'h0: gpio 4'h1: i2s1_2ch_lrck
3:0	RW	0x0	gpio2c0_sel 4'h0: gpio 4'h1: cif_d11m0 4'h2: i2c2_sda

GRF_GPIO2C_IOMUX_H

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio2c7_sel 4'h0: gpio
11:8	RW	0x0	gpio2c6_sel 4'h0: gpio 4'h1: pdm_clk0m1
7:4	RW	0x0	gpio2c5_sel 4'h0: gpio 4'h1: i2s1_2ch_sdi 4'h2: pdm_sdi0m1

Bit	Attr	Reset Value	Description
3:0	RW	0x0	gpio2c4_sel 4'h0: gpio 4'h1: i2s1_2ch_sdo

GRF_GPIO3A_IOMUX_L
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3a3_sel 4'h0: gpio 4'h1: lcdc_denm0 4'h2: i2s2_2ch_lrck 4'h3: cif_d2m1 4'h4: uart5_cts
11:8	RW	0x0	gpio3a2_sel 4'h0: gpio 4'h1: lcdc_vsyncm0 4'h2: i2s2_2ch_sclk 4'h3: cif_d1m1 4'h4: uart5_tx
7:4	RW	0x0	gpio3a1_sel 4'h0: gpio 4'h1: lcdc_hsyncm0 4'h2: i2s2_2ch_mclk 4'h3: cif_d0m1 4'h4: uart5_rx
3:0	RW	0x0	gpio3a0_sel 4'h0: gpio 4'h1: lcdc_clk

GRF_GPIO3A_IOMUX_H
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3a7_sel 4'h0: gpio 4'h1: lcdc_d3m0 4'h2: i2s2_2ch_sdo 4'h3: cif_d4m1
11:8	RW	0x0	gpio3a6_sel 4'h0: gpio 4'h1: lcdc_d2
7:4	RW	0x0	gpio3a5_sel 4'h0: gpio 4'h1: lcdc_d1m0 4'h2: i2s2_2ch_sdi 4'h3: cif_d3m1 4'h4: uart5_rts
3:0	RW	0x0	gpio3a4_sel 4'h0: gpio 4'h1: lcdc_d0

GRF_GPIO3B_IOMUX_L

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3b3_sel 4'h0: gpio 4'h1: lcdc_d7 4'h2: i2s0_8ch_sdi1
11:8	RW	0x0	gpio3b2_sel 4'h0: gpio 4'h1: lcdc_d6 4'h2: spi1_cs1

Bit	Attr	Reset Value	Description
7:4	RW	0x0	gpio3b1_sel 4'h0: gpio 4'h1: lcdc_d5m0 4'h2: i2s0_8ch_sdi2 4'h3: cif_d6m1 4'h4: spi1_csn
3:0	RW	0x0	gpio3b0_sel 4'h0: gpio 4'h1: lcdc_d4m0 4'h2: i2s0_8ch_sdi3 4'h3: cif_d5m1

GRF_GPIO3B_IOMUX_H
Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3b7_sel 4'h0: gpio 4'h1: lcdc_d11m0 4'h2: i2s0_8ch_sdo2 4'h3: cif_d9m1 4'h4: spi1_clk
11:8	RW	0x0	gpio3b6_sel 4'h0: gpio 4'h1: lcdc_d10m0 4'h2: i2s0_8ch_sdo3 4'h3: cif_d8m1 4'h4: spi1_miso
7:4	RW	0x0	gpio3b5_sel 4'h0: gpio 4'h1: lcdc_d9m0 4'h2: i2s0_8ch_lrckrx
3:0	RW	0x0	gpio3b4_sel 4'h0: gpio 4'h1: lcdc_d8m0 4'h2: i2s0_8ch_sclkrx 4'h3: cif_d7m1 4'h4: spi1_mosi

GRF_GPIO3C_IOMUX_L
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3c3_sel 4'h0: gpio 4'h1: lcdc_d15 4'h2: i2s0_8ch_sclktx 4'h3: pwm_5
11:8	RW	0x0	gpio3c2_sel 4'h0: gpio 4'h1: lcdc_d14 4'h2: i2s0_8ch_lrcktx 4'h3: pwm_4
7:4	RW	0x0	gpio3c1_sel 4'h0: gpio 4'h1: lcdc_d13 4'h2: i2s0_8ch_mclk
3:0	RW	0x0	gpio3c0_sel 4'h0: gpio 4'h1: lcdc_d12 4'h2: i2s0_8ch_sdo1

GRF_GPIO3C_IOMUX_H
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:12	RW	0x0	gpio3c7_sel 4'h0: gpio 4'h1: lcdc_d19 4'h2: pdm_clk1 4'h3: cif_d11m1
11:8	RW	0x0	gpio3c6_sel 4'h0: gpio 4'h1: lcdc_d18 4'h2: pdm_clk0m0 4'h3: cif_d10m1
7:4	RW	0x0	gpio3c5_sel 4'h0: gpio 4'h1: lcdc_d17 4'h2: i2s0_8ch_sdi0 4'h3: pwm_7
3:0	RW	0x0	gpio3c4_sel 4'h0: gpio 4'h1: lcdc_d16 4'h2: i2s0_8ch_sdo0 4'h3: pwm_6

GRF_GPIO3D_IOMUX_L
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3d3_sel 4'h0: gpio 4'h1: lcdc_d23 4'h2: pdm_sdi0m0 4'h3: cif_clkinm1 4'h4: isp_fl_trig
11:8	RW	0x0	gpio3d2_sel 4'h0: gpio 4'h1: lcdc_d22 4'h2: pdm_sdi3 4'h3: cif_hrefm1 4'h4: isp_flash_trig

Bit	Attr	Reset Value	Description
7:4	RW	0x0	gpio3d1_sel 4'h0: gpio 4'h1: lcdc_d21 4'h2: pdm_sdi2 4'h3: cif_vsyncm1 4'h4: isp_prelight_trig
3:0	RW	0x0	gpio3d0_sel 4'h0: gpio 4'h1: lcdc_d20 4'h2: pdm_sdi1 4'h3: cif_clkoutm1

GRF_GPIO3D_IOMUX_H
Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	gpio3d7_sel 4'h0: gpio
11:8	RW	0x0	gpio3d6_sel 4'h0: gpio
7:4	RW	0x0	gpio3d5_sel 4'h0: gpio
3:0	RW	0x0	gpio3d4_sel 4'h0: gpio 4'h0: osc_gpi

GRF_GPIO1A_P
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description

15:14	RW	0x1	gpio1a7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x1	gpio1a6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x1	gpio1a5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x1	gpio1a4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x1	gpio1a3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x1	gpio1a2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x1	gpio1a1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x1	gpio1a0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO1B_P
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x1	gpio1b7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x1	gpio1b6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x1	gpio1b5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio1b4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio1b3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x1	gpio1b2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x1	gpio1b1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

1:0	RW	0x1	gpio1b0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO1C_P
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x1	gpio1c7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x1	gpio1c6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio1c5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x1	gpio1c4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x1	gpio1c3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

5:4	RW	0x1	gpio1c2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x1	gpio1c1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x1	gpio1c0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO1D_P
Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x1	gpio1d7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x2	gpio1d6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x1	gpio1d5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

9:8	RW	0x1	gpio1d4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x1	gpio1d3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x1	gpio1d2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x1	gpio1d1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x1	gpio1d0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2A_P
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x2	gpio2a7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

13:12	RW	0x2	gpio2a6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio2a5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio2a4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio2a3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio2a2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio2a1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x2	gpio2a0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2B_P
Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x1	gpio2b7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x2	gpio2b6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio2b5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio2b4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio2b3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio2b2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio2b1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

1:0	RW	0x2	gpio2b0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2C_P
Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x0	gpio2c7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x2	gpio2c6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio2c5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio2c4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio2c3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

5:4	RW	0x2	gpio2c2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio2c1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x1	gpio2c0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3A_P
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x2	gpio3a7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x2	gpio3a6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio3a5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

9:8	RW	0x2	gpio3a4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio3a3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio3a2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio3a1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x2	gpio3a0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3B_P
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x2	gpio3b7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

13:12	RW	0x2	gpio3b6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio3b5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio3b4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio3b3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio3b2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio3b1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x2	gpio3b0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3C_P
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x2	gpio3c7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x2	gpio3c6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x2	gpio3c5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio3c4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio3c3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio3c2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio3c1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

1:0	RW	0x2	gpio3c0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3D_P
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x0	gpio3d7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x0	gpio3d6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x0	gpio3d5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x0	gpio3d4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio3d3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

5:4	RW	0x2	gpio3d2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio3d1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x2	gpio3d0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO1A_SR
Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1a7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio1a6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio1a5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio1a4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio1a3_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio1a2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio1a1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio1a0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1B_SR
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1b7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio1b6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio1b5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio1b4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio1b3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio1b2_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio1b1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio1b0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1C_SR
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1c7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio1c6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio1c5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio1c4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio1c3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio1c2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio1c1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio1c0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1D_SR

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1d7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio1d6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio1d5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio1d4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio1d3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio1d2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio1d1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio1d0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO2A_SR

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio2a7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio2a6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio2a5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio2a4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio2a3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio2a2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio2a1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio2a0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO2B_SR
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved
7	RW	0x0	gpio2b7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio2b6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio2b5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio2b4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio2b3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio2b2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio2b1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio2b0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO2C_SR
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio2c7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio2c6_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
5	RW	0x0	gpio2c5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio2c4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio2c3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio2c2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio2c1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio2c0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3A_SR
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3a7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio3a6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio3a5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio3a4_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3a3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio3a2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio3a1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio3a0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3B_SR
Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3b7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio3b6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio3b5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio3b4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio3b3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio3b2_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio3b1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio3b0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3C_SR
Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3c7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio3c6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio3c5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio3c4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio3c3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio3c2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio3c1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio3c0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3D_SR
Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3d7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio3d6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio3d5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio3d4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio3d3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio3d2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio3d1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio3d0_sr 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1A_SMT
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1a7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio1a6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio1a5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio1a4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio1a3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio1a2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio1a1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio1a0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO1B_SMT
Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved
7	RW	0x0	gpio1b7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio1b6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio1b5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio1b4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio1b3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio1b2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio1b1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio1b0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO1C_SMT
Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1c7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio1c6_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
5	RW	0x0	gpio1c5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio1c4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio1c3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio1c2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio1c1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio1c0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO1D_SMT
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio1d7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio1d6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio1d5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio1d4_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio1d3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio1d2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio1d1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio1d0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO2A_SMT
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio2a7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio2a6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio2a5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio2a4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio2a3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio2a2_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio2a1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio2a0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO2B_SMT
Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio2b7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio2b6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio2b5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio2b4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio2b3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio2b2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio2b1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio2b0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO2C_SMT
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio2c7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio2c6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio2c5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio2c4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio2c3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio2c2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio2c1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio2c0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO3A_SMT
Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3a7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio3a6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio3a5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio3a4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio3a3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio3a2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio3a1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio3a0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO3B_SMT
Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RO	0x0	reserved
7	RW	0x0	gpio3b7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio3b6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio3b5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio3b4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio3b3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio3b2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio3b1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio3b0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO3C_SMT
Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3c7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio3c6_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
5	RW	0x0	gpio3c5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio3c4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio3c3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio3c2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio3c1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio3c0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO3D_SMT
Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio3d7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio3d6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio3d5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio3d4_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3d3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio3d2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio3d1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio3d0_smt 0: No hysteresis 1: Schmitt trigger enabled

GRF_GPIO1A_E
Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	gpio1a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x2	gpio1a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x2	gpio1a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio1a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x2	gpio1a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio1a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio1a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x2	gpio1a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1B_E
Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	gpio1b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x2	gpio1b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
11:10	RW	0x2	gpio1b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio1b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x2	gpio1b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio1b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio1b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x2	gpio1b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1C_E
Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x2	gpio1c7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x2	gpio1c6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x2	gpio1c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio1c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio1c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio1c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio1c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio1c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1D_E
Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x2	gpio1d7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x2	gpio1d6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x2	gpio1d5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio1d4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x2	gpio1d3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio1d2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio1d1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

1:0	RW	0x2	gpio1d0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2A_E
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio2a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x1	gpio2a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x1	gpio2a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x1	gpio2a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio2a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio2a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio2a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio2a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2B_E
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio2b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio2b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio2b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio2b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio2b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio2b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio2b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2C_E
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2c7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
13:12	RW	0x1	gpio2c6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x1	gpio2c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x1	gpio2c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio2c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio2c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio2c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio2c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3A_E
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio3a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x1	gpio3a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x1	gpio3a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x1	gpio3a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio3a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio3a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio3a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio3a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3B_E
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio3b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x1	gpio3b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x1	gpio3b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x1	gpio3b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio3b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio3b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio3b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio3b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3C_E
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	gpio3c7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x1	gpio3c6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x1	gpio3c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
9:8	RW	0x1	gpio3c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio3c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio3c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio3c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio3c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3D_E
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3d7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio3d6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio3d5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio3d4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio3d3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio3d2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x1	gpio3d1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio3d0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_IO_VSEL
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	grf_vccio_oscgpi_vsel IO voltage select 1'b0:3.3V 1'b1:1.8V
6	RW	0x0	grf_vccio5_vsel VCC IO5 voltage select 1'b0:3.3V 1'b1:1.8V
5	RW	0x0	grf_vccio4_vsel VCC IO4 voltage select 1'b0:3.3V 1'b1:1.8V
4	RW	0x0	grf_vccio3_vsel VCC IO3 voltage select 1'b0:3.3V 1'b1:1.8V
3	RW	0x0	grf_vccio2_vsel VCC IO2 voltage select 1'b0:3.3V 1'b1:1.8V
2	RW	0x0	grf_vccio1_vsel VCC IO1 voltage select 1'b0:3.3V 1'b1:1.8V
1	RW	0x0	grf_vccio6_vsel VCC IO6 voltage select 1'b0:3.3V 1'b1:1.8V
0	RW	0x0	grf_vccio6_vsel_src VCC IO6 voltage source select 1'b0: controlled by GPIO0B6 1'b1: controlled by grf_vccio6_vsel

GRF_IOFUNC_CON0
Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	grf_con_i2s0_iomode_sel SCLK input of I2S0 source selection 2'b00: sclk_rx is from GPIO3B4, sclk_tx is from GPIO3C3 2'b01: sclk_rx is from GPIO3C3, sclk_tx is from GPIO3C3 2'b10: sclk_rx is from GPIO3B4, sclk_tx is from GPIO3B4 2'b11: both sclk_rx and sclk_tx are from I2S1 SCLK output LRCK input of I2S0 source selection 2'b00: lrck_rx is from GPIO3B5, lrck_tx is from GPIO3C2 2'b01: lrck_rx is from GPIO3C2, lrck_tx is from GPIO3C2 2'b10: lrck_rx is from GPIO3B5, lrck_tx is from GPIO3B5 2'b11: both lrck_rx and lrck_tx are from I2S1 lrck(tx or rx, grf_iofunc_sel[0]) output

11:10	RW	0x0	grf_con_uart2_iomux_sel
			2'b00: m0(tx:gpio1d2,rx,gpio1d3);
			2'b01: m1(tx:gpio2b4,rx:gpio2b6);
			2'b10,2'b11: USBPHY uart debug port;

9	RW	0x0	grf_con_uart3_iomux_sel 1'b0: m0(tx:gpio0c0,rx,gpio0c1,cts:gpio0c2,rts:gpio0c3); 1'b1: m1(tx:gpio1b6,rx:gpio1b7,cts:gpio1b4,rts:gpio1b5)
8	RW	0x0	grf_con_pdm_iomux_sel 1'b0: m0(gpio3c6,gpio2c5); 1'b1: m1(gpio2c6,gpio3c3)

Bit	Attr	Reset Value	Description
7	RW	0x0	grf_con_cif_iomux_sel 1'b0:m0,1'b1:m1; M0, M1, cif_d0m0 gpio2b4, gpio3a1 cif_d1m0 gpio2b6, gpio3a2 cif_d2m0 gpio2a0, gpio3a3 cif_d3m0 gpio2a1, gpio3a5 cif_d4m0 gpio2a2, gpio3a7 cif_d5m0 gpio2a3, gpio3b0 cif_d6m0 gpio2a4, gpio3b1 cif_d7m0 gpio2a5, gpio3b4 cif_d8m0 gpio2a6, gpio3b6 cif_d9m0 gpio2a7, gpio3b7 cif_d10m0 gpio2b7, gpio3c6 cif_d11m0 gpio2c0, gpio3c7 cif_vsyncm0 gpio2b0, gpio3d1 cif_hrefm0 gpio2b1, gpio3d2 cif_clkinm0 gpio2b2, gpio3d3 cif_clkoutm0 gpio2b3, gpio3d0
6	RW	0x0	grf_con_pwm0_vopb_sel PWM1 io output selection: 1'b1: VOP pwm output; 1'b0: PWM controller output
5	RW	0x0	grf_con_i2s0_sclk_sel 1'b1: tx mode; 1'b0: rx mode
4:2	RO	0x0	reserved
1	RW	0x0	grf_con_i2s2_2ch_lrck 1'b1: tx mode; 1'b0: rx mode
0	RW	0x0	grf_con_i2s1_2ch_lrck 1'b1: tx mode 1'b0: rx mode

GRF_SOC_CON0

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	grf_tsadc_testbit_h tsadc_testbit_h bit register

GRF_SOC_CON1
Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	grf_tsadc_testbit_l tsadc_testbit_l bit register

GRF_SOC_CON2
Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x1	grf_con_wdtns_glb_reset_en WDT NS global reset enable. 1'b0: WDTNS cann't trigger reset. 1'b1: WDTNS can trigger reset
11	RW	0x0	grf_con_uart5_rts_inv uart5_rts_inv_selection
10	RW	0x0	grf_con_uart5_cts_inv uart5_cts_inv_selection
9	RW	0x0	grf_con_uart4_rts_inv uart4_rts_inv_selection
8	RW	0x0	grf_con_uart4_cts_inv uart4_cts_inv_selection
7	RW	0x0	grf_con_uart3_rts_inv uart3_rts_inv_selection

Bit	Attr	Reset Value	Description
6	RW	0x0	grf_con_uart3_cts_inv uart3_cts_inv_selection
5	RW	0x0	grf_con_uart2_rts_inv uart2_rts_inv_selection
4	RW	0x0	grf_con_uart2_cts_inv uart2_cts_inv_selection
3	RW	0x0	grf_con_uart1_rts_inv uart1_rts_inv_selection
2	RW	0x0	grf_con_uart1_cts_inv uart1_cts_inv_selection
1	RW	0x0	grf_con_tsadc_ch_inv tsadc_ch_inv
0	RW	0x0	grf_con_gpll_clk_sel 1'b1: GPLL output bypass level shifter. 1'b0: GPLL output has level shifter

GRF_SOC_CON3
Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_con_id_cif_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
14	RW	0x0	grf_con_id_cif_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
13	RW	0x0	grf_con_id_gpu_aw1 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
12	RW	0x0	grf_con_id_gpu_ar1 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
11	RW	0x0	grf_con_id_gpu_aw0 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module

Bit	Attr	Reset Value	Description
10	RW	0x0	grf_con_id_gpu_ar0 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
9	RW	0x0	grf_con_id_gmac_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
8	RW	0x0	grf_con_id_gmac_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
7	RW	0x0	grf_con_id_dma_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
6	RW	0x0	grf_con_id_dma_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
5	RW	0x0	grf_con_id_dcf_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
4	RW	0x0	grf_con_id_dcf_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
3	RW	0x0	grf_con_id_crypto_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
2	RW	0x0	grf_con_id_crypto_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
1	RW	0x0	grf_con_id_cpu_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
0	RW	0x0	grf_con_id_cpu_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module

GRF_SOC_CON4
Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	grf_con_hevc_vcode_sel Selection for AXI BUS interface of hevc_codec connected to niu 1'b0: vcodec 1'b1: HEVC
14	RW	0x0	grf_con_pmu_pwr_idle_req 1'b1, Requst to idle niu in PD_PMU. 1'b0, No request
13	RW	0x0	grf_con_id_vpu_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
12	RW	0x0	grf_con_id_vpu_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
11	RW	0x0	grf_con_id_vops_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
10	RW	0x0	grf_con_id_vops_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
9	RW	0x0	grf_con_id_vopm_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
8	RW	0x0	grf_con_id_vopm_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
7	RW	0x0	grf_con_id_rga_aw 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
6	RW	0x0	grf_con_id_rga_ar 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
5	RW	0x0	grf_con_id_isp_aw_m1 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
4	RW	0x0	grf_con_id_isp_ar_rd 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
3	RW	0x0	grf_con_id_isp_aw_m3 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
2	RW	0x0	grf_con_id_isp_ar_m3 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module

Bit	Attr	Reset Value	Description
1	RW	0x0	grf_con_id_isp_aw_m2 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module
0	RW	0x0	grf_con_id_isp_ar_m2 1'b1, axi access to ddr stored into buffer. 1'b0, axi access to ddr bypass ddrbuffer module

GRF_SOC_CON5
Address: Operational Base + offset (0x0414)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_con_sdcard_dectn_dly Delay counter setting after sdcard plug out. Count by 24M clock

GRF_PD_VI_CON
Address: Operational Base + offset (0x0430)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved

14:13	RW	0x0	grf_con_isp_cif_if_datawidth
			2'b00: 8bit;
			2'b01: 10bit;
			others: 12bit;

12	RW	0x0	grf_con_cif_clk_inv_sel 1'b1: clock inverted; 1'b0: clock is not inverted
11:10	RO	0x0	reserved
9	RW	0x0	grf_con_csiphy_clkinv_selection 1'b1: enable csiphy clock lane; 1'b0: disable csiphy clock lane
8	RW	0x0	grf_con_csiphy_clklane_en 1'b1: enable csiphy lane0; 1'b0: disable csiphy_lane4
7	RW	0x0	grf_con_csiphy_datalane_en_3 1'b1: enable csiphy lane0; 1'b0: disable csiphy_lane3

Bit	Attr	Reset Value	Description
6	RW	0x0	grf_con_csiphy_datalane_en_2 1'b1: enable csiphy lane0; 1'b0: disable csiphy_lane2
5	RW	0x0	grf_con_csiphy_datalane_en_1 1'b1: enable csiphy lane0; 1'b0: disable csiphy_lane1
4	RW	0x0	grf_con_csiphy_datalane_en_0 1'b1: enable csiphy lane0; 1'b0: disable csiphy_lane0
3	RW	0x0	grf_con_csiphy_forcerxmode_3 1'b1: force to rx mode; 1'b0: disable force control
2	RW	0x0	grf_con_csiphy_forcerxmode_2 1'b1: force to rx mode; 1'b0: disable force control
1	RW	0x0	grf_con_csiphy_forcerxmode_1 1'b1: force to rx mode; 1'b0: disable force control
0	RW	0x0	grf_con_csiphy_forcerxmode_0 1'b1: force to rx mode; 1'b0: disable force control

GRF_PD_VO_CON0

Address: Operational Base + offset (0x0434)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	grf_con_vops_press control bit for NIU in PD_VO
13:12	RW	0x0	grf_con_vopm_press control bit for NIU in PD_VO
11:10	RW	0x0	grf_con_dcf_vop_standby_sel 2'b00: ((aclk_vopm_en | dsp_hold_vopm) & (aclk_vops_en | dsp_hold_vops)) 2'b01: aclk_vopm_en | dsp_hold_vopm Others: aclk_vops_en | dsp_hold_vops ;

Bit	Attr	Reset Value	Description
9	RW	0x0	grf_con_lvds_den_only_tie_value Only valid when grf_con_lvds_den_only == 1 1'b1: LVDS vsync/hsync are tied to 1 1'b0: LVDS vsync/hsync are tied to 0
8	RW	0x0	grf_con_lvds_den_only 1'b1: LVDS vsync/hsync are tied to grf_con_lvds_den_only_tie_value 1'b0: LVDS vsync/hsync normal output
7	RW	0x0	grf_con_dsihost_dpiupdatecfg dsihost_dpiupdatecfg
6	RO	0x0	reserved
5	RW	0x0	grf_con_lvds_dclk_inv_sel 1'b1: LVDS Clock is inverted. 1'b0: Normal clock.
4	RO	0x0	reserved
3	RW	0x0	grf_con_dsihost_dpicolorm dsihost_dpicolorm
2	RW	0x0	grf_con_dsihost_dpishutdn dsihost_dpishutdn
1	RW	0x0	grf_con_vopl_dma_finish_enable 1'b1: dma_finish from vopl to dcf is enabled; 1'b0: dma_finish from vopl to dcf is disable
0	RW	0x0	grf_con_vopb_dma_finish_enable 1'b1: dma_finish from vopb to dcf is enabled; 1'b0: dma_finish from vopb to dcf is disable

GRF_PD_VO_CON1
Address: Operational Base + offset (0x0438)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:13	RW	0x0	grf_con_lvdsformat_lvds_select 2'b00: VESA 24bit 2'b01: JEIDA 24bit 2'b10: JEIDA 18bit 2'b11: VESA 18bit

Bit	Attr	Reset Value	Description
12	RW	0x0	grf_con_dsiphy_lvds_mode grf_con_dsiphy_lvds_mode 1'b1: lvds mode 1'b0: dsi mode
11	RW	0x0	grf_con_lvdsformat_lvds_msbsel 1'b1:MSB, 1'b0: LSB
10	RW	0x0	grf_con_dsiphy_lane3_frctxstpm grf_con_dsiphy_lane1_frctxstpm
9	RW	0x0	grf_con_dsiphy_lane2_frctxstpm grf_con_dsiphy_lane1_frctxstpm
8	RW	0x0	grf_con_dsiphy_lane1_frctxstpm grf_con_dsiphy_lane1_frctxstpm
7	RW	0x0	grf_con_dsiphy_lane0_frctxstpm grf_con_dsiphy_lane0_frctxstpm
6	RW	0x0	grf_con_dsiphy_forcerxmode grf_con_dsiphy_forcerxmode
5	RW	0x0	grf_con_dsiphy_lane0_turndisable Disable Turn-around. This signal is used to prevent Lane from going into transmit mode, even if it observes a turn-around request on the Lane interconnect.
4	RW	0x0	grf_con_lcdc_dclk_inv_sel 1'b0: normal clock 1'b1: inverted clock
3	RW	0x0	grf_con_rgb_bypass 1'b1: bypass data sync, 1'b0: use data sync
2	RW	0x0	grf_con_rgb_vop_sel 1'b0: vopb, 1'b1: vopl
1	RW	0x0	grf_con_lvds_vop_sel 1'b0: vopb, 1'b1: vopl
0	RW	0x0	grf_con_dsihost_vop_sel 1'b0: vopb, 1'b1: vopl

GRF_SOC_STATUS0
Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RO	0x0	pmu_pwr_idle_ack Niu idle acknowledge status
19	RO	0x0	pmu_pwr_idle Niu idle status
18	RO	0x0	vopl_dma_finish vopl_dma_finish_status

Bit	Attr	Reset Value	Description
17	RO	0x0	vopb_dma_finish vopb_dma_finish_status
16	RO	0x0	timer_en_status5 timer5_en_status
15	RO	0x0	timer_en_status4 timer4_en_status
14	RO	0x0	timer_en_status3 timer3_en_status
13	RO	0x0	timer_en_status2 timer2_en_status
12	RO	0x0	timer_en_status1 timer1_en_status
11	RO	0x0	timer_en_status0 timer0_en_status
10	RO	0x0	reserved
9	RO	0x0	opt_sbpi_busy_ns opt_sbpi_busy_ns bit register
8	RO	0x0	opt_user_busy_ns opt_user_busy_ns bit register
7	RO	0x0	opt_sbpi_busy_s opt_sbpi_busy_s bit register
6	RO	0x0	opt_user_busy_s opt_user_busy_s bit register
5	RO	0x0	reserved
4	RO	0x0	npll_lock NPLL lock status
3	RO	0x0	gpll_lock GPLL lock status
2	RO	0x0	cpll_lock CPLL lock status
1	RO	0x0	dpll_lock DPLL lock status
0	RO	0x0	apll_lock APLL lock status

GRF_CPU_CON0
Address: Operational Base + offset (0x0500)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	grf_con_cfgte cpu cfgte bit control
11:8	RW	0x0	grf_con_cfgend cpu cfgend bit control
7	RO	0x0	reserved
6	RW	0x1	grf_con_qnap_dis_dataram qnap_dis_dataram bit control
5	RW	0x1	grf_con_qnap_dis_tagram qnap_dis_tagram bit control
4	RW	0x0	grf_con_l2rstdisable cpu dbgl1 rstdisable
3:0	RW	0x0	grf_con_l1rstdisable cpu dbgl1 rstdisable

GRF_CPU_CON1
Address: Operational Base + offset (0x0504)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x1	grf_force_jtag force jtag bit control
6	RW	0x0	grf_con_cpu_ema_detect_en 1'b1: When grf_ema_l2d/grf_emaw_l2d/grf_ema_ra/grf_emaw_ra/grf_emas_ra changed, hardware automaticly stops cpu and make it valiable to cpu; 1'b0: Disable
5	RW	0x0	grf_con_evento_clear pd_core evento_ack control bit

Bit	Attr	Reset Value	Description
4	RW	0x0	grf_con_eventi pd_core eventi control bit
3	RW	0x1	grf_con_dbgselfaddrv cpu dbgselfaddrv bit control
2	RW	0x1	grf_con_dbgromaddrv cpu dbgromaddrv bit control
1	RW	0x0	grf_con_cfgsdisable cpu cfgsdisbale bit control
0	RW	0x0	grf_con_clrexmonreq cpu clrexmonreq bit control

GRF_CPU_CON2
Address: Operational Base + offset (0x0508)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:11	RO	0x0	reserved
10	WO	0x0	grf_emas_ra cpu sram emas
9:8	RW	0x0	grf_emaw_ra cpu sram emaw
7:5	RW	0x1	grf_ema_ra cpu sram ema
4:3	RW	0x0	grf_emaw_l2d cpu l2 data sram emaw
2:0	RW	0x1	grf_ema_l2d cpu l2 data sram ema

GRF_CPU_STATUS0
Address: Operational Base + offset (0x0520)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:15	RO	0x00	grf_st_cpu_boost_fsm cpu boost module status
14	RO	0x0	grf_st_l2flushdone l2flushdone status
13	RO	0x0	grf_st_clrexmonack clrexmonack status

Bit	Attr	Reset Value	Description
12	RO	0x0	grf_st_jtagnsw jtagnsw_st status
11	RO	0x0	grf_st_jtagtop jtagtop_st status
10	RO	0x0	evento_rising_edge evento_rising_edge status
9:4	RO	0x0	reserved
3:0	RO	0x0	grf_st_smpnamp smpnamp status

GRF_CPU_STATUS1
Address: Operational Base + offset (0x0524)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	grf_st_standbywfil2 standby wfi l2 status
11:8	RO	0x0	reserved
7:4	RO	0x0	grf_st_standbywfi standby wfi status
3:0	RO	0x0	grf_st_standbywfe standby wfe status

GRF_SOC_NOC_CON0
Address: Operational Base + offset (0x0530)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	dtp_vpu_req_msch_pwrDiscTarg_Switch1PwrStall dtp_vpu_req_msch_pwrDiscTarg_Switch1PwrStall
13	RW	0x0	dtp_vo_req_msch_pwrDiscTarg_Switch1PwrStall dtp_vo_req_msch_pwrDiscTarg_Switch1PwrStall
12	RW	0x0	dtp_vo_fwd_vi_pwrDiscTarg_Switch47PwrStall dtp_vo_fwd_vi_pwrDiscTarg_Switch47PwrStall
11	RW	0x0	dtp_vi_req_msch_pwrDiscTarg_Switch1PwrStall dtp_vi_req_msch_pwrDiscTarg_Switch1PwrStall
10	RW	0x0	dtp_peri_req_msch_pwrDiscTarg_Switch8PwrStall dtp_peri_req_msch_pwrDiscTarg_Switch8PwrStall

Bit	Attr	Reset Value	Description
9	RW	0x0	dtp_gpu_req_msch_pwrDiscTarg_Switch29PwrStall dtp_gpu_req_msch_pwrDiscTarg_Switch29PwrStall
8	RW	0x0	dtp_cpu_req_msch_pwrDiscTarg_Switch28PwrStall dtp_cpu_req_msch_pwrDiscTarg_Switch28PwrStall
7	RW	0x0	dtp_cpu_fwd_bus_pwrDiscTarg_Switch18PwrStall dtp_cpu_fwd_bus_pwrDiscTarg_Switch18PwrStall
6	RW	0x0	dtp_bus_fwd_vpu_pwrDiscTarg_Switch44PwrStall dtp_bus_fwd_vpu_pwrDiscTarg_Switch44PwrStall
5	RW	0x0	dtp_bus_fwd_vio_pwrDiscTarg_Switch21PwrStall dtp_bus_fwd_vio_pwrDiscTarg_Switch21PwrStall
4	RW	0x0	dtp_bus_fwd_srvmsch_pwrDiscTarg_service_msch_TPwrStall dtp_bus_fwd_srvmsch_pwrDiscTarg_service_msch_TPwrStall
3	RW	0x0	dtp_bus_fwd_peri_pwrDiscTarg_Switch15PwrStall dtp_bus_fwd_peri_pwrDiscTarg_Switch15PwrStall
2	RW	0x0	dtp_bus_fwd_gpu_pwrDiscTarg_Switch46PwrStall dtp_bus_fwd_gpu_pwrDiscTarg_Switch46PwrStall
1	RW	0x0	dtp_bus_fwd_ddrc_pwrDiscTarg_ddrc_apb_TPwrStall dtp_bus_fwd_ddrc_pwrDiscTarg_ddrc_apb_TPwrStall
0	RW	0x0	dtp_Switch26_pwrDiscTarg_peri2msch_service_TPwrStall dtp_Switch26_pwrDiscTarg_peri2msch_service_TPwrStall

GRF_SOC_NOC_CON1
Address: Operational Base + offset (0x0534)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	msch_split_size This register will decide the splitting size of the interconnect for a transaction from a master and must be set to a proper value according to 'ddrConf' in memory scheduler reginster. 2'b00: Splitting size is 64 bytes. Must be set to this value when 'ddrConf' is 0~6 or 12~13. 2'b01: Splitting size is 32 bytes. Must be set to this value when 'ddrConf' is 9~10. 2'b10: Splitting size is 16 bytes. Must be set to this value when 'ddrConf' is 7~8 or 11. 2'b11: Reserved. Do not set to this value, otherwise, the system will crash

Bit	Attr	Reset Value	Description
13	RW	0x0	dtp_usb_fwd_peri_pwrDiscTarg_Switch9PwrStall dtp_usb_fwd_peri_pwrDiscTarg_Switch9PwrStall
12	RW	0x0	dtp_sdmmc_fwd_peri_pwrDiscTarg_Switch56PwrStall dtp_sdmmc_fwd_peri_pwrDiscTarg_Switch56PwrStall
11	RW	0x0	dtp_peri_fwd_usb_pwrDiscTarg_Switch43PwrStall dtp_peri_fwd_usb_pwrDiscTarg_Switch43PwrStall
10	RW	0x0	dtp_peri_fwd_nand_pwrDiscTarg_Switch42PwrStall dtp_peri_fwd_nand_pwrDiscTarg_Switch42PwrStall
9	RW	0x0	dtp_peri_fwd_mmc_pwrDiscTarg_Switch41PwrStall dtp_peri_fwd_mmc_pwrDiscTarg_Switch41PwrStall
8	RW	0x0	dtp_peri_fwd_gmac_pwrDiscTarg_Switch40PwrStall dtp_peri_fwd_gmac_pwrDiscTarg_Switch40PwrStall
7	RW	0x0	dtp_nand_fwd_peri_pwrDiscTarg_Switch56PwrStall dtp_nand_fwd_peri_pwrDiscTarg_Switch56PwrStall
6	RW	0x0	dtp_mmc_fwd_peri_pwrDiscTarg_Switch55PwrStall dtp_mmc_fwd_peri_pwrDiscTarg_Switch55PwrStall
5	RW	0x0	dtp_gmac_fwd_peri_pwrDiscTarg_Switch55PwrStall dtp_gmac_fwd_peri_pwrDiscTarg_Switch55PwrStall
4	RW	0x0	dtp_crypto_fwd_bus_pwrDiscTarg_Switch11PwrStall dtp_crypto_fwd_bus_pwrDiscTarg_Switch11PwrStall
3	RW	0x0	dtp_bus_fwd_sdcard_pwrDiscTarg_SwitchPwrStall dtp_bus_fwd_sdcard_pwrDiscTarg_SwitchPwrStall
2	RW	0x0	dtp_bus_fwd_pmu_pwrDiscTarg_pmu_apb_TPwrStall dtp_bus_fwd_pmu_pwrDiscTarg_pmu_apb_TPwrStall
1	RW	0x0	dtp_bus_fwd_peri_pwrDiscTarg_Switch16PwrStall dtp_bus_fwd_peri_pwrDiscTarg_Switch16PwrStall
0	RW	0x0	dtp_bus_fwd_crypto_pwrDiscTarg_Switch45PwrStall dtp_bus_fwd_crypto_pwrDiscTarg_Switch45PwrStall

GRF_DDR_BANKHASH_CTRL
Address: Operational Base + offset (0x0550)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x0	bank_offset set the offset of the first bank bit. The real first bank offset bit is 10+bank_offset. For example, if you are using the follwoing ddrConf, set this register to 3: 'RRRRRRRRRRRRRRRRBBBCCCCCCCCC----'
3:1	RW	0x0	manicure_mask bank manicure mask bits 3'b000: when using 4 banks ddr, set to this value 3'b111: when using 8 banks ddr, set to this value others: reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	hash_en bank hash enable control 1'b0: disable 1'b1: enable

GRF_DDR_BANK_MASK0
Address: Operational Base + offset (0x0554)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ddr_bank_mask0 The MSB mask for the first bank bit The bits below R3(the third bit of row bits) must be set to 0 when using 8 banks device. The bits below R0 must be set to 0 when using 4 banks device.

GRF_DDR_BANK_MASK1
Address: Operational Base + offset (0x0558)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ddr_bank_mask1 The MSB mask for the second bank bit. The bits below R3(the third bit of row bits) must be set to 0 when using 8 banks device. The bits below R0 must be set to 0 when using 4 banks device.

GRF_DDR_BANK_MASK2
Address: Operational Base + offset (0x055c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ddr_bank_mask2 The MSB mask for the third bank bit. The bits below R3(the third bit of row bits) must be set to 0 when using 8 banks device. The bits below R0 must be set to 0 when using 4 banks device.

GRF_HOST0_CON0
Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:6	RW	0x20	grf_con_host0_fladj_val_common USB HOST0 fladj_val_common bit control
5:0	RW	0x20	grf_con_host0_fladj_val USB HOST0 fladj bit control

GRF_HOST0_CON1
Address: Operational Base + offset (0x0704)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13	RW	0x0	grf_con_host0_arb_pause host0 ehci/ohci arbiter pause control
12	RW	0x0	grf_con_host0_ohci_susp_lgcy USB HOST0 ohci_susp_lgcy bit control
11	RW	0x0	grf_con_host0_ohci_cntsel USB HOST0 ohci_cntsel bit control
10	RW	0x1	grf_con_host0_ohci_clkcktrst USB HOST0 ohci_clkcktrst bit control
9	RW	0x0	grf_con_host0_app_prt_ovrcur USB HOST0 app_prt_ovrcur bit control
8	RW	0x0	grf_con_host0_autoppd_on_overcur_en USB HOST0 autoppd_on_overcur_en bit control
7	RW	0x1	grf_con_host0_word_if USB HOST0 word_if bit control
6	RW	0x0	grf_con_host0_sim_mode USB HOST0 sim_mode bit control
5	RW	0x1	grf_con_host0_incrx_en USB HOST0 incrx_en bit control
4	RW	0x1	grf_con_host0_incr8_en USB HOST0 incr8_en bit control
3	RW	0x1	grf_con_host0_incr4_en USB HOST0 incr4_en bit control
2	RW	0x1	grf_con_host0_incr16_en USB HOST0 incr16_en bit control
1	RW	0x0	grf_con_host0_hubsetup_min USB HOST0 bubsetup_min bit control

Bit	Attr	Reset Value	Description
0	RW	0x0	grf_con_host0_app_start_clk USB HOST0 app_start_clk bit control

GRF_OTG_CON3
Address: Operational Base + offset (0x0880)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:3	RO	0x0	reserved
2	RW	0x0	otg_dbnce_fltr_bypass OTG dbnce_fltr_bypass bit control
1:0	RW	0x0	otg_scaledown_mode OTG scaledown_mode bit control

GRF_HOST0_STATUS4
Address: Operational Base + offset (0x0890)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	host0_ehci_power_state_ack host0_ehci_power_state_ack bit status
29	RO	0x0	host0_ehci_pme_status host0_ehci_pme_status bit status
28	RO	0x0	grf_stat_host0_ehci_bufacc host0_ehci_bufacc bit status
27	RO	0x0	grf_stat_host0_ehci_xfer_prdc host0_ehci_xfer_prdc bit status
26	RO	0x0	grf_stat_host0_ohci_ccs host0_ohci_ccs bit status
25	RO	0x0	grf_stat_host0_ohci_rwe host0_ohci_rwe bit status
24	RO	0x0	grf_stat_host0_ohci_drwe host0_ohci_drwe bit status
23	RO	0x0	grf_stat_host0_ohci_globalsuspend host0_ohci_globalsuspend bit status
22	RO	0x0	grf_stat_host0_ohci_bufacc host0_ohci_bufacc bit status
21	RO	0x0	grf_stat_host0_ohci_rmtwkp host0_ohci_rmtwkp bit status

Bit	Attr	Reset Value	Description
20:17	RO	0x0	grf_stat_host0_ehci_lpsmc_state host0_ehci_lpsmc_state bit status
16:11	RO	0x00	grf_stat_host0_ehci_usbsts host0_ehci_usbsts bit status
10:0	RO	0x000	grf_stat_host0_ehci_xfer_cnt host0_ehci_xfer_cnt bit status

GRF_MAC_CON1
Address: Operational Base + offset (0x0904)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved
6:4	RW	0x0	gmac2io_phy_intf_sel PHY interface select 3'b001:RGMII 3'b100:RMII All others:Reserved
3	RW	0x0	gmac2io_flowctrl GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frame in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
2	RW	0x0	gmac2io_mac_speed MAC speed 1'b1:100-Mbps 1'b0:10-Mbps
1:0	RO	0x0	reserved

3.4 PMUGRF Register Description

3.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
PMUGRF_GPIO0A_IOMUX	0x0000	W	0x00001040	GPIO0A iomux control bits
PMUGRF_GPIO0B_IOMUX	0x0004	W	0x00001000	GPIO0B iomux control bits
PMUGRF_GPIO0C_IOMUX	0x0008	W	0x00000000	GPIO0C iomux control bits
PMUGRF_GPIO0A_P	0x0010	W	0x0000466a	GPIO0A PU/PD control
PMUGRF_GPIO0B_P	0x0014	W	0x000095a5	GPIO0B PU/PD control
PMUGRF_GPIO0C_P	0x0018	W	0x000000aa	GPIO0C PU/PD control
PMUGRF_GPIO0A_E	0x0020	W	0x00000001	GPIO0A driver strengh control
PMUGRF_GPIO0B_E	0x0024	W	0x00000000	GPIO0B driver strengh control
PMUGRF_GPIO0C_E	0x0028	W	0x00000000	GPIO0C driver strengh control
PMUGRF_GPIO0L_SR	0x0030	W	0x00000000	GPIO0 slow rate control low bits
PMUGRF_GPIO0H_SR	0x0034	W	0x00000000	GPIO0 slow rate control high bits
PMUGRF_GPIO0L_SMT	0x0038	W	0x00000000	GPIO0 smitter control low bits
PMUGRF_GPIO0H_SMT	0x003c	W	0x00000000	GPIO0 smitter control high bits
PMUGRF_SOC_CON0	0x0100	W	0x00000000	PMU SOC control register0
PMUGRF_SOC_CON1	0x0104	W	0x00000000	PMU SOC control register1
PMUGRF_SOC_CON2	0x0108	W	0x00000800	PMU SOC control register2
PMUGRF_FAILSAFE_CON	0x010c	W	0x00000048	FailSafe module configuation
PMUGRF_PVTM_CON0	0x0180	W	0x00000003	PVTM control register0
PMUGRF_PVTM_CON1	0x0184	W	0x00000100	PVTM control register1
PMUGRF_PVTM_STATUS0	0x0190	W	0x00000000	PVTM status register0
PMUGRF_PVTM_STATUS1	0x0194	W	0x00000000	PVTM status register1
PMUGRF_OS_REG0	0x0200	W	0x00000000	pmu grf os register0
PMUGRF_OS_REG1	0x0204	W	0x00000000	pmu grf os register1
PMUGRF_OS_REG2	0x0208	W	0x00000000	pmu grf os register2
PMUGRF_OS_REG3	0x020c	W	0x00000000	pmu grf os register3
PMUGRF_OS_REG4	0x0210	W	0x00000000	pmu grf os register4
PMUGRF_OS_REG5	0x0214	W	0x00000000	pmu grf os register5
PMUGRF_OS_REG6	0x0218	W	0x00000000	pmu grf os register6
PMUGRF_OS_REG7	0x021c	W	0x00000000	pmu grf os register7
PMUGRF_OS_REG8	0x0220	W	0x00000000	pmu grf os register8
PMUGRF_OS_REG9	0x0224	W	0x00000000	pmu grf os register9
PMUGRF_OS_REG10	0x0228	W	0x00000000	pmu grf os register10
PMUGRF_OS_REG11	0x022c	W	0x00000000	pmu grf os register11
PMUGRF_RESET_FUNCTION_STATUS	0x0230	W	0x00000000	system reset status register
PMUGRF_SIG_DETECT_CON	0x0380	W	0x00000000	sdmmc detect control reg
PMUGRF_SIG_DETECT_STATUS	0x0390	W	0x00000000	sdmmc detect status reg
PMUGRF_SIG_DETECT_STATUS_CLEAR	0x03a0	W	0x00000000	sdmmc irq clear reg
PMUGRF_SDMMC_DET_COUNTER	0x03b0	W	0x00030100	sdmmc detect counter reg

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.4.3 Detail Register Description

PMUGRF_GPIO0A_IOMUX
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0a7_sel 2'h0: gpio
13:12	RW	0x1	gpio0a6_sel 2'h0: gpio 2'h1: tsadc_shutm0 2'h2: tsadc_shut_orignal
11:10	RW	0x0	gpio0a5_sel 2'h0: gpio
9:8	RW	0x0	gpio0a4_sel 2'h0: gpio 2'h1: pmic_sleep 2'h2: tsadc_shutm1
7:6	RW	0x1	gpio0a3_sel 2'h0: gpio 2'h1: sdmmc_detn
5:4	RW	0x0	gpio0a2_sel 2'h0: gpio
3:2	RW	0x0	gpio0a1_sel 2'h0: gpio
1:0	RW	0x0	gpio0a0_sel 2'h0: gpio 2'h1: clk_out_wifi

PMUGRF_GPIO0B_IOMUX
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0b7_sel 2'h0: gpio 2'h1: pwm_0 2'h2: otg_drv
13:12	RW	0x1	gpio0b6_sel 2'h0: gpio 2'h1: flash_vol_sel
11:10	RW	0x0	gpio0b5_sel 2'h0: gpio 2'h1: uart0_rts 2'h2: test_clk1
9:8	RW	0x0	gpio0b4_sel 2'h0: gpio 2'h1: uart0_cts 2'h2: pmu_debug2 2'h3: pmu_debug_sout
7:6	RW	0x0	gpio0b3_sel 2'h0: gpio 2'h1: uart0_rx 2'h2: pmu_debug1
5:4	RW	0x0	gpio0b2_sel 2'h0: gpio 2'h1: uart0_tx 2'h2: pmu_debug0
3:2	RW	0x0	gpio0b1_sel 2'h0: gpio 2'h1: i2c0_sda
1:0	RW	0x0	gpio0b0_sel 2'h0: gpio 2'h1: i2c0_scl

PMUGRF_GPIO0C_IOMUX
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0c7_sel 2'h0: gpio
13:12	RW	0x0	gpio0c6_sel 2'h0: gpio
11:10	RW	0x0	gpio0c5_sel 2'h0: gpio 2'h1: osc_gpo
9:8	RW	0x0	gpio0c4_sel 2'h0: gpio 2'h1: clk_inout_32k
7:6	RW	0x0	gpio0c3_sel 2'h0: gpio 2'h1: i2c1_sda 2'h2: uart3_rtsm0
5:4	RW	0x0	gpio0c2_sel 2'h0: gpio 2'h1: i2c1_scl 2'h2: uart3_ctsm0 2'h3: pmu_debug5
3:2	RW	0x0	gpio0c1_sel 2'h0: gpio 2'h1: pwm_3 2'h2: uart3_rxm0 2'h3: pmu_debug4
1:0	RW	0x0	gpio0c0_sel 2'h0: gpio 2'h1: pwm_1 2'h2: uart3_txm0 2'h3: pmu_debug3

PMUGRF_GPIO0A_P
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x1	gpio0a7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x0	gpio0a6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x1	gpio0a5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x2	gpio0a4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x1	gpio0a3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio0a2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio0a1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

1:0	RW	0x2	gpio0a0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

PMUGRF_GPIO0B_P

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x2	gpio0b7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x1	gpio0b6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x1	gpio0b5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

9:8	RW	0x1	gpio0b4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio0b3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

5:4	RW	0x2	gpio0b2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x1	gpio0b1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x1	gpio0b0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

PMUGRF_GPIO0C_P
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

15:14	RW	0x0	gpio0c7_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

13:12	RW	0x0	gpio0c6_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

11:10	RW	0x0	gpio0c5_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

Bit	Attr	Reset Value	Description

9:8	RW	0x0	gpio0c4_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

7:6	RW	0x2	gpio0c3_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

5:4	RW	0x2	gpio0c2_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

3:2	RW	0x2	gpio0c1_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

1:0	RW	0x2	gpio0c0_p
			2'b00: Z(Normal operation);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull_down);
			2'b11: Repeater(Bus keeper)

PMUGRF_GPIO0A_E
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio0a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio0a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio0a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio0a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x1	gpio0a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

PMUGRF_GPIO0B_E
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio0b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio0b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio0b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio0b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio0b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

PMUGRF_GPIO0C_E
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0c7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio0c6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio0c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio0c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio0c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio0c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

PMUGRF_GPIO0L_SR
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpio0b7_sr 1'b0: slow(half frequency) 1'b1: fast
14	RW	0x0	gpio0b6_sr 1'b0: slow(half frequency) 1'b1: fast
13	RW	0x0	gpio0b5_sr 1'b0: slow(half frequency) 1'b1: fast
12	RW	0x0	gpio0b4_sr 1'b0: slow(half frequency) 1'b1: fast
11	RW	0x0	gpio0b3_sr 1'b0: slow(half frequency) 1'b1: fast
10	RW	0x0	gpio0b2_sr 1'b0: slow(half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
9	RW	0x0	gpio0b1_sr 1'b0: slow(half frequency) 1'b1: fast
8	RW	0x0	gpio0b0_sr 1'b0: slow(half frequency) 1'b1: fast
7	RW	0x0	gpio0a7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio0a6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio0a5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio0a4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio0a3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio0a2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio0a1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio0a0_sr 1'b0: slow(half frequency) 1'b1: fast

PMUGRF_GPIO0H_SR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	gpio0c7_sr 1'b0: slow(half frequency) 1'b1: fast
6	RW	0x0	gpio0c6_sr 1'b0: slow(half frequency) 1'b1: fast
5	RW	0x0	gpio0c5_sr 1'b0: slow(half frequency) 1'b1: fast
4	RW	0x0	gpio0c4_sr 1'b0: slow(half frequency) 1'b1: fast
3	RW	0x0	gpio0c3_sr 1'b0: slow(half frequency) 1'b1: fast
2	RW	0x0	gpio0c2_sr 1'b0: slow(half frequency) 1'b1: fast
1	RW	0x0	gpio0c1_sr 1'b0: slow(half frequency) 1'b1: fast
0	RW	0x0	gpio0c0_sr 1'b0: slow(half frequency) 1'b1: fast

PMUGRF_GPIO0L_SMT
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpio0b7_smt 0: No hysteresis 1: Schmitt trigger enabled
14	RW	0x0	gpio0b6_smt 0: No hysteresis 1: Schmitt trigger enabled
13	RW	0x0	gpio0b5_smt 0: No hysteresis 1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio0b4_smt 0: No hysteresis 1: Schmitt trigger enabled
11	RW	0x0	gpio0b3_smt 0: No hysteresis 1: Schmitt trigger enabled
10	RW	0x0	gpio0b2_smt 0: No hysteresis 1: Schmitt trigger enabled
9	RW	0x0	gpio0b1_smt 0: No hysteresis 1: Schmitt trigger enabled
8	RW	0x0	gpio0b0_smt 0: No hysteresis 1: Schmitt trigger enabled
7	RW	0x0	gpio0a7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio0a6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio0a5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio0a4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio0a3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio0a2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio0a1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio0a0_smt 0: No hysteresis 1: Schmitt trigger enabled

PMUGRF_GPIO0H_SMT
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	gpio0c7_smt 0: No hysteresis 1: Schmitt trigger enabled
6	RW	0x0	gpio0c6_smt 0: No hysteresis 1: Schmitt trigger enabled
5	RW	0x0	gpio0c5_smt 0: No hysteresis 1: Schmitt trigger enabled
4	RW	0x0	gpio0c4_smt 0: No hysteresis 1: Schmitt trigger enabled
3	RW	0x0	gpio0c3_smt 0: No hysteresis 1: Schmitt trigger enabled
2	RW	0x0	gpio0c2_smt 0: No hysteresis 1: Schmitt trigger enabled
1	RW	0x0	gpio0c1_smt 0: No hysteresis 1: Schmitt trigger enabled
0	RW	0x0	gpio0c0_smt 0: No hysteresis 1: Schmitt trigger enabled

PMUGRF_SOC_CON0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit0 can be written by software. When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	poc_pmuio2_sel18 PMU VCCIO2 voltage select 1'b0: 3.3V 1'b1: 1.8V
14	RW	0x0	poc_pmuio1_sel18 PMU VCCIO1 voltage select 1'b0: 3.3V 1'b1: 1.8V
13	RW	0x0	ddrphy_bufferen_core 1'b0: enable ddrphy io retention; 1'b1: disable ddrphy io retention;
12	RW	0x0	ddrphy_bufferen_sel 1'b1: ddrphy_bufferen from ddrphy_bufferen_core; 1'b0: ddrphy_bufferen from pmu and ddr_fail_safe
11:7	RO	0x0	reserved
6	RW	0x0	uart0_cts_sel 1'b1: reverse polarity of cts;
5	RW	0x0	uart0_rts_sel 1'b1: reverse polarity of rts;
4:1	RO	0x0	reserved
0	RW	0x0	con_32k_ioe 1'b1: input mode; 1'b0: output mode

PMUGRF_SOC_CON1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	hold_the_ddrfailsafe hold ddrfailsafe reset
11:0	RW	0x000	resetn_hold Please refer to cru_softrst6_con. Each bit has a hold

PMUGRF_SOC_CON2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RW	0x0000	npor_out2chip_pulse_width Pulse width of triggered Npor output. Multipled with XIN_OSC clock period

PMUGRF_FAILSAFE_CON
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	upctl_c_sysreq_cfg 1'b1: always enable requesting DDR controller to enter low power state, when ddr failsafe module is working. 1'b0: After ddr failsafe module enters selfrefresh status, then request DDR controller to enter low power state
7	RO	0x0	reserved
6	RW	0x1	ddr_io_ret_cfg 1'b0: disable ddr io retention during system failure; 1'b1: enable ddr io retention during system failure
5	RW	0x0	ddr_io_ret_de_req 1'b1: request to enter retention, during system failure
4	RW	0x0	ddrc_gating_en 1'b1: enable ddr clock gating during system failure
3	RW	0x1	sref_enter_en 1'b1: enable ddr selfrefresh enter when system is failed
2	RW	0x0	ddrio_ret_en 1'b1: enable ddr io retension when system is failed 1'b0: remain ddr io status when system is failed

1	RW	0x0	wdt_shut_reset_trigger_en
			Enable failsafe wdt input
			1'b1: enable;
			1'b0: disable;

0	RW	0x0	tsadc_shut_reset_trigger_en
			Enable failsafe tsadc input
			1'b1: enable;
			1'b0: disable;

PMUGRF_PVTM_CON0
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:2	RW	0x00	pvtm_clkout_div pvtm_clkout_div
1	RW	0x1	pvtm_pmu_osc_en pvtm_pmu_osc_en 1'b1: enable osc ring in PVTM
0	RW	0x1	pvtm_pmu_start pvtm_pmu_start 1'b1: start pvtm

PMUGRF_PVTM_CON1
Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_pmu_cal_cnt pvtm_pmu_cal_cnt

PMUGRF_PVTM_STATUS0
Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	pvtm_pmu_freq_done pvtm_pmu_freq_done

PMUGRF_PVTM_STATUS1
Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_pmu_freq_cnt pvtm_pmu_freq_cnt

PMUGRF_OS_REG0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg0 reserved

PMUGRF_OS_REG1
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg1 reserved

PMUGRF_OS_REG2
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg2 reserved

PMUGRF_OS_REG3
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg3 reserved

PMUGRF_OS_REG4
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg4 reserved

PMUGRF_OS_REG5
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg5 reserved

PMUGRF_OS_REG6
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg6 reserved

PMUGRF_OS_REG7
Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg7 reserved

PMUGRF_OS_REG8
Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg8 Reserved. Once this reg is written, it can't be reset

PMUGRF_OS_REG9
Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg9 reserved. once this reg is written, it can't be reset

PMUGRF_OS_REG10
Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg10 reserved. once this reg is written, it can't be reset

PMUGRF_OS_REG11
Address: Operational Base + offset (0x022c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_os_reg11 reserved. once this reg is written, it can't be reset

PMUGRF_RESET_FUNCTION_STATUS
Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	st_rstfunc_status 32'H12345678: WDT RESET 32'H23456789: TSADC RESET 32'H3456789A: SOFTWARE RESET

PMUGRF_SIG_DETECT_CON
Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	sdmmc_detectn_neg_irq_msk Enable sdmmc detectn negedge irq 1'b1: enable 1'b0: disable
0	RW	0x0	sdmmc_detectn_pos_irq_msk Enable sdmmc detectn posedge irq 1'b1: enable 1'b0: disable

PMUGRF_SIG_DETECT_STATUS

Address: Operational Base + offset (0x0390)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	sdmmc_detectn_neg_irq 1'b1: irq asserted; 1'b0: no irq
0	RW	0x0	sdmmc_detectn_pos_irq 1'b1: irq asserted; 1'b0: no irq

PMUGRF_SIG_DETECT_STATUS_CLEAR
Address: Operational Base + offset (0x03a0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	WO	0x0	sdmmc_detectn_neg_irq_clr 1'b1: clear irq
0	WO	0x0	sdmmc_detectn_pos_irq_clr 1'b1: clear irq

PMUGRF_SDMMC_DET_COUNTER
Address: Operational Base + offset (0x03b0)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x30100	sdmmc_detectn_count sdmmc_detectn_count bit register

3.5 COREGRF Register Description

3.5.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.5.2 Registers Summary

Name	Offset	Size	Reset Value	Description
COREGRF_CA35_PEFF_CON0	0x0000	W	0x00000000	CA35 performance monitor control register0
COREGRF_CA35_PEFF_CON1	0x0004	W	0x00000000	CA35 performance monitor control register1
COREGRF_CA35_PEFF_CON2	0x0008	W	0x00000000	CA35 performance monitor control register2
COREGRF_CA35_PEFF_CON3	0x000c	W	0x00000000	CA35 performance monitor control register3

Name	Offset	Size	Reset Value	Description
COREGRF_CA35_PEFF_CON4	0x0010	W	0x00000000	CA35 performance monitor control register4
COREGRF_CA35_PEFF_CON5	0x0014	W	0x00000000	CA35 performance monitor control register5
COREGRF_CA35_PEFF_CON6	0x0018	W	0x00000000	CA35 performance monitor control register6
COREGRF_CA35_PEFF_CON7	0x001c	W	0x00000000	CA35 performance monitor control register7
COREGRF_CA35_PEFF_CON8	0x0020	W	0x00000000	CA35 performance monitor control register8
COREGRF_A35_PERF_RD_MAX_LATENCY_NUM	0x0030	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_RD_LATENCY_SAMP_NUM	0x0034	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_RD_LATENCY_ACC_NUM	0x0038	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_RD_AXI_TOTAL_BYTE	0x003c	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_WR_AXI_TOTAL_BYTE	0x0040	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_WORKING_CNT	0x0044	W	0x00000000	CA35 performance monitor status register
COREGRF_A35_PERF_INT_STATUS	0x0048	W	0x00000000	CA35 performance monitor status register
COREGRF_COREPVTM_CON0	0x0080	W	0x00000000	CORE PVTM control register0
COREGRF_COREPVTM_CON1	0x0084	W	0x00000000	CORE PVTM control register1
COREGRF_COREPVTM_STATUS0	0x0088	W	0x00000000	CORE PVTM status register0
COREGRF_COREPVTM_STATUS1	0x008c	W	0x00000000	CORE PVTM status register1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.5.3 Detail Register Description

COREGRF_CA35_PEFF_CON0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	ca35_sw_rd_latency_id_range_e Axi read channel id for latency AXI_PERFormance test
14	RO	0x0	reserved
13:8	RW	0x00	ca35_sw_rd_latency_id 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
7:6	RW	0x0	ca35_sw_ddr_align_type axi_perf counter id control 0: count all write channel id 1: count sw_ar_count_id write channel only
5	RW	0x0	ca35_sw_aw_cnt_id_type axi_perf counter id control 0: count all write channel id 1: count sw_aw_count_id read channel only
4	RW	0x0	ca35_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
3	RW	0x0	ca35_sw_axi_cnt_type_wrap axi_perf counter type wrap 0: no wrap test 1: wrap test
2	RW	0x0	ca35_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	ca35_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	ca35_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

COREGRF_CA35_PEFF_CON1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	ca35_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

COREGRF_CA35_PEFF_CON2
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	ca35_sw_axi_perf_int_clr interrupt clear 1'b1: clear 1'b0: no op
14	RW	0x0	ca35_sw_axi_perf_int_e interrupt enable 1'b1: enable 1'b0: disable
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_aw_count_id When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	ca35_sw_ar_count_id When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

COREGRF_CA35_PEFF_CON3
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	ca35_sw_ar_mon_id mon_id_bmsk bit control
14	RW	0x0	ca35_sw_ar_mon_id_bmsk mon_id_bmsk bit control
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_ar_mon_id_type mon_id_type bit control
7:6	RO	0x0	reserved
5:0	RW	0x00	ca35_sw_ar_mon_id_msk mon_id_msk bit control

COREGRF_CA35_PEFF_CON4
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	ca35_sw_aw_mon_id mon_id_bmsk bit control

Bit	Attr	Reset Value	Description
14	RW	0x0	ca35_sw_aw_mon_id_bmsk mon_id_bmsk bit control
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_aw_mon_id_type mon_id_type bit control
7:6	RO	0x0	reserved
5:0	RW	0x00	ca35_sw_aw_mon_id_msk mon_id_msk bit control

COREGRF_CA35_PEFF_CON5
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ca35_sw_araddr_mon_st monitor read start address

COREGRF_CA35_PEFF_CON6
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ca35_sw_araddr_mon_end monitor read end address

COREGRF_CA35_PEFF_CON7
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ca35_sw_awaddr_mon_st monitor write start address

COREGRF_CA35_PEFF_CON8
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ca35_sw_awaddr_mon_end monitor write end address

COREGRF_A35_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RO	0x0000	rd_max_latency_r axi read max latency output

COREGRF_A35_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RO	0x0000000	rd_latency_samp_r AXI read latency total sample number

COREGRF_A35_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

COREGRF_A35_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rd_axi_total_byte AXI active total read bytes/ddr align read bytes

COREGRF_A35_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	wr_axi_total_byte AXI active total write bytes/ddr align write bytes

COREGRF_A35_PERF_WORKING_CNT
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	working_cnt_r working counter

COREGRF_A35_PERF_INT_STATUS
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RO	0x00	a35_aw_mon_axi_id_status The ID be monitored read from the specific addr area
23:17	RO	0x0	reserved
16	RO	0x0	a35_aw_mon_axi_hit_flag Write from the specific addr area interrupt status
15	RO	0x0	reserved
14:8	RO	0x00	a35_ar_mon_axi_id_status The ID be monitored read from the specific addr area
7:1	RO	0x0	reserved
0	RO	0x0	a35_ar_mon_axi_hit_flag Read from the specific addr area interrupt status

COREGRF_COREPVTM_CON0
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:4	RO	0x0	reserved
3:2	RW	0x0	corepvtm_osc_sel osc_ring selection
1	RW	0x0	corepvtm_osc_en corepvtm_osc_en
0	RW	0x0	corepvtm_start corepvtm_start

COREGRF_COREPVTM_CON1
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	corepvtm_cal_cnt corepvtm_cal_cnt

COREGRF_COREPVTM_STATUS0
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	corepvtm_freq_done corepvtm_freq_done

COREGRF_COREPVTM_STATUS1
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	corepvtm_freq_cnt corepvtm_freq_cnt

3.6 GPUGRF Register Description

3.6.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.6.2 Registers Summary

Name	Offset	Size	Reset Value	Description
GPUGRF_PEFF_CON0	0x0000	W	0x00000000	GPU performance monitor control0
GPUGRF_PEFF_CON1	0x0004	W	0x00000000	GPU performance monitor control0
GPUGRF_PEFF_CON2	0x0008	W	0x00000000	GPU performance monitor control2
GPUGRF_PERF_RD_MAX_LATENCY_NUM	0x0030	W	0x00000000	GPU performance monitor status
GPUGRF_PERF_RD_LATENCY_SAMP_NUM	0x0034	W	0x00000000	GPU performance monitor status
GPUGRF_PERF_RD_LATENCY_ACC_NUM	0x0038	W	0x00000000	GPU performance monitor status
GPUGRF_PERF_RD_AXI_TOTAL_BYTE	0x003c	W	0x00000000	GPU performance monitor status
GPUGRF_PERF_WR_AXI_TOTAL_BYTE	0x0040	W	0x00000000	GPU performance monitor status
GPUGRF_PERF_WORKING_CNT	0x0044	W	0x00000000	GPU performance monitor status
GPUGRF_GPU_CON0	0x0060	W	0x00000040	GPU GRF control

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.6.3 Detail Register Description

GPUGRF_PEFF_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpu_sw_rd_latency_id_range_e gpu_sw_rd_latency_id_range_e Axi read channel id for latency AXI_PERFormance test
14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x00	gpu_sw_rd_latency_id gpu_sw_rd_latency_id 0: 16-Byte align 1: 32-Byte align 2: 64-Byte align 3: 128-Byte align
7:6	RW	0x0	gpu_sw_ddr_align_type gpu_sw_ddr_align_type axi_perf counter id control 0: count all write channel id 1: count sw_ar_count_id write channel only
5	RW	0x0	gpu_sw_aw_cnt_id_type gpu_sw_aw_cnt_id_type axi_perf counter id control 0: count all write channel id 1: count sw_aw_count_id read channel only
4	RW	0x0	gpu_sw_ar_cnt_id_type gpu_sw_ar_cnt_id_type axi_perf counter id control 0: count all read channel id 1: count sw_ar_count_id read channel only
3	RW	0x0	gpu_sw_axi_cnt_type_wrap gpu_sw_axi_cnt_type_wrap axi_perf counter type wrap 0: no wrap test 1: wrap test
2	RW	0x0	gpu_sw_axi_cnt_type gpu_sw_axi_cnt_type axi_perf counter type 0: axi transfer test 1: ddr align transfer test
1	RW	0x0	gpu_sw_axi_perf_clr gpu_sw_axi_perf_clr axi_perf clear bit 0: disable 1: enable
0	RW	0x0	gpu_sw_axi_perf_work gpu_sw_axi_perf_work axi_perf enable bit 0: disable 1: enable

GPUGRF_PEFF_CON1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	gpu_sw_rd_latency_thr gpu_sw_rd_latency_thr Axi read channel id for latency AXI_PERFormance test

GPUGRF_PEFF_CON2
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpu_sw_axi_perf_int_clr gpu_sw_axi_perf_int_clr interrupt clear 1'b1: clear 1'b0: no op
14	RW	0x0	gpu_sw_axi_perf_int_e gpu_sw_axi_perf_int_e interrupt enable 1'b1: enable 1'b0: disable
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x00	gpu_sw_aw_count_id gpu_sw_aw_count_id When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
7:6	RO	0x0	reserved
5:0	RW	0x00	gpu_sw_ar_count_id gpu_sw_ar_count_id When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

GPUGRF_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:0	RO	0x0000	rd_max_latency_r rd_max_latency_r axi read max latency output

GPUGRF_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RO	0x0000000	rd_latency_samp_r rd_latency_samp_r AXI read latency total sample number

GPUGRF_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rd_latency_acc_cnt_r rd_latency_acc_cnt_r AXI read latency (>sw_rd_latency_thr) total number

GPUGRF_PERF_RD_AXI_TOTAL_BYTE

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rd_axi_total_byte rd_axi_total_byte AXI active total read bytes/ddr align read bytes

GPUGRF_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	wr_axi_total_byte wr_axi_total_byte AXI active total write bytes/ddr align write bytes

GPUGRF_PERF_WORKING_CNT
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	working_cnt_r working_cnt_r working counter

GPUGRF_GPU_CON0
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x4	grf_gpu_emaa grf_gpu_emaa SRAM EMAA control
3	RO	0x0	reserved
2:0	RW	0x0	grf_con_dvalin_striping_granule grf_con_dvalin_striping_granule memory striping in level two cache 3'b000: 4KB Select L2C #0, if PA[12] == 0. 3'b001: 128 bytes Select L2C #0, if PA[7]==0. 3'b010: 256 bytes Select L2C #0, if PA[8] == 0. 3'b011: 512 bytes Select L2C #0, if PA[9] == 0. 3'b100: 1KB Select L2C #0, if PA[10] == 0. 3'b101: 2KB Select L2C #0, if PA[11] == 0. 3'b110: This value is reserved. 3'b111: 256B An address hash function is used for striping

3.7 USB PHY GRF Register Description

3.7.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.7.2 Registers Summary

Name	Offset	Size	Reset Value	Description
USBPHY_GRF_REG0	0x0000	W	0x00008518	USB PHY Register0
USBPHY_GRF_REG1	0x0004	W	0x0000e007	USB PHY Register1
USBPHY_GRF_REG2	0x0008	W	0x000002e7	USB PHY Register2
USBPHY_GRF_REG3	0x000c	W	0x00000200	USB PHY Register3
USBPHY_GRF_REG4	0x0010	W	0x00005556	USB PHY Register4
USBPHY_GRF_REG5	0x0014	W	0x00004555	USB PHY Register5
USBPHY_GRF_REG6	0x0018	W	0x00000005	USB PHY Register6
USBPHY_GRF_REG7	0x001c	W	0x000068c0	USB PHY Register7
USBPHY_GRF_REG8	0x0020	W	0x00000000	USB PHY Register8
USBPHY_GRF_REG9	0x0024	W	0x00000000	USB PHY Register9
USBPHY_GRF_REG10	0x0028	W	0x00000000	USB PHY Register10
USBPHY_GRF_REG11	0x002c	W	0x00000000	USB PHY Register11
USBPHY_GRF_REG12	0x0030	W	0x00008518	USB PHY Register12
USBPHY_GRF_REG13	0x0034	W	0x0000e007	USB PHY Register13
USBPHY_GRF_REG14	0x0038	W	0x000002e7	USB PHY Register14
USBPHY_GRF_REG15	0x003c	W	0x00000200	USB PHY Register15
USBPHY_GRF_REG16	0x0040	W	0x00005556	USB PHY Register16
USBPHY_GRF_REG17	0x0044	W	0x00004555	USB PHY Register17
USBPHY_GRF_REG18	0x0048	W	0x00000005	USB PHY Register18
USBPHY_GRF_REG19	0x004c	W	0x000068c0	USB PHY Register19
USBPHY_GRF_REG20	0x0050	W	0x00000000	USB PHY Register20
USBPHY_GRF_REG21	0x0054	W	0x00000000	USB PHY Register21
USBPHY_GRF_REG22	0x0058	W	0x00000000	USB PHY Register22
USBPHY_GRF_REG23	0x005c	W	0x00000000	USB PHY Register23
USBPHY_GRF_CON0	0x0100	W	0x00000452	USB PHY control register0
USBPHY_GRF_CON1	0x0104	W	0x000001d2	USB PHY control register1
USBPHY_GRF_CON2	0x0108	W	0x00000000	USB PHY control register2
USBPHY_GRF_CON3	0x010c	W	0x00000019	USB PHY control register3
USBPHY_GRF_INT_MASK	0x0110	W	0x00000000	USB2PHY interrupt mask register
USBPHY_GRF_INT_STATUS	0x0114	W	0x00000000	USB2PHY interrupt status register
USBPHY_GRF_INT_STATUS_CLR	0x0118	W	0x00000000	USB2PHY interrupt status clear register
USBPHY_GRF_STATUS	0x0120	W	0x00000000	USB2PHY status register
USBPHY_GRF_LS_CON	0x0130	W	0x00030100	USB2PHY linestate control register
USBPHY_GRF_DIS_CON	0x0134	W	0x00030100	USB2PHY disconnect control register
USBPHY_GRF_BVALID_CON	0x0138	W	0x00030100	USB2PHY bvalid control register
USBPHY_GRF_ID_CON	0x013c	W	0x00030100	USB2PHY id control register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.7.3 Detail Register Description

USBPHY_GRF_REG0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x8518	usbphy_reg0 usbcomb phy control reg. BIT15 to 0

USBPHY_GRF_REG1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0xe007	usbphy_reg1 usbcomb phy control reg. BIT31 to 16

USBPHY_GRF_REG2
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x02e7	usbphy_reg2 usbcomb phy control reg. BIT47 to 32

USBPHY_GRF_REG3
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0200	usbphy_reg3 usbcomb phy control reg. BIT63 to 48

USBPHY_GRF_REG4
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x5556	usbphy_reg4 usbcomb phy control reg. BIT79 to 64

USBPHY_GRF_REG5
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x4555	usbphy_reg5 usbcomb phy control reg. BIT95 to 80

USBPHY_GRF_REG6
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0005	usbphy_reg6 usbcomb phy control reg. BIT111 to 96

USBPHY_GRF_REG7
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x68c0	usbphy_reg7 usbcomb phy control reg. BIT127 to 112

USBPHY_GRF_REG8
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg8 usbcomb phy control reg. BIT143 to 128

USBPHY_GRF_REG9
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg9 usbcomb phy control reg. BIT159 to 144

USBPHY_GRF_REG10
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg10 usbcomb phy control reg. BIT175 to 160

USBPHY_GRF_REG11
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg11 usbcomb phy control reg. BIT191 to 176

USBPHY_GRF_REG12
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x8518	usbphy_reg12 usbcomb phy control reg. BIT207 to 192

USBPHY_GRF_REG13
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0xe007	usbphy_reg13 usbcomb phy control reg. BIT223 to 208

USBPHY_GRF_REG14
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x02e7	usbphy_reg14 usbcomb phy control reg. BIT239 to 224

USBPHY_GRF_REG15
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0200	usbphy_reg15 usbcomb phy control reg. BIT255 to 240

USBPHY_GRF_REG16
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x5556	usbphy_reg16 usbcomb phy control reg. BIT271 to 256

USBPHY_GRF_REG17
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x4555	usbphy_reg17 usbcomb phy control reg. BIT287 to 272

USBPHY_GRF_REG18
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0005	usbphy_reg18 usbcomb phy control reg. BIT303 to 288

USBPHY_GRF_REG19
Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x68c0	usbphy_reg19 usbcomb phy control reg. BIT319 to 304

USBPHY_GRF_REG20
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg20 usbcomb phy control reg. BIT335 to 320

USBPHY_GRF_REG21
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg21 usbcomb phy control reg. BIT351 to 336

USBPHY_GRF_REG22
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg22 usbcomb phy control reg. BIT367 to 352

USBPHY_GRF_REG23
Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	usbphy_reg23 usbcomb phy control reg. BIT383 to 368

USBPHY_GRF_CON0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:11	RO	0x0	reserved
10	RW	0x1	usbotg_utmi_iddig GRF USB otg Plug iddig Indicator
9	RW	0x0	usbotg_utmi_iddig_sel USB otg plug indicator output selection 1'b0:select phy iddig status to controller 1'b1: select grf plug iddig indicator to controller
8	RW	0x0	usbotg_utmi_dmpulldown GRF otg DM pulldown resistor

Bit	Attr	Reset Value	Description
7	RW	0x0	usbotg_utmi_dppulldown GRF otg DP pulldown resistor
6	RW	0x1	usbotg_utmi_termselect GRF otg termination select between FS/LS/HS speed
5:4	RW	0x1	usbotg_utmi_xcvrselect GRF otg transceiver select between FS/LS/HS speed
3:2	RW	0x0	usbotg_utmi_opmode GRF otg operational mode selection
1	RW	0x1	usbotg_utmi_suspend_n GRF otg suspend mode 1'b0:suspend 1'b1:normal
0	RW	0x0	usbotg_utmi_sel 1'b0:select otg controller utmi interface to phy 1'b1:select GRF utmi interface to phy

USBPHY_GRF_CON1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:9	RO	0x0	reserved
8	RW	0x1	usbhost_utmi_dmpulldown GRF host DM pulldown resistor
7	RW	0x1	usbhost_utmi_dppulldown GRF host DP pulldown resistor
6	RW	0x1	usbhost_utmi_termselect GRF host termination select between FS/LS/HS speed
5:4	RW	0x1	usbhost_utmi_xcvrselect GRF host transceiver select between FS/LS/HS speed
3:2	RW	0x0	usbhost_utmi_opmode GRF host operational mode selection
1	RW	0x1	usbhost_utmi_suspend_n GRF host suspend mode 1'b0: suspend 1'b1: normal
0	RW	0x0	usbhost_utmi_sel 1'b0: select host controller utmi interface to phy 1'b1: select grf utmi interface to phy

USBPHY_GRF_CON2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:13	RO	0x0	reserved
12	RW	0x0	vdm_src_en_usbotg open dm voltage source
11	RW	0x0	vdp_src_en_usbotg open dp voltage source
10	RW	0x0	rdm_pdwn_en_usbotg open dm pull down resistor
9	RW	0x0	idp_src_en_usbotg open dm source current
8	RW	0x0	idm_sink_en_usbotg open dm sink current
7	RW	0x0	idp_sink_en_usbotg open dp sink current
6:5	RO	0x0	reserved
4	RW	0x0	usbphy_commononn configure PLL clock output in suspend mode 0: 480MHz clock always on 1: 480MHz clock will turn off when both ports suspend asserted. If the supsend of any port deassert, it will wait 1ms to make 480MHz clock stable
3	RW	0x0	bypasssel_usbotg bypass select
2	RW	0x0	bypassdmen_usbotg bypass dm enable
1	RW	0x0	usbotg_disable_1 bypass OTG function
0	RW	0x0	usbotg_disable_0 bypass OTG function

USBPHY_GRF_CON3
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:12	RO	0x0	reserved
11	RW	0x0	usbotg_utmi_drvvbus USB OTG grf utmi_drvvbus
10	RW	0x0	usbotg_utmi_drvvbus_sel USB OTG utmi_drvvbus_sel bit control 0:select otg controller drvvbus to phy 1:select otg grf utmidrvvbus to phy

Bit	Attr	Reset Value	Description
9	RW	0x0	usbotg_utmi_fs_se0 USB OTG utmi_fs_se0 bit control
8	RW	0x0	usbotg_utmi_fs_data USB OTG utmi_fs_data bit control
7	RW	0x0	usbotg_utmi_fs_oe USB OTG utmi_fs_oe bit control
6	RW	0x0	usbotg_utmi_fs_xver_own USB OTG utmi_fs_xver_own bit control
5	RW	0x0	usbhost_utmi_idpullup USB HOST utmi_idpullup bit control
4	RW	0x1	usbhost_utmi_dmpulldown Enable DMINUS Pull Down resistor
3	RW	0x1	usbhost_utmi_dppulldown Enable DPLUS Pull Down resistor
2	RW	0x0	usbhost_utmi_dischrgvbus USB HOST utmi_dischrgvbus bit control
1	RW	0x0	usbhost_utmi_chrgvbus USB HOST utmi_chrgvbus bit control
0	RW	0x1	usbhost_utmi_drvvbus USB HOST utmi_drvvbus bit control

USBPHY_GRF_INT_MASK
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:10	RO	0x0	reserved
9:8	RW	0x0	host0_disconnect_irq_en host0_disconnect_irq edge status enable x1: hostdisconnect rising edge irq status enable 1x: hostdisconnect falling edge irq status enable
7:6	RW	0x0	otg0_disconnect_irq_en otg0_disconnect_irq edge status enable x1: hostdisconnect rising edge irq status enable 1x: hostdisconnect falling edge irq status enable
5:4	RW	0x0	otg0_id_irq_en otg0_id edge status enable x1: id rising edge irq status enable 1x: id falling edge irq status enable
3:2	RW	0x0	otg0_bvalid_irq_en otg0_bvalid edge status irq enable x1: bvalid rising edge irq status enable 1x: bvalid falling edge irq status enable

Bit	Attr	Reset Value	Description
1	RW	0x0	host0_linestate_irq_en host0_linestate change status irq enable
0	RW	0x0	otg0_linestate_irq_en otg0_linestate change status irq enable

USBPHY_GRF_INT_STATUS
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:8	RO	0x0	host0_disconnect_irq host0_disconnect edge irq status x1: hostdisconnect rising edge irq status 1x: hostdisconnect falling edge irq status
7:6	RO	0x0	otg0_disconnect_irq otg0_disconnect edge irq status x1: hostdisconnect rising edge irq status 1x: hostdisconnect falling edge irq status
5:4	RO	0x0	otg0_id_irq otg0_id edge irq status x1: id rising edge irq status 1x: id falling edge irq status
3:2	RO	0x0	otg0_bvalid_irq otg0_bvalid edge irq status x1: bvalid rising edge irq status 1x: bvalid falling edge irq status
1	RO	0x0	host0_linestate_irq host0_linestate change irq status
0	RO	0x0	otg0_linestate_irq otg0_linestate change irq status

USBPHY_GRF_INT_STATUS_CLR
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:8	WO	0x0	host0_disconnect_irq_clr host0_disconnect_irq_clr irq status clear 01: hostdisconnect rising edge irq status clear 10: hostdisconnect falling edge irq status clear
7:6	WO	0x0	otg0_disconnect_irq_clr otg0_disconnect_irq_clr irq status clear 01: hostdisconnect rising edge irq status clear 10: hostdisconnect falling edge irq status clear

Bit	Attr	Reset Value	Description
5:4	WO	0x0	otg0_id_irq_clr otg0_id edge irq status clear 01: id rising edge irq status clear 10: id falling edge irq status clear
3:2	WO	0x0	otg0_bvalid_irq_clr otg0_bvalid edge irq status clear 01: bvalid rising edge irq status clear 10: bvalid falling edge irq status clear
1	WO	0x0	host0_linestate_irq_clr host0_linestate change irq status clear, write 1 to clear irq status
0	WO	0x0	otg0_linestate_irq_clr otg0_linestate change irq status clear, write 1 to clear irq status

USBPHY_GRF_STATUS

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RO	0x0	grf_stat_usbphy_dp_detected grf_stat_usbphy_dp_detected bit status
24	RO	0x0	grf_stat_usbphy_cp_detected grf_stat_usbphy_cp_detected bit status
23	RO	0x0	grf_stat_usbphy_dcp_detected grf_stat_usbphy_dcp_detected bit status
22	RO	0x0	usbhost_phy_ls_fs_rcv host_phy_ls_fs_rcv status
21	RO	0x0	usbhost_utmi_avalid host_utmi_avalid status
20	RO	0x0	usbhost_utmi_bvalid host_utmi_bvalid status
19	RO	0x0	usbhost_utmi_hostdisconnect host_utmi_hostdisconnect status
18	RO	0x0	usbhost_utmi_iddig_o host_utmi_iddig status
17:16	RO	0x0	usbhost_utmi_linestate host_utmi_linestate status
15	RO	0x0	usbhost_utmi_sessend host_utmi_sessend status
14	RO	0x0	usbhost_utmi_vbusvalid host_utmi_vbusvalid status
13	RO	0x0	usbhost_utmi_vmi host_utmi_vmi status
12	RO	0x0	usbhost_utmi_vpi host_utmi_vpi status

Bit	Attr	Reset Value	Description
11	RO	0x0	usbotg_phy_ls_fs_rcv utmi_phy_ls_fs_rcv_out status
10	RO	0x0	usbotg_utmi_avalid otg_utmi avalid bit status
9	RO	0x0	usbotg_utmi_bvalid otg_utmi bvalid bit status
8	RO	0x0	usbotg_utmi_fs_xver_own OTG utmi_fs_xver_own status
7	RO	0x0	usbotg_utmi_hostdisconnect otg_utmi_hostdisconnect status
6	RO	0x0	usbotg_utmi_iddig usbotg_utmi_iddig status
5:4	RO	0x0	usbotg_utmi_linestate otg_utmi_linestate status
3	RO	0x0	usbotg_utmi_sessend otg_utmi_sessend bit status
2	RO	0x0	usbotg_utmi_vbusvalid otg_utmi_vbusvalid bit status
1	RO	0x0	usbotg_utmi_vmi otg_utmi_vmi bit status
0	RO	0x0	usbotg_utmi_vpi otg_utmi_vpi bit status

USBPHY_GRF_LS_CON
Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x30100	linestate_filter_con host/otg port linestate filter time control register. Unit: pclk(up to 100MHz)

USBPHY_GRF_DIS_CON
Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x30100	disconnect_filter_con host/otg port hostdisconnect filter time control register. Unit: pclk(up to 100MHz)

USBPHY_GRF_BVALID_CON
Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x30100	bvalid_filter_con otg port bvalid filter time control register. Unit: pclk(up to 100MHz)

USBPHY_GRF_ID_CON
Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:0	RW	0x0030100	id_filter_con otg port linestate filter time control register. Unit: pclk(up to 100MHz)

3.8 DDRGRF Register Description

3.8.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

3.8.2 Registers Summary

Name	Offset	Size	Reset Value	Description
DDR_GRF_CON0	0x0000	W	0x00000000	DDR Control Register0
DDR_GRF_CON1	0x0004	W	0x00000600	DDR Control Register1
DDR_GRF_SPLIT_CON	0x0008	W	0x00000010	DDR AXI SPLIT Control Register
DDR_GRF_LP_CON	0x0020	W	0x00001101	DDR PHY Lower Power Control Register
DDR_GRF_MSC_CTRL	0x0080	W	0x00000000	MSC_CTRL register
DDR_GRF_CPU_IDLE_TH	0x0084	W	0x00000000	cpu idle threshold register
DDR_GRF_READY_LOW_CYCLES	0x0088	W	0x00000000	read low cyclse register
DDR_GRF_READY_HIGH_CYCLES	0x008c	W	0x00000000	ready high cycles register
DDR_GRF_PRIORITY_IDLE_TH	0x0090	W	0x00000000	piriority idle threshold register
DDR_GRF_PRIORITY_LEVLE_TH	0x0094	W	0x00000000	priority level threshold register
DDR_GRF_STATUS0	0x0100	W	0x00000000	DDR Status Register0
DDR_GRF_STATUS1	0x0104	W	0x00000000	DDR Status Register1
DDR_GRF_STATUS2	0x0108	W	0x00000000	DDR Status Register2
DDR_GRF_STATUS3	0x010c	W	0x00000000	DDR Status Register3
DDR_GRF_STATUS4	0x0110	W	0x00000000	DDR Status Register4

Name	Offset	Size	Reset Value	Description
DDR_GRF_STATUS5	0x0114	W	0x00000000	DDR Status Register5
DDR_GRF_STATUS6	0x0118	W	0x00000000	DDR Status Register6
DDR_GRF_STATUS7	0x011c	W	0x00000000	DDR Status Register7
DDR_GRF_STATUS8	0x0120	W	0x00000000	DDR Status Register8
DDR_GRF_STATUS9	0x0124	W	0x00000000	DDR Status Register9

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

3.8.3 Detail Register Description

DDR_GRF_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable
15	RW	0x0	grf_ddrbuf_en 1: enable ddr_buffer, disable msch_ready_ctrl 0: disable ddr_buffer, enable msch_ready_ctrl
14	RW	0x0	grf_awpoison AXI write poison
13	RW	0x0	grf_awurgent AXI write urgent
12	RW	0x0	grf_arpoison AXI read poison
11	RW	0x0	grf_arurgent AXI read urgent
10	RW	0x0	grf_pa_wmask When asserted(active high), it will prevent the corresponding write to PA
9:8	RW	0x0	grf_pa_rmask When asserted(active high), it will prevent the corresponding read to PA
7:6	RO	0x0	reserved
5	RW	0x0	grf_csysreq_upctl_ddrstdby 0: disable stdby controls upctl csysreq_ddrc 1: enable stdby control upctl csysreq_ddrc
4	RW	0x0	grf_csysreq_upctl_pmu 0: disable pmu controls upctl csysreq_ddrc 1: enable pmu controls upctl csysreq_ddrc
3	RW	0x0	grf_csysreq_aclk 0: request upctl aclk enter low power 1: request upctl aclk exit low power
2	RW	0x0	grf_dfi_init_start grf_dfi_init start value

Bit	Attr	Reset Value	Description
1	RW	0x0	grf_dfi_init_start_sel 1: grf_dfi_init_start controls dfi_init_start 0: upctl controls dfi_init_start
0	RW	0x0	grf_upctl_slverr_enable 0: disable upctl apb slverr response 1: enable upcttl apb slverr response

DDR_GRF_CON1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x6	grf_auto_sr_dly The delay of auto gated ddrc_core_clk. It should be to be 0x6
7:5	RO	0x0	reserved
4	RW	0x0	grf_upctl_syscreq_cg_en 0: disable force ddrc_core_clk ungating when external ddrc_csysreq asserted 1: enable force ddrc_core ungating when external ddrc_csysreq asserted
3	RW	0x0	grf_selfref_type2_en 0: disable ddrc_core_clk auto gating in type2 selfrefresh 1: enable ddrc_core_clk auto gating in type2 selfrefresh
2	RW	0x0	grf_upctl_core_cg_en 0: disable ddrc_core_clk auto gating 1: enable ddrc_core_clk auto gating
1	RW	0x0	grf_upctl_apb_cg_en 0: disable function of force aclk/ddrc_core_clk ungated when apb access is going 1: enable function of force aclk/ddrc_core_clk ungated when apb access is going
0	RW	0x0	grf_upctl_axi_cg_en 0: disable aclk auto gating 1: enable aclk auto gating

DDR_GRF_SPLIT_CON
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable
15:11	RO	0x0	reserved
10:9	RW	0x0	SPMODE Split mode select. 2'b00:DDR controller and phy works at 32 bits mode. Low 16 bits are valid if access address is above split address. 2'b01:DDR controller and phy works at 32 bits mode. High 16 bits are valid if access address is above split address. 2'b10:DDR controller and phy works at 16 bits mode. Low 8 bits are valid if access address is above split address. 2'b11:DDR controller and phy works at 16 bits mode. High 8 bits are valid if access address is above split address
8	RW	0x0	BYPASS 0: enable axi split 1: bypass axi split
7:0	RW	0x10	SPADDR Split address high 8 bits of 32bit address. For example, if SPADDR=0x10, then the split address is 0x10000000. The axi_split module will be bypassed if reading or writing DDR below split address, otherwise axi burst will be split

DDR_GRF_LP_CON
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable
15:14	RO	0x0	reserved
13	RW	0x0	sr_ctl_en 0: disable sr exit/enter reload/inverse lpckdis_ini 1: enable sr exit/eneter reload/inverse lpckdis_ini
12	RW	0x1	pd_ctl_en 0: disable pd exit/enter reload/inverse lpckdis_ini 1: enable pd exit/eneter reload/inverse lpckdis_ini
11:10	RO	0x0	reserved
9	RW	0x0	lpckdis_en 0: disable ddr phy low power fuction 1: enable ddr phy low power function
8	RW	0x1	lpckdis_ini lpckdis intial value
7:3	RO	0x0	reserved
2	RW	0x0	lp23_mode 1: enable LPDDR2/LPDDR3 mode 0: disable LPDDR2/LPDDR3 mode

Bit	Attr	Reset Value	Description
1	RW	0x0	ddr4_mode 1: enable DDR4 mode 0: disable DDR4 mode
0	RW	0x1	ddr23_mode 1: enable DDR2/DDR3 mode 0: disable DDR2/DDR3 mode

DDR_GRF_MSC_CTRL
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	write_enable Bit0~3 write enable
15:4	RO	0x0	reserved
3	RW	0x0	read_bypass_en 1'b1: bypass all read traffics 1'b0: not bypass
2	RW	0x0	priority_bypass_en 1'b1: bypass traffics with priority higher than priority_level_th 1'b0: not bypass
1	RW	0x0	cpu_bypass_en 1'b1: bypass cpu traffics 1'b0: not bypass
0	RW	0x0	global_en msch_ready_ctrl global enable 1'b1: enable 1'b0: disable note: msch_ready_ctrl only works when grf_ddrbuf_en in DDR_CON0 is set to 0

DDR_GRF_CPU_IDLE_TH
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	cpu_idle_threshold Only when there is not any cpu traffics after cpu_idle_threshold memory scheduler clock cycles, the msch_ready_ctrl can drive ready low. Only used when cpu_bypass_en is 1'b1

DDR_GRF_READY_LOW_CYCLES
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	ready_low_cycles The total memory scheduler clock cycles to keep ready low

DDR_GRF_READY_HIGH_CYCLES
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	ready_high_cycles The total memory scheduler clock cycles to keep ready high

DDR_GRF_PRIORITY_IDLE_TH
Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	priority_idle_threshold Only when there is not any traffics with priority higher than PRIORITY_LEVEL_TH after priority_idle_threshold memory scheduler clock cycles, the msch_ready_ctrl can drive ready low. Only used when priority_bypass_en is 1'b1

DDR_GRF_PRIORITY_LEVLE_TH
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable
15:0	RW	0x0000	priority_level_threshold When priority is higher than this value, the traffics will be bypassed

DDR_GRF_STATUS0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[31:0] DDR_STATUS0~DDR_STATUS7 are Mode Register Read Data. mrr_data0[31:0] data status. (LPDDR2/3/4): Mode register read data. (DDR4): Multi-purpose register (MPR) read data. Valid when hif_mrr_data_valid is high. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 or DDR4 For DDR4, the width of this signal is equal to the width of the dfi_rddata signal. DDR4 MPR read data received on the DFI interface can be read on hif_mrr_data when hif_mrr_data_valid is asserted

DDR_GRF_STATUS1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[63:32] mrr_data0[63:32] data status. See DDR_STATUS0

DDR_GRF_STATUS2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[95:64] mrr_data0[95:64] data status. See DDR_STATUS0

DDR_GRF_STATUS3
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[127:96] mrr_data0[127:96] data status. See DDR_STATUS0

DDR_GRF_STATUS4
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[31:0] mrr_data1[31:0] data status. See DDR_STATUS0

DDR_GRF_STATUS5
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[63:32] mrr_data1[63:32] data status. See DDR_STATUS0

DDR_GRF_STATUS6
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[95:64] mrr_data1[95:64] data status. See DDR_STATUS0

DDR_GRF_STATUS7
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[127:96] mrr_data1[127:96] data status. See DDR_STATUS0

DDR_GRF_STATUS8
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RO	0x0	cpu_port_probe_mainStatAlarm cpu_port_probe_mainStatAlarm
17	RO	0x0	cpu_port_probe_mainTraceAlarm cpu_port_probe_mainTraceAlarm
16	RO	0x0	gpu_port_probe_mainStatAlarm gpu_port_probe_mainStatAlarm
15	RO	0x0	gpu_port_probe_mainTraceAlarm gpu_port_probe_mainTraceAlarm
14	RO	0x0	mmip_port_probe_mainStatAlarm mmip_port_probe_mainStatAlarm
13	RO	0x0	mmip_port_probe_mainTraceAlarm mmip_port_probe_mainTraceAlarm
12	RO	0x0	peri_port_probe_mainStatAlarm peri_port_probe_mainStatAlarm
11	RW	0x0	peri_port_probe_mainTraceAlarm peri_port_probe_mainTraceAlarm
10	RW	0x0	dfi_scramble_read_of dfi_scramble_read_of
9	RW	0x0	dfi_scramble_write_of dfi_scramble_write_of
8	RW	0x0	dfi_scramble_key_ready dfi_scramble_key_ready
7:6	RW	0x0	grf_st_ddrc_reg_selfref_type grf_st_ddrc_reg_selfref_type
5	RW	0x0	grf_st_cactive_aclk grf_st_cactive_aclk
4	RW	0x0	grf_st_csysaclk_aclk grf_st_csysaclk_aclk

Bit	Attr	Reset Value	Description
3	RO	0x0	grf_con_csysreq_aclk grf_con_csysreq_aclk
2	RO	0x0	cactive_ddrc external cactive_ddrc
1	RO	0x0	csysack_ddrc external csysack_ddrc
0	RO	0x0	csysreq_ddrc external csysreq_ddrc

DDR_GRF_STATUS9
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31	RO	0x0	dfi_lp_ck_disable status low power of ddr phy
30	RO	0x0	reserved
29:24	RO	0x00	grf_st_hif_refresh_req_bank grf_st_hif_refresh_req_bank
23	RO	0x0	reserved
22:16	RO	0x00	grf_st_wr_credit_cnt grf_st_wr_credit_cnt
15	RO	0x0	reserved
14:8	RO	0x00	grf_st_hpr_credit_cnt grf_st_hpr_credit_cnt
7	RO	0x0	reserved
6:0	RO	0x00	grf_st_lpr_credit_cnt grf_st_lpr_credit_cnt

Chapter 4 Graphics Process Unit (GPU)

4.1 Overview

The GPU is a hardware accelerator for 2D and 3D graphics systems.
The GPU supports these compute API standards:

- OpenCL 2.0 Full Profile.

The GPU supports these graphics API standards:

- DirectX 11 FL9_3.
- OpenGLES 1.1, 2.0, and 3.2.
- Vulkan 1.0.

The GPU consists of:

- Job manager
- One double-pixel shader core, with one execution engine.
- Hierarchical tiler.
- Memory management unit.
- A single 64k L2 cache slice.

The GPU contains 1128-bit AXI slave bus and 1128-bit AXI master bus. CPU configures GPU through AXIslave bus, GPU read and write data through AXI master bus.

4.2 Block Diagram

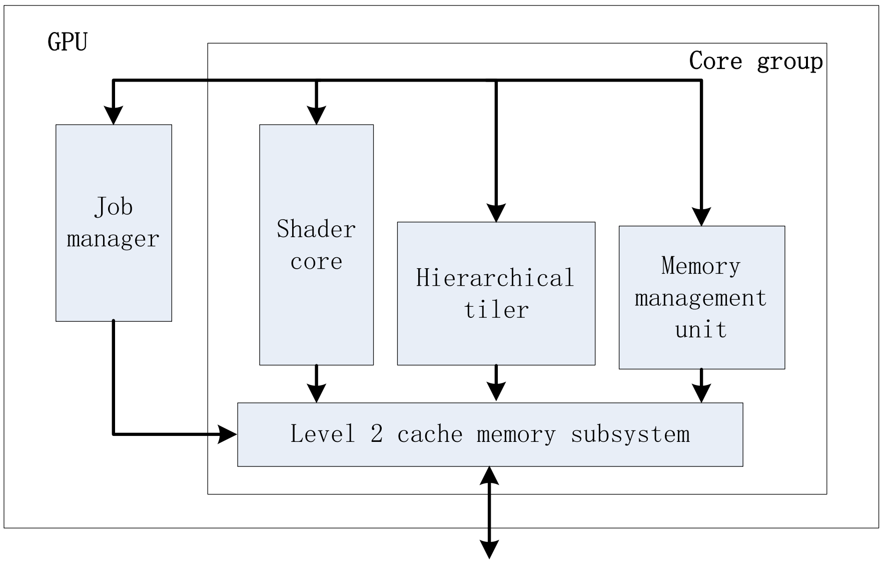


Fig. 4-1GPU block diagram

4.3 Register Description

The GPU base addressis 0XFF40_0000. Please refer to the document of "*ARM_mali??_r0p0_00eac0_TechnicalReferenceManual.pdf*"for the detailed register description

Chapter 5 Cortex-A35

5.1 Overview

The PX30 has a quad-core Cortex-A35 cluster with 256K L2 memory. Cortex-A35 processor, which is a mid-range, low-power processor that implements the ARMv8-A architecture.
The Cortex-A35 processor includes following features:

- Full implementation of the ARMv8-A A64, A32, and T32 instruction sets.
- Both the AArch32 and AArch64 execution states at all Exception levels (EL0 to EL3).
- In-order pipeline with direct and indirect branch prediction.
- Separate Level 1 (L1) data and instruction side memory systems with a Memory Management Unit(MMU).
- Level 2 (L2) memory system that provides cluster memory coherency.
- L2 cache.
- TrustZone.
- Support data engine that implements the Advanced SIMD and floating-point architecture support.
- Support Cryptographic Extension.
- ARMv8 debug logic.
- Support Generic Interrupt Controller (GIC) CPU interface to connect to an external distributor.
- Generic Timers supporting 64-bit count input from an external system counter.

The configuration details are shown in following tables

Table 5-1 CPU Configuration

Number of CPU	4
L1 I cache size	32K
L1 D cache size	32K
L2 cache size	256K
L2 data RAM output latency	3 cycles
L2 data RAM input latency	2 cycles
CPU cache protection	No
SCU L2 cache protection	No
BUS master interface	AXI4
NEON and floating point support	Yes
Cryptography extension	Yes

5.2 Block Diagram

The Cortex-A35 sub system is shown in Figure 1-1. As illustrated, quad-core Cortex-A35 connects to system bus through SCU-L2 which can handle with CDC(clock domain crossing) issue.
The Cortex-A35 is connected with system counter, which can run under a constant frequency clock, for PPI interrupt generation.

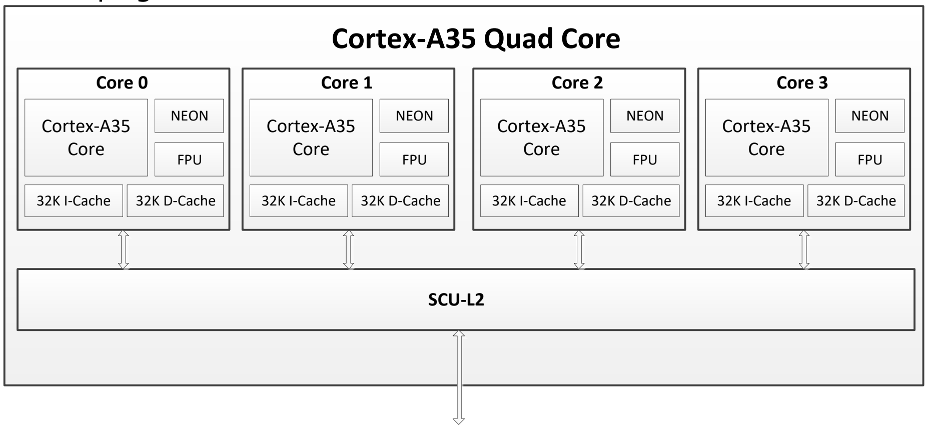


Fig. 5-1 Block Diagram

5.3 Function Description

Please refer to the document cortex_a35_r0p2_trm.pdf for the detail function description.

Chapter 6 Embedded SRAM

6.1 Overview

There are two embedded SRAMs, SYSTEM_SRAM and PMU_SRAM. the AXI slave device, which supports read and write access to provide system fast access data storage

6.1.1 Features supported

- SYSTEM_SRAM
  - Provide 16KB access space
  - Support security and non-security access
  - Security or non-security space is software programmable
  - Security space is nx4KB(up to whole memory space)

- PMU_SRAM
  - Provide 8KB access space
  - Support security access only

6.2 Block Diagram

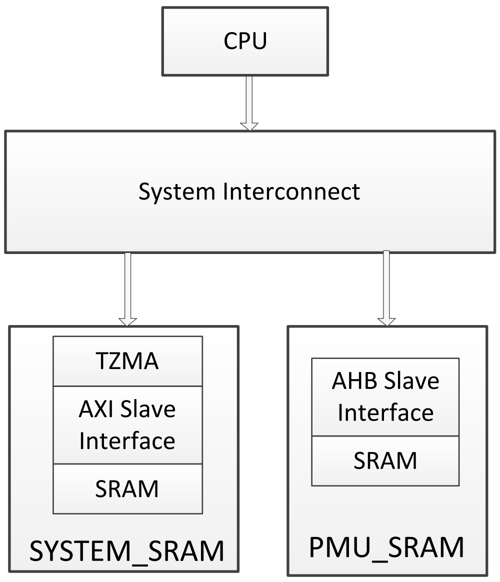


Fig. 6-1 Embedded SRAM block diagram

6.3 Function Description

6.3.1 AXI slave interface of SYSTEM_SRAM

The AXI slave interface is bridge which translate AXI bus access to SRAM interface of SYSTEM_SRAM.

6.3.2 AXI slave interface of PMU_SRAM

The AHB slave interface is bridge which translate AHB bus access to SRAM interface of PMU_SRAM.

6.3.3 Embedded SRAM access path

The SYSTEM_SRAM can only be accessed by CPU, DMAC_BUS, CRYPTO and NANDC. The PMU_SRAM can only accessed by CPU.

Chapter 7 Nand Flash Controller (NandC)

7.1 Overview

Nand Flash Controller (NandC) is used to control data transmission from host to flash device or from flash device to host. NandC is connected to AHB BUS through an AHB Master and an AHB Slave. The data transmission between host and external memory can be done through AHB Master interface or AHB Slave interface.
NandC supports the following features:

- Software Interface Type
  - Support directly mode
  - Support LLP(Linked List Pointer) mode
- Flash Interface Type
  - Support Asynchronous Flash Interface with 8bits datawidth (“Asyn8x” for short)
  - Support ONFI Synchronous Flash Interface (“ONFI Syn” for short)
  - Support Toggle Flash Interface (“Toggle” for short)
  - Support 2 flash devices at most
- Flash Type
  - Support Managed NAND Flash(LBA) and Raw NAND Flash(NO-LBA)
  - Support SLC/MLC/TLC Flash
- Flash Interface Timing
  - Asyn8x: configurable timing, one byte per two host clocks at the fastest speed
  - ONFI Syn: configurable timing, two bytes per two host clocks at the fastest speed
  - Toggle: configurable timing, two byte per two host clocks at the fastest speed
- Randomizer Ability
  - Supporttwo randomizer mode with different polynomial
  - Support two randomizer width, 8bit and 16bit parallel
- BCH/ECC Ability
  - 24bit/1KB BCH/ECC: support 24 bit BCH/ECC, which can detect and correct up to 24 error bits in every 1K bytes data
  - 40bit/1KB BCH/ECC: support 40bit BCH/ECC, which can detect and correct up to 40 error bits in every 1K bytes data
  - 60bit/1KB BCH/ECC: support 60bit BCH/ECC, which can detect and correct up to 60 error bits in every 1K bytes data
  - 70bit/1KB BCH/ECC: support 70 bit BCH/ECC, which can detect and correct up to 70 error bits in every 1K bytes data
  - 24bit/512B BCH/ECC: support 24 bit BCH/ECC, which can detect and correct up to 24 error bits in every 512 bytes data
  - 40bit/512B BCH/ECC: support 40bit BCH/ECC, which can detect and correct up to 40 error bits in every 512 bytes data
  - 60bit/512B BCH/ECC: support 60bit BCH/ECC, which can detect and correct up to 60 error bits in every 512 bytes data
  - 70bit/512B BCH/ECC: support 70bit BCH/ECC, which can detect and correct up to 70 error bits in every 512 bytes data
- Transmission Ability
  - Support 32K bytes data transmission at a time at most
  - Support two transfer working modes: Bypass or DMA
  - Support two transfer codewords size for Managed NAND Flash: 1024 bytes/codeword or 512 bytes/codeword
- Internal Memory
  - 2 built-in srams, and the size is 1k bytes respectively
  - Can be accessed by other masters
  - Can be operated in pingpong mode by other masters

7.2 Block Diagram

NandC comprises with:

- MIF: AHB Master Interface
- SIF : AHB Slave Interface
- SRIF : Sram Interface
- TRANSC : Transfer Controller
- LLPC : LLP Controller
- BCHENC : BCH Encoder
- BCHDEC : BCH Decoder
- RANDMZ : Randomizer
- FIF_GEN : Flash Interface Generation
- DLC : Delay Line Controller
- NAND_IO : Flash IO Interface

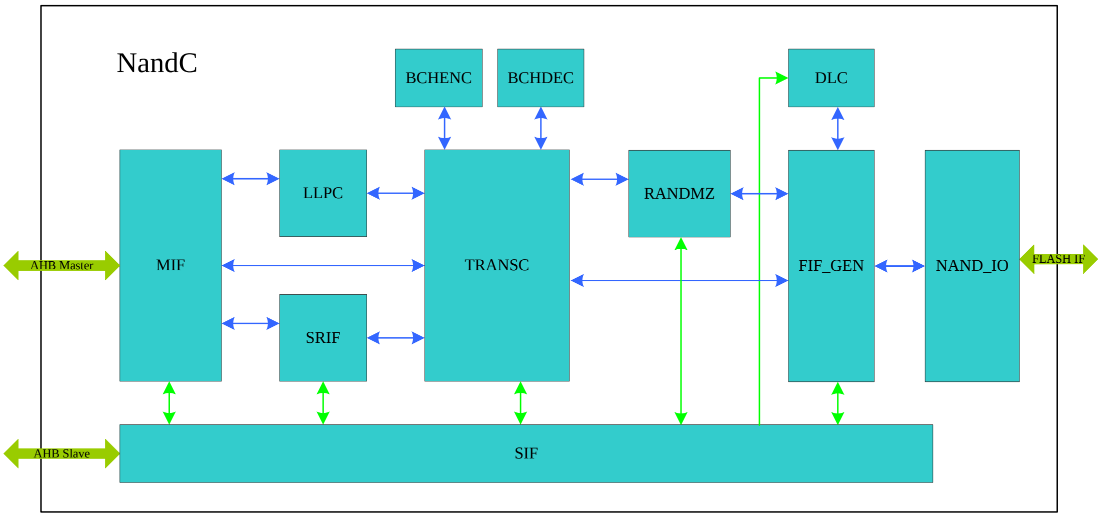


Fig.7-1NandC Block Diagram

7.3 Function Description

7.3.1 AHB Interface

There is an AHB master interface in NandC, which is selectable and configurable. It is responsible for transferring data from external memory to internal memory when flash program, or inverse when flash read; and transferring LLP data from external memory to internal register file when LLP is active.
There is an AHB slave interface in NandC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in 1.4.1.

7.3.2 Flash Type/Flash Interface

Flash device with different types of interfaces is supported. These interfaces include: asynchronous 8bits flash interface, ONFI synchronous flash interface, toggle flash interface, and so on. You can select one of them by software (configure FMCTL) to suit for these devices. Also you can configure their timing parameters by software (configure FMWAIT_ASYN/FMWAIT_SYN) to have your desired rate.

7.3.3 Linked List Pointer Mode (LLP)

To save the software resource and improve the performance, a LLP is add, which is selectable. When LLP is selected, the flash operation instructions stored in external memory with specific format should be loaded for flash working. The detailed format and working flow are referred to 15.7.8.

7.3.4 BCH Encoder/BCH Decoder

The BCH Encoder is responsible for encoding data to be written into flash device. The max encoded length is 1152bytes, in which the data length is 1024bytes, system information is 4bytes, BCH code is 124bytes.
The BCH Decoder is responsible for decoding data read from flash device. The max decoded length is 1152bytes, in which the data length is 1024bytes, spare length is 128bytes.

7.3.5 Randomizer

To improve device lifetime, a randomizer is added in NandC. It includes two parts: Scrambler and Descrambler, which is responsible for scrambling data to be written into flash after bch encoding, and descramblingdata read from flash before bch decoding.

7.3.6 Delay Line Controller

For ONFI Synchronous Flash or Toggle Flash, the data read from flash follows with a strobe signal: DQS, where a skew between them exists. To remove the skew and improve the timing between data and DQS, a Delay Line Controller is needed. It is responsible for detecting the phase of the signal similar to DQS, determining the element number to be shifted, and then shifting the DQS with the determined number.

7.3.7 NAND_IO

Different Interface signals such as asynchronous, onfi and toggle interface are multiplexedand some related logic are included.

7.4 Register Description

7.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

Name	Offset	Size	Reset Value	Description
NANDC_FMCTL	0x0000	W	0x00000a00	Flash Interface Control Register
NANDC_FMWAIT_ASYN	0x0004	W	0x3f3ff7ff	Flash Timing Control Register For Asynchronous Timing
NANDC_FMWAIT_SYN	0x0008	W	0x00000000	Flash Timing Control Register For Synchronous Timing
NANDC_FLCTL	0x0010	W	0x00100000	Internal Transfer Control Register
NANDC_FIFO_ACCESS	0x0014	W	0x00000000	FIFO access Register
NANDC_BCHCTL	0x0020	W	0x00000008	BCH Control Register
NANDC_MTRANS_CFG	0x0030	W	0x000001d0	Bus Transfer Configuration Register
NANDC_MTRANS_SADDR0	0x0034	W	0x00000000	Start Address Register For Page Data Transmission
NANDC_MTRANS_SADDR1	0x0038	W	0x00000000	Start Address Register For Spare Data Transmission
NANDC_MTRANS_STAT	0x0040	W	0x00000000	Bus Transfer Status Register
NANDC_MTRANS_STAT2	0x0044	W	0x00000000	Bus Transfer Status Register2
NANDC_DLL_CTL_REG0	0x0050	W	0x007f7f05	DLL Control Register 0
NANDC_DLL_CTL_REG1	0x0054	W	0x00000022	DLL Control Register 1
NANDC_DLL_OBS_REG0	0x0058	W	0x00000200	DLL Status Register
NANDC_NANDC_VER	0x0080	W	0x56393030	Nandc Version Register
NANDC_LLP_CTL	0x0090	W	0x00000000	LLP Control Register
NANDC_LLP_STAT	0x0094	W	0x00000001	LLP Status Register
NANDC_LLI_FOP7	0x00a0	W	0x00000000	LLI flash operation byte 7;
NANDC_LLI_FOP8	0x00a4	W	0x00000000	LLI flash operation byte 8;
NANDC_LLI_FOP9	0x00a8	W	0x00000000	LLI flash operation byte 9;
NANDC_LLI_FOP10	0x00ac	W	0x00000000	LLI flash operation byte 10;
NANDC_LLI_FOP11	0x00b0	W	0x00000000	LLI flash operation byte 11;
NANDC_LLI_FOP12	0x00b4	W	0x00000000	LLI flash operation byte 12;
NANDC_LLI_FOP13	0x00b8	W	0x00000000	LLI flash operation byte 13;
NANDC_LLI_FOP14	0x00bc	W	0x00000000	LLI flash operation byte 14;
NANDC_LLI_NXT_LLP	0x00c0	W	0x00000000	Next LLI
NANDC_LLI_FOP0	0x00c4	W	0x00000000	LLI flash operation byte 0;
NANDC_LLI_FOP1	0x00c8	W	0x00000000	LLI flash operation byte 1;
NANDC_LLI_FOP2	0x00cc	W	0x00000000	LLI flash operation byte 2;
NANDC_LLI_FOP3	0x00d0	W	0x00000000	LLI flash operation byte 3;
NANDC_LLI_FOP4	0x00d4	W	0x00000000	LLI flash operation byte 4;
NANDC_LLI_FOP5	0x00d8	W	0x00000000	LLI flash operation byte 5;
NANDC_LLI_FOP6	0x00dc	W	0x00000000	LLI flash operation byte 6;

Name	Offset	Size	Reset Value	Description
NANDC_INTEN	0x0120	W	0x00000000	NandC Interrupt Enable Register
NANDC_INTCLR	0x0124	W	0x00000000	NandC Interrupt Clear Register
NANDC_INTST	0x0128	W	0x00000000	NandC Interrupt Status Register
NANDC_BCHST0	0x0150	W	0x80000000	BCH Status Register For Codeword 0~1
NANDC_BCHST1	0x0154	W	0x00000000	BCH Status Register For Codeword 2~3
NANDC_BCHST2	0x0158	W	0x00000000	BCH Status Register For Codeword 4~5
NANDC_BCHST3	0x015c	W	0x00000000	BCH Status Register For Codeword 6~7
NANDC_BCHST4	0x0160	W	0x00000000	BCH Status Register For Codeword 8~9
NANDC_BCHST5	0x0164	W	0x00000000	BCH Status Register For Codeword 10~11
NANDC_BCHST6	0x0168	W	0x00000000	BCH Status Register For Codeword 12~13
NANDC_BCHST7	0x016c	W	0x00000000	BCH Status Register For Codeword 14~15
NANDC_BCHST8	0x0170	W	0x00000000	BCH Status Register For Codeword 16~17
NANDC_BCHST9	0x0174	W	0x00000000	BCH Status Register For Codeword 18~19
NANDC_BCHST10	0x0178	W	0x00000000	BCH Status Register For Codeword 20~21
NANDC_BCHST11	0x017c	W	0x00000000	BCH Status Register For Codeword 22~23
NANDC_BCHST12	0x0180	W	0x00000000	BCH Status Register For Codeword 24~25
NANDC_BCHST13	0x0184	W	0x00000000	BCH Status Register For Codeword 26~27
NANDC_BCHST14	0x0188	W	0x00000000	BCH Status Register For Codeword 28~29
NANDC_BCHST15	0x018c	W	0x00000000	BCH Status Register For Codeword 30~31
NANDC_SPARE0_0	0x0200	W	0xffffffff	System Information for codeword 0
NANDC_SPARE1_0	0x0204	W	0xffffffff	System Information for codeword 1
NANDC_RANDMZ_CFG	0x0208	W	0x00000000	Randomizer Configure Register
NANDC_SEED_BCHST	0x020c	W	0x00000000	Bchst Seed

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

7.4.2 Detail Register Description

NANDC_FMCTL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

23	RW	0x0	Data_mux_sel
			Used to select nandc pin function;
			0: dq0~7 pin used as "DQ[0]~[7]" function;
			1: dq0~7 pin used as "DQ[7]~[0]" function;

22	RW	0x0	Cmd_mux_sel Used to select nandc pin function; 0: We pin used as "WE" function; Ale pin used as "ALE" function; Cle pin used as "CLE" function; 1: We pin used as "ALE" function; Ale pin used as "WE" function; Cle pin used as "WP" function;
21	RW	0x0	Diff_mux_sel Used to select nandc pin function; 0: rdn pin used as "RE" function; dqs pin used as "DQS" function; 1: rdn pin used as "DQS" function dqs pin used as "RE" function;
20:18	RW	0x0	sif_read_delay Used to control the delay time when asynchronous mode
17	RO	0x0	flash_abort_stat Function1: flash_abort_stat, RO. Function2: flash_abort_clear, RW, auto clear. flash_abort_stat is set to 1 when flash abort if flash_abort_en=1, set to 0 when flash_abort_clear=1.
16	RW	0x0	flash_abort_en Flash abort protect enable signal, 1 active. 0: Flash abort protect disable. 1: Flash abort protect enable. Notes: 1. when in dma mode, if the time from last read operation start to the last read valid exceeds 1024 cycles, flash_abort_stat is set to high. 2. when in bypass mode, if the time from current read operation start to the read valid exceed 1024 cycles, flash_abort_stat is set to high. 3. when in llp bypass read/read match mode, when the operation is long than 1024 cycles, flash_abort_stat is set to high.

Bit	Attr	Reset Value	Description
15	RW	0x0	syn_mode Toggle enable signal, 1 active. 0: ONFI synchronous flash. 1: Toggle synchronous flash.
14	RW	0x0	syn_clken Synchronous flash clock enable signal, 1 active. Only available in Synchronous Mode. 0: flash clock is disabled. 1: flash clock is enabled.
13	RW	0x0	tm Timing mode indication. 0: Asynchronous Mode. 1: Synchronous Mode (Toggle or ONFI Synchronous).
12	RO	0x0	reserved
11	RO	0x1	Dma_f_flag Indication for the all f byte in the current DMA transmission. 0: the transmission is not all f 1: the transmission is all f
10	RO	0x0	fifo_empty fifo empty signal. 1'b0: fifo is not empty; 1'b1: fifo is emtpy;
9	RO	0x1	frdy Flash ready/busy indicate signal. 0: flash is busy. 1: flash is ready. This bit is the sample of the pin of R/Bn.
8	RW	0x0	reserved
7	RW	0x0	fcs7 Flash memory chip 7 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
6	RW	0x0	fcs6 Flash memory chip 6 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
5	RW	0x0	fcs5 Flash memory chip 5 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
4	RW	0x0	fcs4 Flash memory chip 4 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.

Bit	Attr	Reset Value	Description
3	RW	0x0	fcs3 Flash memory chip 3 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
2	RW	0x0	fcs2 Flash memory chip 2 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
1	RW	0x0	fcs1 Flash memory chip 1 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.
0	RW	0x0	fcs0 Flash memory chip 0 select control. 1: hold flash memory chip select activity. 0: flash memory chip select activity free.

NANDC_FMWAIT_ASYN
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	fmw_dly_en fmw_dly enable signal,1 active.
29:24	RW	0x3f	fmw_dly The number of delay cycle between two codeword transmission.
23	RO	0x0	reserved
22:18	RW	0x0f	wait_frdy_dly The number of delay cycle to accept the flash ready signal.
17:12	RW	0x3f	csrw When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the falling edge of CSn to the falling edge of RDn or WRn. The min value of csrw is 0.
11	RW	0x0	hard_rdy Hardware handshaking controller bit. When asserted, an external device asserts signal "RDY" to extend a wait-state access and the rest bits in this register will be ignored.
10:5	RW	0x3f	rwpw When in Asynchronous mode or Toggle address/command mode, this field specifies the width of RDn or WRn in processor clock cycles, 0x0<=rwpw<=0x3f.

Bit	Attr	Reset Value	Description
4:0	RW	0x1f	rwcs When in Asynchronous mode or Toggle address/command mode, this field specifies the number of processor clock cycles from the rising edge of RDn or WRn to the rising edge of CSn, 0x0<=rwcs<=0x1f.

NANDC_FMWAIT_SYN
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	ssyn_xle_sel ALE/CLE selection signal for ONFI synchronous flash: 0: ALE/CLE aligned to the falling edge of WRN 1: ALE/CLE aligned to the center of WRN low level
14:9	RW	0x00	pst Write/Read Postamble time for ONFI synchronous mode or Toggle data mode. This field specifies the number of processor clock cycle for Postamb- le time.
8:3	RW	0x00	pre Write/Read Preamble time for ONFI synchronous mode or Toggle data mode. This field specifies the number of processor clock cycle for preamble time.
2:0	RW	0x0	fclk Half hclk cycle number for flash clock for ONFI synchronous mode or Toggle data mode

NANDC_FLCTL
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

30	RW	0x0	bypass_fifo_mode
			The enable sigal for bypass with fifo mode.
			1'b0: disable;
			1'b1: enable;

29	RW	0x0	async_tog_mix Nandc async mode and tog mode compatible control 0: async write data can't be read by tog read 1: async write data can be read by tog read
28	RW	0x0	low_power Nandc low power control 0: normal mode 1: low power mode

Bit	Attr	Reset Value	Description
27:22	RW	0x00	page_num Transmission codeword number in internal DMA mode when bus-mode is master-mode 1~32: 1~32 codeword. default: not support. Notes: a. Only active in internal DMA mode b. Only active when bus-mode is master-mode
21	RW	0x0	page_size Transmission codeword size in internal DMA mode 0: 1024bytes/codeword 1: 512bytes/codeword Note: only used when lba_en=1;
20	RO	0x1	tr_rdy Internal DMA transmission ready indication. 0: internal DMA transmission is busy 1: internal DMA transmission is ready When reading flash, tr_rdy should not be set to 1 until all data transmission and correct finished. When programing flash, tr_rdy should not be set to 1 until all data transmission finished. Notes: Only active in internal DMA mode.
19	RW	0x0	bchst_trans Transmission the status of BCH to external memory 0: not transmission 1: transmission
18:13	RO	0x0	reserved

12	RW	0x0	lba_spare_sel
			Spare byte number selector when lba_en=1.
			0: spare size is 0;
			1: spare size is 4bytes;

Bit	Attr	Reset Value	Description
11	RW	0x0	lba_en LBA mode indication, 1 active. 0: NO-LBA mode, NandC should transfer both page data and spare data in every codeword, and the page size is 1024 bytes or 512 bytes determined by BCHCTL[16](bchpage), spare size is 46/74/109 bytes or 127 bytes determined by BCHCTL[27:25]. 1: LBA mode, NandC should transfer both page data and spare data in every codeword, and the page size is 1024 bytes or 512 bytes determined by FLCTL[21](page_size), spare size is determined by FLCTL[12](lba_spare_sel). Notes: a. When lba_en is active, BCH CODEC should be disabled, spare_size and page_size are configurable. b. When lba_en is active, cor_able is inactive.
10	RW	0x0	cor_able Auto correct enable indication, 1 active. 0: auto correct disable 1: auto correct enable Notes: a. Only active in internal DMA mode. b. lba_en is prior to cor_able. When lba_en=1, cor_able is ignored.
9	RW	0x0	trans_seed Transfer the randomizer seed to flash 0: not transfer the seed to flash. 1: transfer the seed to flash.
8	RW	0x0	not_trans_data Not Transfer the data 0: transfer the data with spare. 1: Not transfer the data.
7	RW	0x0	flash_st_mod Mode for NandC to start internal data transmission in internal DMA mode. 0: busy mode: hardware should not start internal data transmission until flash is ready even flash_st is asserted. 1: ready mode: hardware should start internal data transmission directly when flash_st is asserted. Notes: Only active in internal DMA mode.

Bit	Attr	Reset Value	Description
6:5	RW	0x0	tr_count Transmission codeword number in internal DMA mode when bus-mode is slave-mode 00: 0 codeword need transferred 01: 1 codeword need transferred 10: 2 codeword need transferred 11: not supported Notes: a. Only active in internal DMA mode. b. Only active when bus-mode is slave-mode.
4	RW	0x0	st_addr Start buffer address. 0: start transfer from sram0 1: start transfer from sram1 Notes: Only active in internal DMA mode.
3	RW	0x0	bypass NandC internal DMA bypass indication. 0: bypass the internal DMA, data are transferred to/from flash by direct path. 1: internal DMA active, data are transferred to/from flash by internal DMA.
2	R/W SC	0x0	flash_st Start signal for NandC to transfer data between flash and internal buffer in internal DMA mode. When asserted, it will auto cleared. 0: not start transmission 1: start transmission Notes: Only active in internal DMA mode
1	RW	0x0	flash_rdn Indicate data flow direction. 0: NandC read data from flash. 1: NandC write data to flash
0	R/W SC	0x0	flash_rst NandC software reset indication. When asserted, it will auto cleared. 0: not software reset 1: software reset Notes: flash_rst is prior to flash_st

NANDC_FIFO_ACCESS

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description

31:30	WO	0x0	Fifo_cnt
			Indicate valid data number;
			2'b00: indicate byte0~2 are invalid;
			2'b01: indicate byte0 is valid;
			2'b10: indicate byte0~1 are valid;
			2'b11: indicate byte0~2 are valid;

29:28	WO	0x0	Byte2_attr Indicate byte2 attribute; 2'b00: data 2'b01: address 2'b10: command 2'b11: data
27:26	WO	0x0	Byte1_attr Indicate byte1 attribute; 2'b00: data 2'b01: address 2'b10: command 2'b11: data
25:24	WO	0x0	byte0_attr Indicate byte0 attribute; 2'b00: data 2'b01: address 2'b10: command 2'b11: data
23:16	WO	0x00	Fifo_byte2 Byte2 of transfer data
15:8	WO	0x00	Fifo_byte1 Byte1 of transfer data
7:0	WO	0x00	fifo_byte0 Byte0 of transfer data

NANDC_BCHCTL

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RW	0x0	bchmode BCH mode selection； 000: 70bitBCH 001: 24bitBCH 010: 40bitBCH 011: 60bitBCH
24:17	RW	0x00	bchthres BCH error number threshold

Bit	Attr	Reset Value	Description
16	RW	0x0	bchpage The data size indication when BCH is active. 0: 1024 bytes, all the 1024 bytes data in codeword are valid data to be transferred. 1: 512 bytes, higher 512bytes are valid, and lower 512bytes are invalid and stuffed with 0xff. Notes: a. Only active when data transferred in internal DMA mode. b. Only active for asynchronous flash.
15:4	RO	0x0	reserved
3	RW	0x1	bchepd BCH encoder/decoder power down indication. 0: BCH encoder/decoder working. 1: BCH encoder/decoder not working.

2	RW	0x0	bch_gate_en
			Bch decoder clock gating enable, high active.
			"0": normal mode;
			"1": clock gating mode;

1	RW	0x0	wcnt_clear To clear the write counter of BCHST. When asserted, it will auto cleared. 0: not clear the counter 1: clear the counter
0	R/W SC	0x0	bchrst BCH software reset indication, When asserted, it will auto cleared. 0: not software reset 1: software reset Notes: a. BCH Decoder should be software reset before decode begin. b. bch software reset should be used with nandc software reset at the same time.

NANDC_MTRANS_CFG
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x000	redundance_size The num of all f byte to write to flash, the maximum of the size is 2K -1
15	R/W SC	0x0	ahb_rst ahb master interface software reset, auto cleared

Bit	Attr	Reset Value	Description
14	RW	0x0	fl_pwd Flash power down indication, 1 active. 0: Flash power on, data transferred through master interface is data that to be written into or read from flash. 1: Flash power down, data transferred through master interface is not data that to be written into or read from flash. NandC is just used as DMA for external memory and internal memory.
13:9	RW	0x00	incr_num AHB Master incr num indication. incr_num=1~16. When burst=001, software should configure incr_num. Notes: Only active for master-mode.
8:6	RW	0x7	burst AHB Master burst type indication: 000 : Single transfer 011 : 4-beat burst 101 : 8-beat Burst 111 : 16-beat burst default : not supported Notes: Only active for master-mode.
5:3	RW	0x2	hsize AHB Master data size indication: 000 : 8 bits 001 : 16 bits 010 : 32 bits default : not supported Notes: Only active for master-mode.
2	RW	0x0	bus_mode Bus interface selection. 0: Slave interface, flash data is transferred through slave interface 1: Master interface, flash data is transferred through master interface
1	RW	0x0	ahb_wr Data transfer direction through master interface. 0: write direction(internal memory ->external memory) 1: read direction(external memory->internal memory) Notes: a. Only active for master-mode. b. When read flash(flash_rdn=0), ahb_wr=1; when program flash(flash_rdn=1), ahb_wr=0.

Bit	Attr	Reset Value	Description
0	R/W SC	0x0	ahb_wr_st Start indication for loading data from external memory to internal memory or storing data from internal memory to external memory through master. When asserted, it will auto cleared. Notes: a. Only active for master-mode and fl_pwd=1. b. When fl_pwd=0, flash is active, NandC start to transfer data through master interface if flash_st=1 c. When fl_pwd=1, flash is not active, NandC start to transfer data through master interface if ahb_wr_st=1

NANDC_MTRANS_SADDR0
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr0 Start address for page data transmision. Notes: a. Only active for master-mode. b. Should be aligned with hsize in MTRANS_CFG[5:3].

NANDC_MTRANS_SADDR1
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	saddr1 Start address for spare data. Notes: a. Only active for master-mode. b. Should be aligned with hsize in MTRANS_CFG[5:3].

NANDC_MTRANS_STAT
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:16	RO	0x00	mtrans_cnt finished counter for codeword transmission through Master interface Notes: Only active for master-mode.

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	bus_err Bus error indication for codeword0~15. [0] : bus error for codeword 0 …… [15] : bus error for codeword 15 Notes: Only active for master-mode.

NANDC_MTRANS_STAT2
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	bus_err2 Bus error indication for codeword16~31. [0] : bus error for codeword 16 …… [15] : bus error for codeword 31 Notes: Only active for master-mode.

NANDC_DLL_CTL_REG0
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x7f	dll_dqs_dly_bypass Holds the read DQS delay setting when the DLL is operating in bypass mode.
15:8	RW	0x7f	dll_dqs_dly Holds the read DQS delay setting when the DLL is operating in normal mode. Typically, this value is 1/4 of a clock cycle. Each increment of this field represents 1/128th of a clock cycle.
7:0	RW	0x05	dll_start_point DLL Start Point Control. This value is loaded into the DLL at initialization and is the value at which the DLL will begin searching for a lock. Each increment of this field represents 1/128th of a clock cycle.

NANDC_DLL_CTL_REG1
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:4	RW	0x02	dll_incr DLL Increment Value. This sets the increment used by the DLL when searching for a lock. It is recommended keeping this field small (around 0x4) to keep the steps gradual
3:2	RW	0x0	dll_qtren Quarter flag of DLL, active in no-bypass mode. 01:1/4 fclk, dqs_dly=128. 10:1/8 fclk, dqs_dly=64. Default: dqs_dly=dll_dqs_dly(DLL_CTL_REG0[15:8]). When dll_qtr='b01 or 'b10, software not need to configure dll_dqs_dly , and hardware should delay the input signal for 1/4 or 1/8 fclk cycle time; When dll_qtr=0, software need to configure dll_dqs_dly.
1	RW	0x1	dll_bypass DLL Bypass Control, 1active 0: dll not bypass, dll_dqs_dleay= dqs_dly 1: dll bypass, dll_dqs_dleay= dll_dqs_dly_bypass
0	RW	0x0	dll_start Start signal for DLL, 1 active. Notes: It will keep high until dll disabled.

NANDC_DLL_OBS_REG0
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:9	RO	0x01	dll_dqs_delay_value Report the delay value for the read DQS signal
8:1	RO	0x00	dll_lock_value Reports the DLL encoder value from the master DLL to the slave DLL's. The slaves use this value to set up their delays for the clk_wr and read DQS signals.
0	RO	0x0	dll_lock DLL Lock indication: 0: DLL has not locked 1: DLL is locked.

NANDC_NANDC_VER
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	RO	0x56393030	version Version indication for NANDC

NANDC_LLP_CTL

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:6	RW	0x0000000	llp_loc Starting address for LLI0, 64byte align
5	RW	0x0	llp_frdy Working time for FOP_WAIT_FRDY for all FOP in first LLP group: 0: FOP_WAIT_FRDY begin working when started 1: FOP_WAIT_FRDY not begin working until 16 cycles later after started
4:3	RO	0x0	reserved
2	R/W SC	0x0	llp_rst Reset signal for LLP. When asserted, it will auto cleared.
1	RW	0x0	llp_mode 0- current LLI only has FOP 1-current LLI has both CFG and FOP
0	RW	0x0	llp_en Enable signal for LLP 0-LLP disable 1-LLP enable

NANDC_LLP_STAT
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	llp_stat latest LLI_LOC finished, 64byte align
5:2	RO	0x0	reserved
1	RO	0x0	llp_err error status for llp load or execute 0-llp is correct 1-llp is error
0	RO	0x1	llp_rdy ready status for all llp load 0-llp load is busy 1-llp load is ready

NANDC_LLI_FOP7
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP8
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation

Bit	Attr	Reset Value	Description
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP9
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;

Bit	Attr	Reset Value	Description
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP10
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP11
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP12
Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP13
Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation

Bit	Attr	Reset Value	Description
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP14
Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;

Bit	Attr	Reset Value	Description
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_NXT_LLP
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	loc Starting address for next LLI
5	RW	0x0	frdy Flash_rdy will not be used until 16 cycles after FOP_WAIT_FRDY start; "1" active
4:2	RO	0x0	reserved
1	RO	0x0	llp_mode 0: next LLI only has FOP; 1:next LLI has both FOP and CFG
0	RO	0x0	en Enable signal for next LLP

NANDC_LLI_FOP0
Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation

Bit	Attr	Reset Value	Description
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP1
Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;

Bit	Attr	Reset Value	Description
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP2
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP3
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP4
Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP5
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation

Bit	Attr	Reset Value	Description
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_LLI_FOP6
Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	Fop_type flash operation type: 000:nop operation 001:flash bypass write operation 010:flash bypass read operation 011:flash bypass read with match operation 100:flash DMA write/read operation
28	RO	0x0	Fop_matchmod when FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. When FOP_MATCHMOD=0, it is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
27:24	RO	0x0	Fop_cs Flash chip select; "1" active "1000"~"1111" indicate select cs0~cs7;

Bit	Attr	Reset Value	Description
23:20	RO	0x0	Fop_nxtid Next FOP ID;
19	RO	0x0	Fop_wait_frdy Indicate the current FOP will excute when flash ready; "1" active
18	RO	0x0	Fop_wait_trdy Indicate the current FOP will excute when DMA transfer ready; "1" active
17:16	RO	0x0	Fop_addr flash address type, FOP_ADDR[0]-ALE, FOP_ADDR[1]-CLE
15:0	RO	0x0000	Fop_inst flash write operation: indicate flash write data(command/address/data) ； flash read with match operation: indicate match pattern data

NANDC_INTEN
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	Seed_bcherr_int_en Enable for seed bch error interrupt. 0-interrupt disable 1-interrupt enable When seed bcherr_int_en is active, an interrupt is generated.
8	RW	0x0	Seed_bchfail_int_en Enable for seed bch fail interrupt. 0-interrupt disable 1-interrupt enable When seed bchfail_int_en is active, an interrupt is generated if seed bch decode failed
7	RW	0x0	rd_1stpage_int_en Enable for the first page read interrupt. 0: interrupt disable 1: interrupt enable When sif_bus_wr is active, an interrupt is generated if the first page read operation is finished
6	RW	0x0	master_idle_int_en Enable for master idle interrupt 0-interrupt disable 1-interrupt enable When master_idle_int_en is active, an interrupt is generated if posedge of master idle happen

Bit	Attr	Reset Value	Description
5	RW	0x0	flash_abort_int_en Enable for flash read abort interrupt. 0: interrupt disable 1: interrupt enable When flash_abort_int_en is active, an interrupt is generated if DQS input is abort. Available when flash interface is ONFI synchronous or toggle. When read data number is out of range of flash page size, dqs input is abort. An interrupt is generated if flash_abort_int_en is enable
4	RW	0x0	llp_int_en Enable for LLP finished interrupt. 0: interrupt disable 1: interrupt enable When llp_en_en is active, an interrupt is generated if LLP operation is finished
3	RW	0x0	bchfail_int_en Enable for bch fail interrupt. 0-interrupt disable 1-interrupt enable When bchfail_int_en is active, an interrupt is generated if bch decode failed
2	RW	0x0	bcherr_int_en Enable for bch error interrupt. 0-interrupt disable 1-interrupt enable When bcherr_int_en is active, an interrupt is generated if bch decode error bit is larger than bchthres(BCHCTL[26:19])
1	RW	0x0	frdy_int_en Enable for flash_rdy interrupt 0-interrupt disable 1-interrupt enable When frdy_int_en is active, an interrupt is generated if flash R/B# changes from 0 to 1
0	RW	0x0	dma_int_en Enable for internal DMA transfer finished interrupt 0-interrupt disable 1-interrupt enable When dma_int_en is active, an interrupt is generated if page_num(FLCTL[27:22]) of flash data transfer in DMA mode is finished

NANDC_INTCLR
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	R/W SC	0x0	Seed_bcherr_int_clr Clear for seed bch error interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
8	R/W SC	0x0	Seed_bchfail_int_clr Clear for seed bch decode fail interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
7	R/W SC	0x0	rd_1stpage_int_clr Clear for first page read interrupt. When asserted, this bit will be auto cleared. 0: interrupt not cleared 1: interrupt cleared
6	R/W SC	0x0	master_idle_int_clr Clear for master idle interrupt. When asserted, this bit will be auto cleared. 0: interrupt not cleared 1: interrupt cleared
5	R/W SC	0x0	flash_abort_int_clr Clear for flash abort interrupt. When asserted, this bit will be auto cleared. 0: interrupt not cleared 1: interrupt cleared Available when flash interface is ONFI synchronous or toggle
4	R/W SC	0x0	llp_int_clr Clear for LLP finished interrupt. When asserted, this bit will be auto cleared. 0: interrupt not cleared 1: interrupt cleared
3	R/W SC	0x0	bchfail_int_clr Clear for bch decode fail interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
2	R/W SC	0x0	bcherr_int_clr Clear for bch error interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared

Bit	Attr	Reset Value	Description
1	R/W SC	0x0	frdy_int_clr Clear for flash_rdy interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared
0	R/W SC	0x0	dma_int_clr Clear for internal DMA transfer finished interrupt. When asserted, this bit will be auto cleared. 0-interrupt cleared 1-interrupt not cleared

NANDC_INTST
Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	Seed_bcherr_int_stat Staus for seed bch decode error interrupt, high active
8	RO	0x0	Seed_bchfail_int_stat Staus for seed bch decode fail interrupt, high active
7	RO	0x0	rd_1stpage_int_stat Status for first page read interrupt, high active
6	RO	0x0	master_idle_int_stat Status for master idle interrupt, high active
5	RO	0x0	flash_abort_int_stat Status for flash abort, high active Available when flash interface is ONFI synchronous or toggle
4	RO	0x0	llp_int_stat Status for LLP finished interrupt, high active
3	RO	0x0	bchfail_int_stat Status for bch decode fail interrupt, high active
2	RO	0x0	bcherr_int_stat Status for bch error interrupt, high active
1	RO	0x0	frdy_int_stat Status for flash_rdy interrupt, high active
0	RO	0x0	dma_int_stat Status for internal DMA transfer finished interrupt, high active

NANDC_BCHST0
Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31	RO	0x1	bchst_bchrdy Ready indication for bch encoder/decoder, 1 active. 0: bch encoder/decoder is busy 1: bch encoder/decoder is ready

Bit	Attr	Reset Value	Description
30	RC	0x0	decode_done_rdy Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
29:27	RO	0x0	reserved
26	RO	0x0	all_f_flag1 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum1 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail1 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done1 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf1 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag0 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum0 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail0 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done0 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf0 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST1
Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RO	0x0	all_f_flag3 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum3 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail3 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done3 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf3 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag2 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum2 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail2 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done2 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf2 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST2
Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag5 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum5 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail5 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done5 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf5 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag4 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum4 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail4 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done4 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf4 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST3
Address: Operational Base + offset (0x015c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag7 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum7 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail7 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done7 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf7 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag6 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum6 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail6 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done6 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf0 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST4
Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag9 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum9 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail9 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done9 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf9 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag8 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum8 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail8 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done8 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf8 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST5

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag11 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum11 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail11 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done11 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf11 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag10 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum10 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail10 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done10 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf10 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST6
Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag13 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum13 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail13 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done13 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf13 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag12 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum12 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail12 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done12 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf12 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST7
Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag15 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum15 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail15 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done15 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf15 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag14 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum14 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail14 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done14 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf14 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST8
Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag17 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum17 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail17 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done17 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf17 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag16 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum16 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail16 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done16 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf16 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST9
Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag19 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum19 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail19 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done19 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf19 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag18 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum18 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail18 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done18 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf18 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST10
Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag21 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum21 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail21 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done21 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf21 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag20 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum20 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail20 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done20 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf20 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST11
Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag23 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum23 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail23 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done23 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf23 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag22 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum22 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail22 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done22 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf22 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST12
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag25 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum25 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail25 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done25 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf25 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag24 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum24 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail24 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done24 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf24 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST13
Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag27 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum27 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail27 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done27 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf27 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag26 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum26 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail26 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done26 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf26 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST14
Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag29 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum29 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail29 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done29 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf29 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag28 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum28 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail28 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done28 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf28 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_BCHST15

Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26	RO	0x0	all_f_flag31 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
25:19	RO	0x00	err_tnum31 Indication for the number of error in current backup codeword
18	RO	0x0	decode_fail31 Indication for the 1st backup codeword decoded failed or not. 0: decode successfully 1: decode fail
17	RO	0x0	decode_done31 Indication for finishing decoding the 1st backup codeword 0: not finished 1: finished
16	RO	0x0	errf31 Indication for error found in 1st backup codeword. 0: no error 1: error found
15:11	RO	0x0	reserved
10	RO	0x0	all_f_flag30 Indication for the all f byte in the current codeword. 0: the current codeword is not all f 1: the current codeword is all f
9:3	RO	0x00	err_tnum30 Indication for the number of error in current backup codeword
2	RO	0x0	decode_fail30 Indication for current backup codeword decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	decode_done30 Indication for finishing decoding the current backup codeword. 0: not finished 1: finished
0	RO	0x0	errf30 Indication for error found in current backup codeword. 0: no error 1: error found

NANDC_SPARE0_0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 0
23:16	RW	0xff	system_2 the 3rd system byte of codeword 0
15:8	RW	0xff	system_1 the 2nd system byte of codeword 0
7:0	RW	0xff	system_0 the 1st system byte of codeword 0

NANDC_SPARE1_0
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 1
23:16	RW	0xff	system_2 the 3rd system byte of codeword 1
15:8	RW	0xff	system_1 the 2nd system byte of codeword 1
7:0	RW	0xff	system_0 the 1st system byte of codeword 1

NANDC_RANDMZ_CFG
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31	RW	0x0	randmz_en Randomizer enable indication, 1 active. 0: Randomizer not active 1: Randomizer active Notes: a. Not active when data transmission in bypass mode. b. Just active for data, but not for address and command. c. Not active when BchPage=1.
30:29	RW	0x0	randmz_mode Randomizer mode: 00- Samsung randomizer Polynomial=1+x+x^15 10- Samsung randomizer Polynomial=1+x^14+x^15
28:20	RO	0x0	reserved
19:0	RW	0x00000	randmz_seed when Samsung randomizer: The seed for randomizer(initial value); when Toshiba randomizer: Seed Agitation Register.

NANDC_SEED_BCHST

Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	Seed_bchst_rdy Indication for randmz seed bchst is ready or not 0: bchst is not ready 1: bchst is ready
4:3	RW	0x0	Seed_err_tnum Indication for the number of error in randmz seed
2	RO	0x0	Seed_decode_fail Indication for randmz seed decode failed or not 0: decode successfully 1: decode fail
1	RO	0x0	Seed_decode_done Indication for finishing decoding the randmz seed. 0: not finished 1: finished
0	RO	0x0	seed_errf Indication for error found in randmz seed. 0: no error 1: error found

7.5 Interface Description

Table 7-1NandC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
flash_ale	O	IO_FLASHale_EMMCrstn_GPIO1B3vccio0	GRF_GPIO1B_IOMUX_L[14:12]=3'b001
flash_cle	O	IO_FLASHcle_UART3ctsm1_SPI0mosi_I2C3sda_GPIO1B4vccio0	GRF_GPIO1B_IOMUX_H [2:0]= 3'b001
flash_wrn	O	IO_FLASHwrn_UART3rtsm1_SPI0miso_I2C3scl_GPIO1B5vccio0	GRF_GPIO1B_IOMUX_H [6:4]= 3'b001
flash_rdn	O	IO_FLASHrdn_UART3rxm1_SPI0clk_GPIO1B7vccio0	GRF_GPIO1B_IOMUX_H [14:12]=3'b001
flash_data[0]	I/O	IO_FLASHd0_EMMCd0_SFCsio0_GPIO1A0vccio0	GRF_GPIO1A_IOMUX_L [2:0]= 3'b001
flash_data[1]	I/O	IO_FLASHd1_EMMCd1_SFCsio1_GPIO1A1vccio0	GRF_GPIO1A_IOMUX_L [6:4]= 3'b001
flash_data[2]	I/O	IO_FLASHd2_EMMCd2_SFCsio2_GPIO1A2vccio0	GRF_GPIO1A_IOMUX_L [10:8]= 3'b001
flash_data[3]	I/O	IO_FLASHd3_EMMCd3_SFCsio3_GPIO1A3vccio0	GRF_GPIO1A_IOMUX_L [14:12]=3'b001
flash_data[4]	I/O	IO_FLASHd4_EMMCd4_SFCcsn0_GPIO1A4vccio0	GRF_GPIO1A_IOMUX_H [2:0]= 3'b001
flash_data[5]	I/O	IO_FLASHd5_EMMCd5_GPIO1A5vccio0	GRF_GPIO1A_IOMUX_H [6:4]= 3'b001
flash_data[6]	I/O	IO_FLASHd6_EMMCd6_GPIO1A6vccio0	GRF_GPIO1A_IOMUX_H [10:8]= 3'b001
flash_data[7]	I/O	IO_FLASHd7_EMMCd7_GPIO1A7vccio0	GRF_GPIO1A_IOMUX_H [14:12]=3'b001
flash_dqs	I/O	IO_FLASHdqs_EMMCcmd_GPIO1B2vccio0	GRF_GPIO1B_IOMUX_L [10:8]= 3'b001
flash_rdy	I	IO_FLASHrdy_EMMCclkout_SFCclk_GPIO1B1vccio0	GRF_GPIO1B_IOMUX_L [6:4]= 3'b001

flash_csn0	O	IO_FLASHcs0_EMMCpwren_GPIO1B0vccio0	GRF_GPIO1B_IOMUX_L [2:0]=1
flash_csn1	O	IO_FLASHcs1_UART3txm1_SPI0csn_GPIO1B6vccio0	GRF_GPIO1B_IOMUX_H [10:8]=3'b001

Notes: I=input, O=output, I/O=input/output, bidirectional

Furthermore, different IOs are selected and connected to different flash interface, which is shown as follows.

Table 7-2NandC Interface Connection

Module Pin	Direction	Flash Interface		
		Asyn8x	ONFI	Toggle
flash_csni(i=0~1)	O	√	√	√
flash_ale	O	√	√	√
flash_cle	O	√	√	√
flash_wrn	O	√	√	√
flash_rdn	O	√	√	√
flash_data[7:0]	I/O	√	√	√
flash_dqs	I/O	-	√	√
flash_rdy	I	√	√	√

7.6 Application Notes

7.6.1 BCHST/SPARE Application

7.6.1.1 BCHST

There are 16 BCHST-registers in NandC to store 32 codeword's BCH decode status(bchst) information. Every register stores 2 codeword's bchst information except BCHST0, which not only includes bchst information, but also includes one bit for *bchrdy*.
Letbchst_cwd0~bchst_cwd31 be the bchst information for 32 codewords. NandC support bchst transfer function. Software can enable the function by FLCTL[19]. When FLCTL[19]=1, Nandc will transmit the status of BCH to external memory, and software need configure spare step to 8. Detailed format for spare data and BCH status in every unit is shown in figures1.3.

7.6.1.2 SPARE

SPARE includes two register-groups, SPARE0 and SPARE1. Each group has 1 register: SPARE0_0 and SPARE1_0.
When in bch encoding, SPARE0_0 stores system information for codeword in sram0; SPARE1_0 stores system information for codeword in sram1.
When in bch decoding, SPARE0_0 stores the spare data read from flash for codeword in sram0; SPARE1_0 stores the spare data read from flash for codeword in sram1.

7.6.2 Bus Mode Application

MTRANS_CFG[2] determines whether the data load/store between internal memory and external memory is through slave interface or master interface.

7.6.2.1 Slave Mode

When MTRANS_CFG[2]=0, slave is selected. i. e. , flash data load/store between internal memory and external memory is through slave interface by cpu or external DMA.
In this mode, software should store page data into internal memory and spare data into SPARE registers before starting flash program operation; and should load page data from internal memory and spare data from SPARE registers after finishing flash read operation.
In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are unused. The transfer codeword number is determined by FLCTL[6:5], and the maximum number is 2. The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transfer is finished.

7.6.2.2 Master Mode

When MTRANS_CFG[2]=1, master is selected. i. e. , flash data load/store between internal memory and external memory is through master interface.
In this mode, software should initialize page data and spare data into external memory, and

set their addresses in MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash program operation. Similarly, software should configure MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting flash read operation and could read data from addresses in MTRANS_SADDR0 and MTRANS_SADDR1 after NandC transfer finish.
In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are used. The transfer codeword number is determined by FLCTL[27:22], and the maximum number is 32. The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transmission is finished.
When MTRANS_CFG[2]=1, page data and spare data are stored in the continuous space of external memory respectively.
For page data, source address is named Saddr0, specified in MTRANS_SADDR0. The space can be divided into many continuous units, and the unit size(named PUnit) is 1024 bytes or 512 bytes determined by FLCTL[21] and FLCTL[11]:

a. when FLCTL[11]=0, PUnit is always equal to 1024 bytes
b. when FLCTL[11]=1 and FLCTL[21]=0, PUnit is equal to 1024 bytes
c. when FLCTL[11]=1 and FLCTL[21]=1, PUnit is equal to 512 bytes

For spare data, source address is named Saddr1, specified in MTRANS_SADDR1. The space can be divided into many continuous units, and the unit size(named SUnit) is 4 bytes or 8 bytes determined by FLCTL[19]:

a. When FLCTL[19]=0 , SUnit is equal to 4 bytes
b. When FLCTL[19]=1 , SUnit is equal to 8 bytes

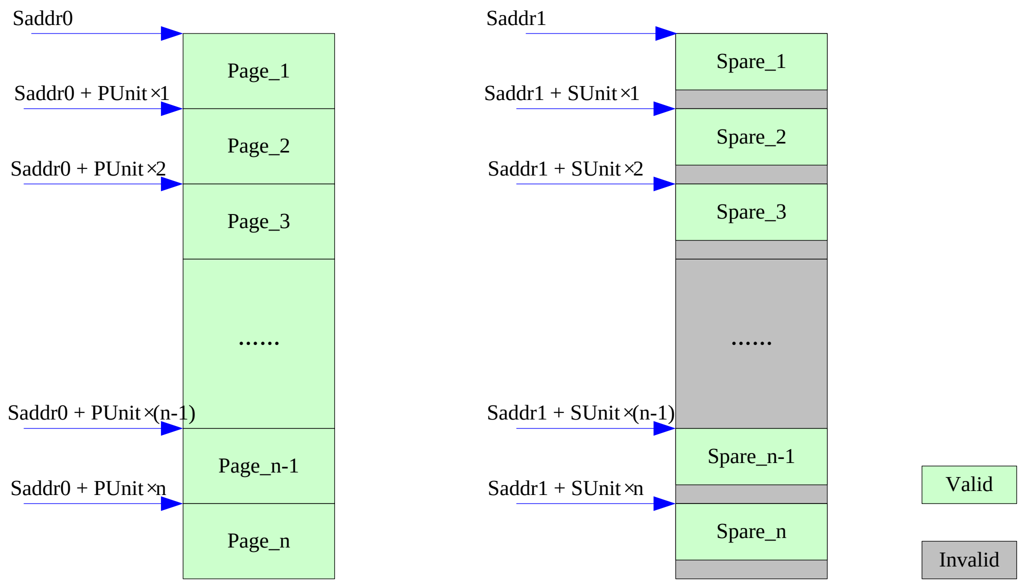


Fig.7-2NandC Address Assignment

The detailed format for page data and spare data in every unit is shown in following figures.

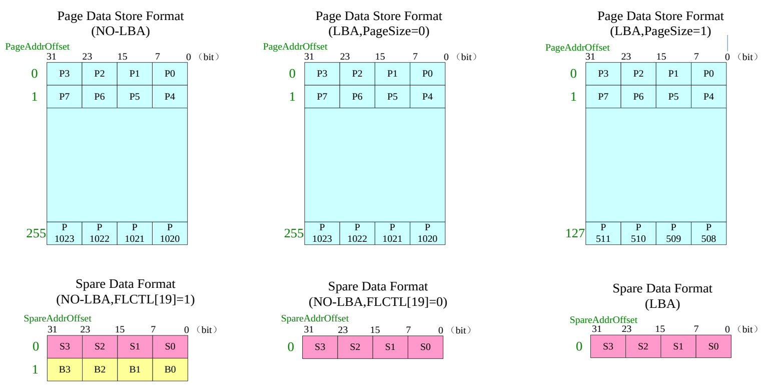

Page Data Store Format (NO-LBA)
Page Data Store Format (LBA,PageSize=0)
Page Data Store Format (LBA,PageSize=1)
PageAddrOffset
31 23 15 7 0 (bit)
P3 P2 P1 P0
P7 P6 P5 P4
P1023 P1022 P1021 P1020
P511 P510 P509 P508
0
1
255
127
Spare Data Format (NO-LBA,FLCTL[19]=1)
Spare Data Format (NO-LBA,FLCTL[19]=0)
Spare Data Format (LBA)
SpareAddrOffset
S3 S2 S1 S0
B3 B2 B1 B0

Fig.7-3NandC DataFormat

7.6.3 BchPage Application

BCHCTL[16] determines whether codeword size for page data is 1024 bytes or 512 bytes when FLCTL[11] is 0.

7.6.3.1 1024bytes

When BCHCTL[16]=0, BchPage=0, hardware needs to write 1024 bytes page data and spare data into flash or read 1024 bytes page data and spare data from flash. All the 1024 bytes page data and spare data are encoded when writing or decoded when reading.

7.6.3.2 512bytes

When BCHCTL[16]=1, BchPage=1, hardware needs to write 512 bytes page data and spare data into flash or read 512 bytes page data and spare data from flash.
In this mode, the page data unit size for BCH encoder and BCH decoder is still 1024byte. So to support BCH encoder and decoder, software should configure page data as follows:1th~512th bytes are invalid data which must be stuffed with 0xff, 513th~1024th bytes arevalid page data.
However, Randomizer function is not supported under this condition.

7.6.4 PageSize/SpareSize Application

FLCTL[21] determines whether the codeword size is 1024 bytes or 512 bytes when FLCTL[11] is 1.

7.6.4.1 Big Page

When FLCTL[11]=0(LbaEn=0), the flash to be operated is Raw NAND Flash. Every codeword size is 1024 bytes and FLCTL[21] should always be set to 0, and the PageStep in external memory is 1024 bytes if bus mode is master mode.
At this mode, the spare size and SpareStep in external memory are determined by FLCTL[19] as follows:
FLCTL[19]=0: spare size=4bytes , SpareStep=4bytes
FLCTL[19]=1: spare size=4bytes , SpareStep=8bytes

7.6.4.2 Small Page

When FLCTL[11]=1, LbaEn=1, the flash to be operated is Managed NAND Flash. Every codeword size could be 1024 bytes or 512 bytes according to FLCTL[21]. If FLCTL[21]=0, codeword size is 1024 bytes, PageStep in external memory is 1024 bytes, and SpareStep is 4bytes. If FLCTL[21]=1, codeword size is 512 bytes, PageStep in external memory is 512 bytes, and SpareStep is 4 bytes.
At this mode, the spare size is configured in FLCTL[12], and the max available number is 4.
In the summary, the total data size in every codeword for flash or for software including page data and spare data, is determined by BCHCTL[27:25], FLCTL[11], FLCTL[21], BCHCTL[4]. Their relationship is shown as follows.

Table 7-3NandC Page/Spare size for flash

Page/spare size for software		Page size/codeword	Spare size/codeword
FLCTL[11]=0	24bit ECC	1024 byte	(4+42)byte
	40 bit ECC	1024 byte	(4+70)byte
	60 bit ECC	1024 byte	(4+105)byte
	70 bit ECC	1024 byte	(4+123)byte
FLCTL[11]=1	FLCTL[21]=0	1024 byte	FLCTL[12]
	FLCTL[21]=1	512 byte	FLCTL[12]

Notes: that "page/spare size for flash" means that hardware should transfer these numbers of bytes in every codeword to or from flash.

7.6.5 Randomizer Application

RANDMZ_CFG[31] determines whether randomizer is enable or not. When RANDMZ_CFG[31] equals to 1, randomizer is active. Data should be scrambled before written into flash, and descrambled after read from flash.
RANDMZ_CFG[30] determines the randomizer polynomial.
  When RANDMZ_CFG[30]=0, Polynomial=1+x+x^15
  When RANDMZ_CFG[30]=1, Polynomial=1+x^14+x^15
RANDMZ_CFG[19:0] is the seed for randomizer. It should be ensured that data in the same page should have the same randomizer polynomial and randomizer seed when in flash program or flash read operation.
The data unit for randomizer is one codeword(data+spare).

However, Randomizer is just available for data transfer by internal DMA mode, but not by for bypass mode. Furthermore, it should not be enable if BCHCTL[16]=0 (BchPage=512bytes).

7.6.6 DLL Application

When Toggle Flashor ONFI Synchronous Flash interface is active, DLL should be used to adjust DQS input with DQ when reading flash.
There are 2 registers for DLL configuration(DLL_CFG_REG0 and DLL_CFG_REG1), and 1 register for DLL status(DLL_OBS_REG0).
The usage guide is as follows:
If bypass mode is used, you should set *dll_bypass* in DLL_CFG_REG1[1] to 1, and set *dll_dqs_dly_bypass* in DLL_CFG_REG0[23:16] to determine the dll element number needed. And then set *dll_start* in DLL_CFG_REG1[0] to 1 to start the DLL.
If auto adjusting is used, you should set *dll_bypass* in DLL_CFG_REG1[1] to 0, and set the *dll_start_point* in DLL_CFG_REG0[7:0] and *dll_incr* in DLL_CFG_REG1[11:4]. You also should set the adjusting mode *dll_qtren* in DLL_CFG_REG1[3:2] to compute the dll element number needed. If *dll_qtren*=2'b00, the dll element number is determined by *dll_dqs_dly* in DLL_CFG_REG0[15:8]; otherwise, it is 1/4 or 1/8 of the total number of dll elements used for *dll_qtren*=2'b01 or *dll_qtren*=2'b10 separately. The last step is to set *dll_start* in DLL_CFG_REG1[0] to 1 to start the DLL.
If you want to monitor the dll working status, you could read DLL_OBS_REG0. If DLL_OBS_REG0[0]=0, it means that DLL is not locked, and still in detecting status. Otherwise, it means that DLL is locked, and *dll_lock_value* in DLL_OBS_REG0[8:1] is the total number of dll elements used, *dll_dqs_delay_value* in DLL_OBS_REG0[16:9] is the total number of DQS delay used.

7.6.7 NandC Interrupt Application

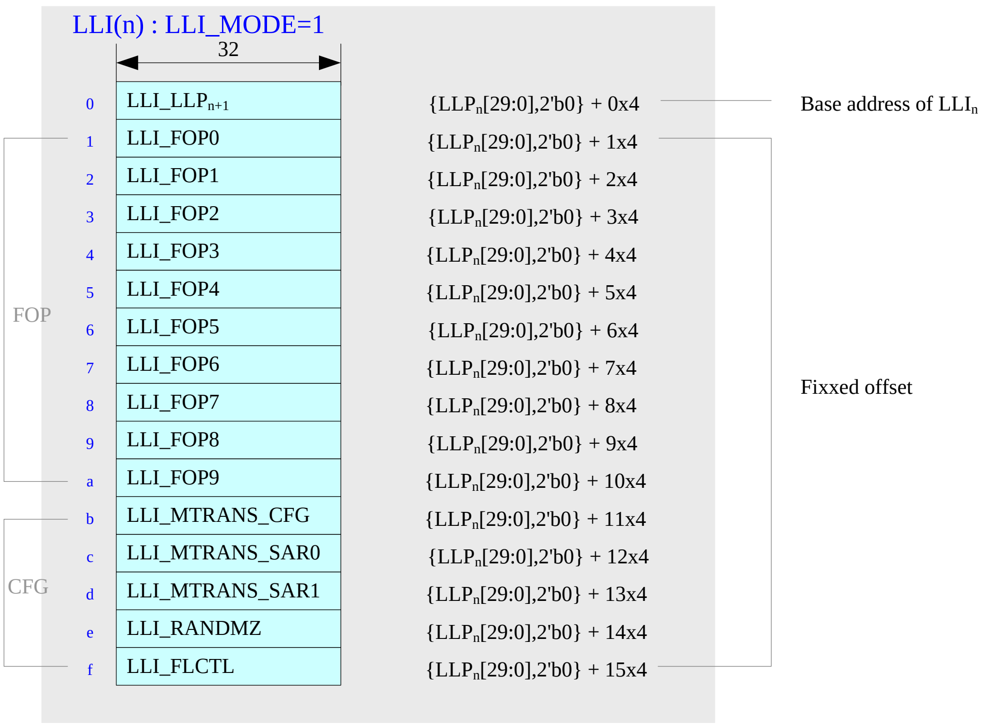
NandC has 1 interrupt output signal and 10 interrupt sources: seed bch error interrupt source, seed bch fail interrupt source, read first page interrupt source, master idle interrupt source, flash abort interrupt source, LLP interrupt source, dma finish interrupt source, flash ready interrupt source, bch error interrupt source, bchfail interrupt source. When one or more of these interrupt source are enabled, NandC interrupt is asserted if one or more interrupt source is high. Software can determine the interrupt source by reading INTST and clear interrupt by writing corresponding bit in INTCLR.

7.6.8 LLP Application

LLP is used in NandC to store and execute instruction groups configured in external memory by software. When LLPCTL[0]=1, LLP is active, NandC will load instruction groups stored in {LLPCTL[31:6], 6'h0} and execute them. Next instruction groups should not be loaded until current instruction execution finished.

7.6.8.1 LLP Structure

The structure of LLP is shown as follows:


LLI(n) : LLI_MODE=1
32
0 LLI_LLP$_{n+1}$ {LLP$_n$[29:0],2'b0} + 0x4
1 LLI_FOP0 {LLP$_n$[29:0],2'b0} + 1x4
2 LLI_FOP1 {LLP$_n$[29:0],2'b0} + 2x4
3 LLI_FOP2 {LLP$_n$[29:0],2'b0} + 3x4
4 LLI_FOP3 {LLP$_n$[29:0],2'b0} + 4x4
5 LLI_FOP4 {LLP$_n$[29:0],2'b0} + 5x4
6 LLI_FOP5 {LLP$_n$[29:0],2'b0} + 6x4
7 LLI_FOP6 {LLP$_n$[29:0],2'b0} + 7x4
8 LLI_FOP7 {LLP$_n$[29:0],2'b0} + 8x4
9 LLI_FOP8 {LLP$_n$[29:0],2'b0} + 9x4
a LLI_FOP9 {LLP$_n$[29:0],2'b0} + 10x4
b LLI_MTRANS_CFG {LLP$_n$[29:0],2'b0} + 11x4
c LLI_MTRANS_SAR0 {LLP$_n$[29:0],2'b0} + 12x4
d LLI_MTRANS_SAR1 {LLP$_n$[29:0],2'b0} + 13x4
e LLI_RANDMZ {LLP$_n$[29:0],2'b0} + 14x4
f LLI_FLCTL {LLP$_n$[29:0],2'b0} + 15x4
FOP
CFG
Base address of LLI$_n$
Fixxed offset

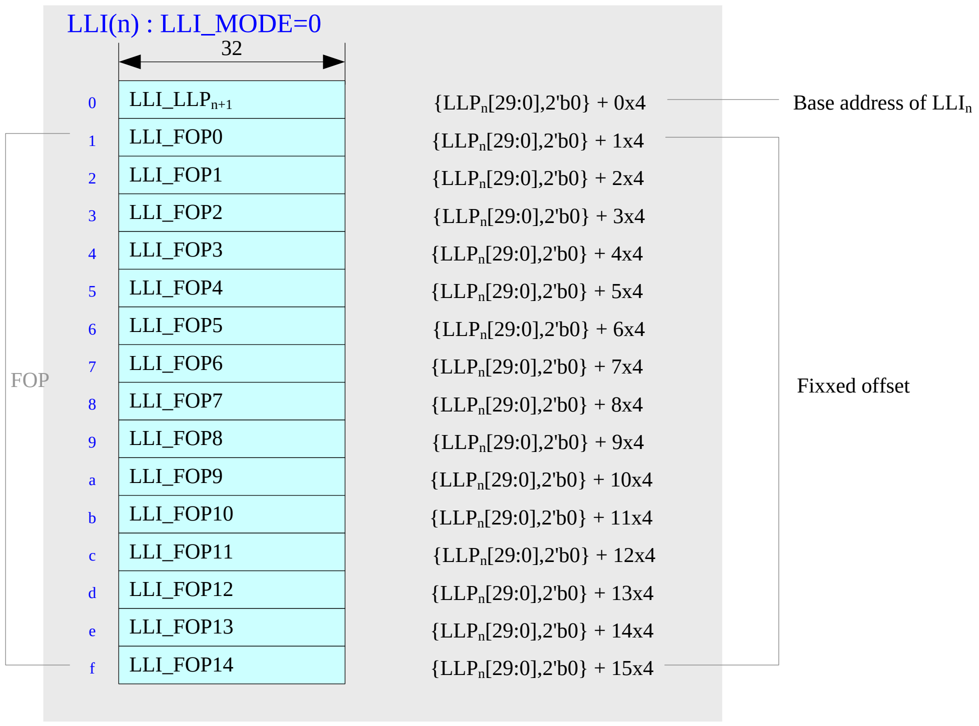
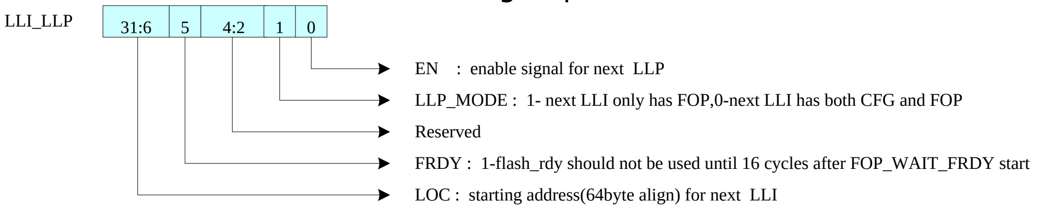

LLI(n) : LLI_MODE=0
32
0 LLI_LLP$_{n+1}$ {LLP$_n$[29:0],2'b0} + 0x4
1 LLI_FOP0 {LLP$_n$[29:0],2'b0} + 1x4
2 LLI_FOP1 {LLP$_n$[29:0],2'b0} + 2x4
3 LLI_FOP2 {LLP$_n$[29:0],2'b0} + 3x4
4 LLI_FOP3 {LLP$_n$[29:0],2'b0} + 4x4
5 LLI_FOP4 {LLP$_n$[29:0],2'b0} + 5x4
6 LLI_FOP5 {LLP$_n$[29:0],2'b0} + 6x4
7 LLI_FOP6 {LLP$_n$[29:0],2'b0} + 7x4
8 LLI_FOP7 {LLP$_n$[29:0],2'b0} + 8x4
9 LLI_FOP8 {LLP$_n$[29:0],2'b0} + 9x4
a LLI_FOP9 {LLP$_n$[29:0],2'b0} + 10x4
b LLI_FOP10 {LLP$_n$[29:0],2'b0} + 11x4
c LLI_FOP11 {LLP$_n$[29:0],2'b0} + 12x4
d LLI_FOP12 {LLP$_n$[29:0],2'b0} + 13x4
e LLI_FOP13 {LLP$_n$[29:0],2'b0} + 14x4
f LLI_FOP14 {LLP$_n$[29:0],2'b0} + 15x4
FOP
Base address of LLI$_n$
Fixxed offset

Fig.7-4NandC LLP Data Format

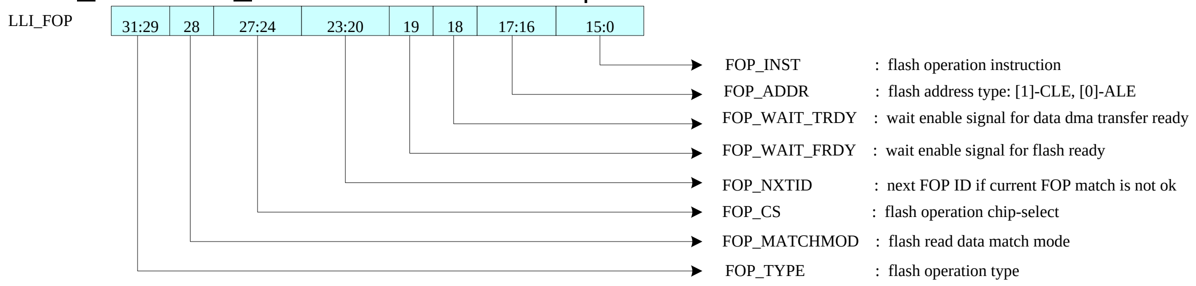
LLI_MODE is determined by LLPCTL[1]. If current operation is flash program or flash read, then LLI_MODE=1 is need; otherwise, LLI_MODE=0 is workable.
In addition, you could do more than one flash operation in one LLP group, but you should not separate one flash operation into two LLI groups.

7.6.8.2 LLI Format

a. LLI_LLP$_{n+1}$ stores the address for next LLI group data


LLI_LLP
31:6
5
4:2
1
0
EN : enable signal for next LLP
LLP_MODE : 1- next LLI only has FOP,0-next LLI has both CFG and FOP
Reserved
FRDY : 1-flash_rdy should not be used until 16 cycles after FOP_WAIT_FRDY start
LOC : starting address(64byte align) for next LLI

b. LLI_FOP0~LLI_FOP14 store the flash operation instruction


LLI_FOP
31:29
28
27:24
23:20
19
18
17:16
15:0
FOP_INST : flash operation instruction
FOP_ADDR : flash address type: [1]-CLE, [0]-ALE
FOP_WAIT_TRDY : wait enable signal for data dma transfer ready
FOP_WAIT_FRDY : wait enable signal for flash ready
FOP_NXTID : next FOP ID if current FOP match is not ok
FOP_CS : flash operation chip-select
FOP_MATCHMOD : flash read data match mode
FOP_TYPE : flash operation type

FOP_TYPE:
000 : nop
001 : flash write
010 : flash read
011 : flash read with match operation
100 : DMA

When FOP_TYPE=3'b011, match operation is active, and the PATTERN is LLI_FOP[15:0]. It is matched when "RDATA|PATTERN=PATTERN" with FOP_MATCHMOD=0, or when "RDATA&PATTERN=PATTERN" with FOP_MATCHMOD=1.
c. LLI_MTRANS_CFG/LLI_MTRANS_SADDR0/LLI_MTRANS_SADDR1/ LLI_RANDMZ/ LLI_FLCTL store the configuration for MTRANS_CFG/ MTRANS_SADDR0/MTRANS_SADDR1/RANDMZ/FLCTL.

7.6.8.3 LLP Working Mode

There are two working modes for LLP:
a. Normal mode: LLPCTL[0] is kept to 1 until all LLP loading and executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].
b. Pause mode: LLPCTL[0] is changed from 1 to 0 during LLP loading or LLP executing. NandC should not stop working until current LLP executing finished. Software can monitor the progress by LLPSTAT[31:6], LLPSTAT[0].

7.6.9 Seed Application

Nandc supports randomizer seed transmission. When FLCTL[9]=1 and RANDMZ_CFG[31]=1,Nandc will transmit seed to flash before page data transmission and receive seed before page data receiving.
Seed has BCH encoder/decoder separately and support 1bit BCH. Software can query seed BCH result by accessing SEED_BCHST.

7.6.10 Redundance Application

Nandc supports write "FF" to flash as redundance. Software can configure redundance size by NANDC_MTRANS_CFG[26:16].

7.6.11 IOMUX Application

Nandc support IOMUX. Software can change pin function by FMCTL[23:21].

Chapter 8 Power Management Unit (PMU)

8.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in PX30. The PX30 PMU is dedicated for managing the power of the whole chip.

8.1.1 Features

- Support 3 voltage domains: VD_CORE, VD_LOGIC, VD_PMU
- Support power off VD_CORE only
- 4 Power domains in VD_CORE:PD_CPU_0/1/2/3
- PD_CPU_0/1/2/3 support cpu auto power down ， support SCU auto power down
- power domains in VD_LOGIC include PD_GPU, PD_VPU, PD_VI, PD_VO, PD_MMC_NAND, PD_SDIO, PD_MAC, PD_DDR
- Support DDR self-refresh, auto-gating and retention
- Support wakeup source
  - Timer
  - Usb detect
  - Sdmmc detect
  - Sdio
  - Interrupt of Gpio0
  - Timeout
  - GPIO0A[7:0], GPIO0B[7:0], GPIO0C[4:0]
  - Uart0
  - Interrupt output from GIC
- Support Flush L2 by software and hardware
- Support NIU idle interface(idle request , ack and status)

8.2 Block Diagram

8.2.1 Voltage partition

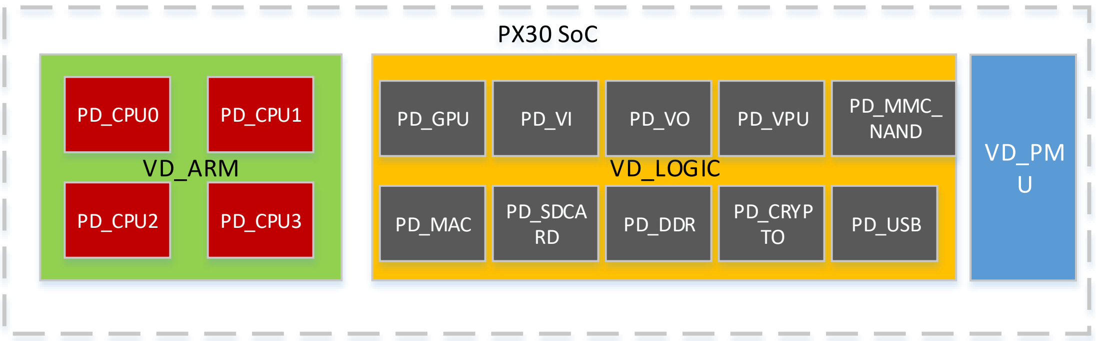


Fig. 8-1PX30 Power Domain Partition

The above diagram describes voltage domain partition.

Table 8-1PX30 Power Domain and Voltage Domain Summary

Voltage Domain	**Blocks (not real power domain)**	**Description**
VD_ARM	PD_CPU0	CPU Core 0 with NEON and FPU
	PD_CPU1	CPU Core 1 with NEON and FPU
	PD_CPU2	CPU Core 2 with NEON and FPU
	PD_CPU3	CPU Core 3 with NEON and FPU
	PD_SCU(ALIVE)	DAP Lite, SCU and 256KB L2
VD_LOGIC	PD_GPU	GPU

Voltage Domain	Blocks (not real power domain)	Description
	PD_VI	ISP and VIP
	PD_VO	VOP_M, VOP_S, RGA and DSI
	PD_VPU	VCODEC
	PD_DDR	DDR_CTRL, DDR_GRF, DDR_STDBY and DDR_MONITOR
	PD_MAC	MAC
	PD_MMC_NAND	SFC, EMMC, NAND and SDIO
	PD_SDCARD	SDCARD
	PD_USB	USB_OTG and USB_HOST
	PD_CRYPTO	CRYPTO
	PD_BUS(ALIVE)	DCF，DMAC, GIC, I2S0/1/2, PDM, INTMEM, ROM, OTP_S, KEYREADER, USB_GRF, CRU, CPU_BOOST,GRF,I2C,WDT_S/NS, TIMER_S/NS, TSADC,SARADC, OTP_NS, SPI, PWM, GPIO1/2/3, UART1/2/3/4/5, DCF, PLL and ANALOG PHYs
VD_PMU	PD_PMU	PMU, UART0, GPIO0, PMU_GRF, PMU_INTMEM and SGRF

8.2.2 PMU block diagram

The following figure is the PMU block diagram. The PMU includes the 3 following sections:

- APB interface and register, which can accept the system configuration
- Low Power State Control, which generate low power control signals.
- Power Switch Control, which control all power domain switch

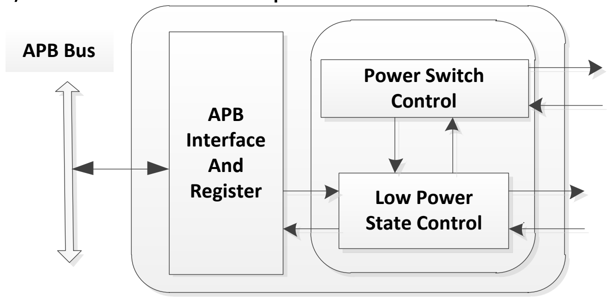


Fig. 8-2PMU Bock Diagram

8.3 Function Description

First of all, we define two operation modes of PMU, normal mode and low power mode.
When operating at normal mode, that means software can manage power sources directly by accessing PMU register.
For example, Cortex-A35 CPU can write PMU_PWRDN_CON register to determine that power off/on which power domain independently.
When operating at low power mode, software manages power sources indirectly through FSM (Finite States Machine) in PMU and those settings always not take effect immediately. That means software also can configure PMU registers to power down/up some power resources, but these setting will not be executed immediately after configuration. They will delay to execute after FSM running in particular phase.
To entering low power mode, after setting some power configurations, the PMU_POWER_MODE[0] bit must be set 1 to enable PMU FSM. Then Cortex-A35 CPU needs to execute a WFI command to perform ready signal. After PMU detects all Cortex-A35 CPUs in WFI status, then the FSM will be fetched. And the specific power sources will be controlled during specific status in FSM. So the low power mode is a “delay affect” way to handle power sources inside the PX30 chip.

8.4 Register Description

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG0_LO	0x0000	W	0x00000000	Wakeup source config 0 low 16 bit
PMU_WAKEUP_CFG0_HI	0x0004	W	0x00000000	Wakeup source config 0 high 16 bit
PMU_WAKEUP_CFG1_LO	0x0008	W	0x00000000	Wakeup source config 1 low 16 bit
PMU_WAKEUP_CFG1_HI	0x000c	W	0x00000000	Wakeup source config 1 high 16 bit
PMU_WAKEUP_CFG2_LO	0x0010	W	0x00000000	Wakeup source config 0 low 16 bit
PMU_PWRDN_CON_LO	0x0018	W	0x00000000	power down control register
PMU_PWRDN_ST	0x0020	W	0x00000000	power status register(read only)
PMU_PWRMODE_CORE_CON_LO	0x0024	W	0x00000000	power mode control register for core low 16 bit
PMU_PWRMODE_CORE_CON_HI	0x0028	W	0x00000000	power mode control register for core high 16 bit
PMU_PWRMODE_COMMON_CON_LO	0x002c	W	0x00000000	power mode control register for chip low 16 bit
PMU_PWRMODE_COMMON_CON_HI	0x0030	W	0x00000000	power mode control register for chip high 16 bit
PMU_SFT_CON_LO	0x0034	W	0x00000000	software configure register
PMU_BUS_IDLE_REQ_LO	0x0064	W	0x00000000	bus idle request register
PMU_BUS_IDLE_ST	0x006c	W	0x00000000	bus idle status register
PMU_OSC_CNT_LO	0x0074	W	0x00005dc0	osc count low 16 bit
PMU_OSC_CNT_HI	0x0078	W	0x00000000	osc count high 16 bit
PMU_PLLLOCK_CNT_LO	0x007c	W	0x00005dc0	plllock count low 16 bit
PMU_PLLLOCK_CNT_HI	0x0080	W	0x00000000	plllock count low high bit
PMU_PLLRST_CNT_LO	0x0084	W	0x00005dc0	pll reset count low 16 bit
PMU_PLLRST_CNT_HI	0x0088	W	0x00000000	pll reset count high 16 bit
PMU_STABLE_CNT_LO	0x008c	W	0x00005dc0	PMU stable count low 16 bit
PMU_STABLE_CNT_HI	0x0090	W	0x00000000	PMU stable count high 16 bit
PMU_WAKEUP_RST_CLR_CNT_LO	0x009c	W	0x00005dc0	wakeup reset count low 16 bit
PMU_WAKEUP_RST_CLR_CNT_HI	0x00a0	W	0x00000000	wakeup reset count high 16 bit
PMU_DDR_SREF_ST	0x00a4	W	0x00000000	ddr self-refresh status
PMU_SYS_REG0_LO	0x00a8	W	0x00000000	system register0 low 16 bit
PMU_SYS_REG0_HI	0x00ac	W	0x00000000	system register0 high 16 bit
PMU_SYS_REG1_LO	0x00b0	W	0x00000000	system register1 low 16 bit
PMU_SYS_REG1_HI	0x00b4	W	0x00000000	system register1 high 16 bit
PMU_SYS_REG2_LO	0x00b8	W	0x00000000	system register2 low 16 bit
PMU_SYS_REG2_HI	0x00bc	W	0x00000000	system register2 high 16 bit
PMU_SYS_REG3_LO	0x00c0	W	0x00000000	system register3 low 16 bit

Name	Offset	Size	Reset Value	Description
PMU_SYS_REG3_HI	0x00c4	W	0x00000000	system register3 high 16 bit
PMU_SCU_PWRDN_CNT_LO	0x00c8	W	0x00005dc0	scu power down count low 16 bit
PMU_SCU_PWRDN_CNT_HI	0x00cc	W	0x00000000	scu power down count high 16 bit
PMU_SCU_PWRUP_CNT_LO	0x00d0	W	0x00005dc0	scu power up count low 16 bit
PMU_SCU_PWRUP_CNT_HI	0x00d4	W	0x00000000	scu power up count low 16 bit
PMU_TIMEOUT_CNT_LO	0x00d8	W	0x00005dc0	time out count low 16 bit
PMU_TIMEOUT_CNT_HI	0x00dc	W	0x00000000	time out count high 16 bit
PMU_CPU0APM_CON	0x00e0	W	0x00000000	cpu0 apm control register
PMU_CPU1APM_CON	0x00e4	W	0x00000000	cpu1 apm control register
PMU_CPU2APM_CON	0x00e8	W	0x00000000	cpu2 apm control register
PMU_CPU3APM_CON	0x00ec	W	0x00000000	cpu3 apm control register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

8.4.2 Detail Register Description

PMU_WAKEUP_CFG0_LO
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x0000	wakeup_gpio_pos_en_lo wakeup posedge enable for gpio 0 [15:0]

PMU_WAKEUP_CFG0_HI
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x0000	wakeup_gpio_pos_en_hi wakeup posedge enable for gpio 0 [31:16]

PMU_WAKEUP_CFG1_LO
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x0000	wakeup_gpio_neg_en_lo wakeup posedge enable for gpio 0 [15:0]

PMU_WAKEUP_CFG1_HI

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x0000	wakeup_gpio_neg_en_hi wakeup posedge enable for gpio 0 [31:16]

PMU_WAKEUP_CFG2_LO
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:11	RO	0x0	reserved
10	RW	0x0	wakeup_timeout_en timeout wakeup enable
9	RO	0x0	reserved
8	RW	0x0	wakeup_sft_en software wakeup enable
7	RW	0x0	wakeup_usbdev_en usb wakeup enable
6	RW	0x0	wakeup_timer_en timer wakeup enable
5	RW	0x0	wakeup_uart0_en uart0 wakeup enable
4	RW	0x0	wakeup_sdmmc_en sdmmc wakeup enable
3	RW	0x0	wakeup_sdio_en sdio wakeup enable
2	RW	0x0	wakeup_gpio0_int_en gpio0 interrupt wakeup enable
1	RO	0x0	reserved
0	RW	0x0	wakeup_int_cluster_en cluster interrupt wakeup enable

PMU_PWRDN_CON_LO
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15	RW	0x0	pd_gpu_pwrdwn_en pd_gpu power down enable
14	RW	0x0	pd_vi_pwrdwn_en pd_vi power down enable
13	RW	0x0	pd_vo_pwrdwn_en pd_vo power down enable

Bit	Attr	Reset Value	Description
12	RW	0x0	pd_vpu_pwrdwn_en pd_vpu power down enable
11	RW	0x0	pd_mmc_nand_pwrdwn_en pd_mmc_nand power down enable
10	RW	0x0	pd_MAC_pwrdwn_en pd_MAC power down enable
9	RW	0x0	pd_crypto_pwrdwn_en pd_crypto power down enable
8	RW	0x0	pd_sdcard_pwrdwn_en pd_sdcard power down enable
7	RO	0x0	reserved
6	RW	0x0	pd_ddr_pwrdwn_en pd_ddr power down enable
5	RW	0x0	pd_usb_pwrdwn_en pd_usb power down enable
4	RW	0x0	pd_scu_pwrdwn_en pd_scu power down enable
3	RW	0x0	pd_a35_3_pwrdwn_en pd_a35_3 power down enable
2	RW	0x0	pd_a35_2_pwrdwn_en pd_a35_2 power down enable
1	RW	0x0	pd_a35_1_pwrdwn_en pd_a35_1 power down enable
0	RW	0x0	pd_a35_0_pwrdwn_en pd_a35_0 power down enable

PMU_PWRDN_ST
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	pd_gpu_pwr_status 1: pd_gpu is power down 0: pd_gpu is power up
14	RW	0x0	pd_vi_pwr_status 1: pd_vi is power down 0:pd_vi is power up
13	RW	0x0	pd_vo_pwr_status 1: pd_vo is power down 0: pd_vi is power up
12	RW	0x0	pd_vpu_pwr_status 1: pd_vpu is power down 0: pd_vpu is power up

Bit	Attr	Reset Value	Description
11	RW	0x0	pd_mmc_nand_pwr_status 1: pd_mmc_nand is power down 0: pd_mmc_nand is power up
10	RW	0x0	pd_MAC_pwr_status 1: pd_MAC is power down 0: pd_MAC is power up
9	RW	0x0	pd_crypto_pwr_status 1: pd_crypto is power down 0: pd_crypto is power up
8	RW	0x0	pd_sdcard_pwr_status 1: pd_sdcard is power down 0: pd_sdcard is power up
7	RO	0x0	reserved
6	RW	0x0	pd_ddr_pwr_status 1: pd_ddr is power down 0: pd_ddr is power up
5	RW	0x0	pd_usb_pwr_status 1: pd_usb is power down 0: pd_usb is power up
4	RW	0x0	pd_scu_pwr_status 1: pd_scu is power down 0: pd_scu is power up
3	RW	0x0	pd_a35_3_pwr_status 1: pd_a35_3 is power down 0: pd_a35_3 is power up
2	RW	0x0	pd_a35_2_pwr_status 1: pd_a35_2 is power down 0: pd_a35_2 is power up
1	RW	0x0	pd_a35_1_pwr_status 1: pd_a35_1 is power down 0: pd_a35_1 is power up
0	RO	0x0	pd_a35_0_pwr_status 1: pd_a35_0 is power down 0: pd_a35_0 is power up

PMU_PWRMODE_CORE_CON_LO

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:12	RO	0x0	reserved
11	RW	0x0	clr_peri2msch clear peri2msch niu when power down

Bit	Attr	Reset Value	Description
10	RW	0x0	clr_bus2main clr bus2main niu when power down
9	RW	0x0	l2_flush_en 1: flush L2 when in power mode
8	RW	0x0	l2_idle_en 1: wait for L2 idle when in power mode
7	RO	0x0	reserved
6	RW	0x0	scu_pd_en 1: power down scu(vd_core) when power mode
5	RW	0x0	clr_core 1: clear core niu when power mode
4	RO	0x0	reserved
3	RW	0x0	cpu0_pd_en 1: power down cpu0 when power mode
2	RO	0x0	reserved
1	RW	0x0	clk_core_src_gate_en 1: core clock gating when power mode
0	RW	0x0	global_int_disable_cfg 1:global interrupt disable 0:global interrupt enable

PMU_PWRMODE_CORE_CON_HI
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:8	RO	0x0	reserved
7	RW	0x0	npll_pd_en 1: power down npll when power mode
6	RW	0x0	gpll_pd_en 1: power down gpll when power mode
5	RW	0x0	cpll_pd_en 1: power down cpll when power mode
4	RW	0x0	dpll_pd_en 1: power down dpll when power mode
3	RW	0x0	apll_pd_en 1: power down apll when in power mode
2:0	RO	0x0	reserved

PMU_PWRMODE_COMMON_CON_LO
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15	RW	0x0	clr_peri_pmu 1: clear peri mid niu when in power mode
14	RW	0x0	clr_pmu 1:clear pmu niu when in power mode
13	RW	0x0	ddr_ret_de_req de-request for ddr retention bit
12	RW	0x0	ddr_ret_en ddr retention when in power mode
11	RW	0x0	ddrc_gating_en gating ddrc clock when in power mode
10	RW	0x0	sref_enter_en ddr enter self-refresh when power mode
9	RW	0x0	input_clamp_en 1: clamp pmu input when in power mode
8	RW	0x0	osc_24m_dis 1: disable 24M osc when power mode
7	RW	0x0	alive_use_lf 1: alive switch to low freqency clock when power mode
6	RW	0x0	pmu_use_lf 1: pmu clock swith to low frequency when in power mode
5	RO	0x0	reserved
4	RW	0x0	pll_pd_en 1: power down pll when in power mode
3	RW	0x0	wakeup_reset_en 1: wake up resetn when in power mode
2	RO	0x0	reserved
1	RW	0x0	ddr_pd_en 1: power down pd_ddr when power mode
0	RW	0x0	power_mode_en 1: power mode enable

PMU_PWRMODE_COMMON_CON_HI
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:14	RO	0x0	reserved
13	RW	0x0	pd_bus_clk_src_gate_en clock gating bus niu when in power mode
12	RW	0x0	pd_peri_clk_src_gate_en clock gating peri niu when in power mode

Bit	Attr	Reset Value	Description
11	RW	0x0	wait_wakeup_begin_cfg start to oberserve wakeup source
10	RW	0x0	clr_crypto clear crypto niu when in power mode
9	RW	0x0	clr_vpu clear vpu niu when in power mode
8	RW	0x0	clr_usb clear usb niu when in power mode
7	RW	0x0	clr_gpu clear gpu niu when in power mode
6	RW	0x0	clr_vi clear vi niu when in power mode
5	RW	0x0	clr_vo clear vo when in power mode
4	RW	0x0	clr_MAC clear MAC niu when in power mode
3	RW	0x0	clr_nandc clear nandc niu when in power mode
2	RW	0x0	clr_msch clear msch niu when in power mode
1	RW	0x0	clr_mmc clear mmc niu when in power mode
0	RW	0x0	clr_bus 1: clear bus niu when in power mode

PMU_SFT_CON_LO
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:11	RO	0x0	reserved
10	RW	0x0	upctl_c_sysreq_cfg software config upctl for idle
9:7	RO	0x0	reserved
6	RW	0x0	l2flushreq_cluster_cfg software flush L2 config
5	RW	0x0	gpll_pd_cfg software config gpll power down
4	RW	0x0	cpll_pd_cfg software config cpll power down
3	RW	0x0	dpll_pd_cfg software config dpll power down
2	RW	0x0	apll_pd_cfg software config apll power down

Bit	Attr	Reset Value	Description
1	RW	0x0	npll_pd_cfg software config npll power down
0	RW	0x0	wakeup_sft software wake up , a 0 to 1 posedge make it work(no use for gemini project)

PMU_BUS_IDLE_REQ_LO
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15	RW	0x0	idle_req_peri2msch_cfg software configperi2msch niu idle request
14	RW	0x0	idle_req_vpu_cfg software config vpu niu idle request
13	RW	0x0	idle_req_pmu_cfg software config pmu niu idle request
12	RW	0x0	idle_req_peri_mid_cfg software config peri_mid niu idle request
11	RW	0x0	idle_req_msch_cfg software config msch niu idle request
10	RW	0x0	idle_req_usb_cfg software config usb niu idle request
9	RW	0x0	idle_req_sdcard_cfg software config sdcard niu idle request
8	RW	0x0	idle_req_vi_cfg software config vi niu idle request
7	RW	0x0	idle_req_vo_cfg software config vo niu idle request
6	RW	0x0	idle_req_MAC_cfg software config MAC niu idle request
5	RW	0x0	idle_req_mmc_nand_cfg software config mmc_nand niu idle request
4	RW	0x0	idle_req_crypto_cfg software config crypto niu idle request
3	RW	0x0	idle_req_core_cfg software config core niu idle request
2	RW	0x0	idle_req_gpu_cfg software config bus niu idle request
1	RW	0x0	idle_req_bus2main_cfg software config bus2main niu idle request
0	RW	0x0	idle_req_bus_cfg software config bus niu idle request

PMU_BUS_IDLE_ST

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31	RW	0x0	idle_peri2msch peri2msch niu idle status
30	RW	0x0	idle_vpu vpu niu idle status
29	RW	0x0	idle_pmu pmu niu idle status
28	RW	0x0	idle_peri_mid peri_mid niu idle status
27	RW	0x0	idle_msch msch niu idle status
26	RW	0x0	idle_usb usb niu idle status
25	RW	0x0	idle_sdcard sdcard niu idle status
24	RW	0x0	idle_vi vi niu idle status
23	RW	0x0	idle_vo vo niu idle status
22	RW	0x0	idle_MAC MAC niu idle status
21	RW	0x0	mmc_nand_idle bus niu idle status
20	RW	0x0	idle_crypto crypto niu idle status
19	RW	0x0	idle_core core niu idle status
18	RW	0x0	idle_gpu gpu niu idle status
17	RW	0x0	idle_bus2main bus2main niu idle status
16	RW	0x0	idle_bus bus niu idle status
15	RO	0x0	idle_ack_peri2msch peri2msch niu idle ack status
14	RO	0x0	idle_ack_vpu vpu niu idle ack status
13	RO	0x0	idle_ack_pmu pmu niu idle ack status
12	RO	0x0	idle_ack_peri_mid peri_mid niu idle ack status
11	RO	0x0	idle_ack_msch msch niu idle ack status
10	RO	0x0	idle_ack_usb usb niu idle ack status

Bit	Attr	Reset Value	Description
9	RO	0x0	idle_ack_sdcard sdcard niu idle ack status
8	RO	0x0	idle_ack_vi vi niu idle ack status
7	RO	0x0	idle_ack_vo vo niu idle ack status
6	RO	0x0	idle_ack_MAC MAC niu idle ack status
5	RO	0x0	idle_ack_mmc_nand mmc_nand niu idle ack status
4	RO	0x0	idle_ack_crypto crypto niu idle ack status
3	RO	0x0	idle_ack_core core niu idle ack status
2	RO	0x0	idle_ack_gpu gpu niu idle ack status
1	RO	0x0	idle_ack_bus2main bus2main niu idle ack status
0	RO	0x0	idle_ack_bus bus niu idle ack status

PMU_OSC_CNT_LO
Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_osc_cnt_lo osc_cnt[15:0]

PMU_OSC_CNT_HI
Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_osc_cnt_hi osc_cnt[19:16]

PMU_PLLLOCK_CNT_LO
Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_plllock_cnt_lo plllock_cnt[15:0]

PMU_PLLLOCK_CNT_HI
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_plllock_cnt_hi plllock_cnt[19:16]

PMU_PLLRST_CNT_LO
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_pllrst_cnt_lo pllrst_cnt[15:0]

PMU_PLLRST_CNT_HI
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_pllrst_cnt_hi pllrst_cnt[19:16]

PMU_STABLE_CNT_LO
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_stable_cnt_lo stable_cnt[15:0]

PMU_STABLE_CNT_HI
Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_stable_cnt_hi stable_cnt[19:16]

PMU_WAKEUP_RST_CLR_CNT_HI
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_wakeup_rst_cnt_hi wakeuprst_cnt[19:16]

PMU_WAKEUP_RST_CLR_CNT_LO
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_wakeup_rst_cnt_lo wakeuprst_cnt[15:0]

PMU_DDR_SREF_ST
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	upctl_c_sysack upctl c_sysack status
0	RO	0x0	upctl_c_active upctl c_active status

PMU_SYS_REG0_LO
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg0_lo sysreg0[15:0]

PMU_SYS_REG0_HI
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg0_hi sysreg0[31:16]

PMU_SYS_REG1_LO
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg1_lo sysreg1[15:0]

PMU_SYS_REG1_HI
Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg1_hi sysreg1[31:16]

PMU_SYS_REG2_LO
Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg2_lo sysreg2[15:0]

PMU_SYS_REG2_HI
Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg2_hi sysreg2[31:16]

PMU_SYS_REG3_LO
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg3_lo sysreg3[15:0]

PMU_SYS_REG3_HI
Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16bit write mask for lsb
15:0	RW	0x0000	pmu_sys_reg3_hi sysreg3[31:16]

PMU_SCU_PWRDN_CNT_LO
Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_scu_pwrdn_cnt_lo scu_pwrdn_cnt[15:0]

PMU_SCU_PWRDN_CNT_HI
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_scu_pwrdn_cnt_hi scu_pwrdn_cnt[19:16]

PMU_SCU_PWRUP_CNT_LO
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_scu_pwrdn_cnt_lo scu_pwrdn_cnt[15:0]

PMU_SCU_PWRUP_CNT_HI
Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_scu_pwrdn_cnt_hi scu_pwrdn_cnt[19:16]

PMU_TIMEOUT_CNT_LO
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_timeout_cnt_lo timeout_cnt[15:0]

PMU_TIMEOUT_CNT_HI
Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_timeout_cnt_hi timeout_cnt[19:16]

PMU_CPU0APM_CON
Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu0_sft_wakeup software wakeup cpu0 when auto power down mode
2	RW	0x0	global_int_disable0_cfg disable interrupt to cpu0
1	RW	0x0	cpu0_int_wakeup_en 1: cpu0 auto power down interrupt wakeup enable
0	RW	0x0	cpu0_wfi_pwrdn_en 1: enable cpu0 wfi auto power down

PMU_CPU1APM_CON
Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu1_sft_wakeup software wakeup cpu1 when auto power down mode
2	RW	0x0	global_int_disable1_cfg disable interrupt to cpu1
1	RW	0x0	cpu1_int_wakeup_en 1: cpu1 auto power down interrupt wakeup enable
0	RW	0x0	cpu1_wfi_pwrdn_en 1: enable cpu1 wfi auto power down

PMU_CPU2APM_CON
Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu2_sft_wakeup software wakeup cpu2 when auto power down mode
2	RW	0x0	global_int_disable2_cfg disable interrupt to cpu2
1	RW	0x0	cpu2_int_wakeup_en 1: cpu2 auto power down interrupt wakeup enable
0	RW	0x0	cpu2_wfi_pwrdn_en 1: enable cpu2 wfi auto power down

PMU_CPU3APM_CON
Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu3_sft_wakeup software wakeup cpu3 when auto power down mode
2	RW	0x0	global_int_disable3_cfg disable interrupt to cpu3
1	RW	0x0	cpu3_int_wakeup_en 1: cpu3 auto power down interrupt wakeup enable
0	RW	0x0	cpu3_wfi_pwrdn_en 1: enable cpu0 wfi auto power down

8.5 Timing Diagram

8.5.1 Each domain power switch timing

The following figure is the each domain power down and power up timing.

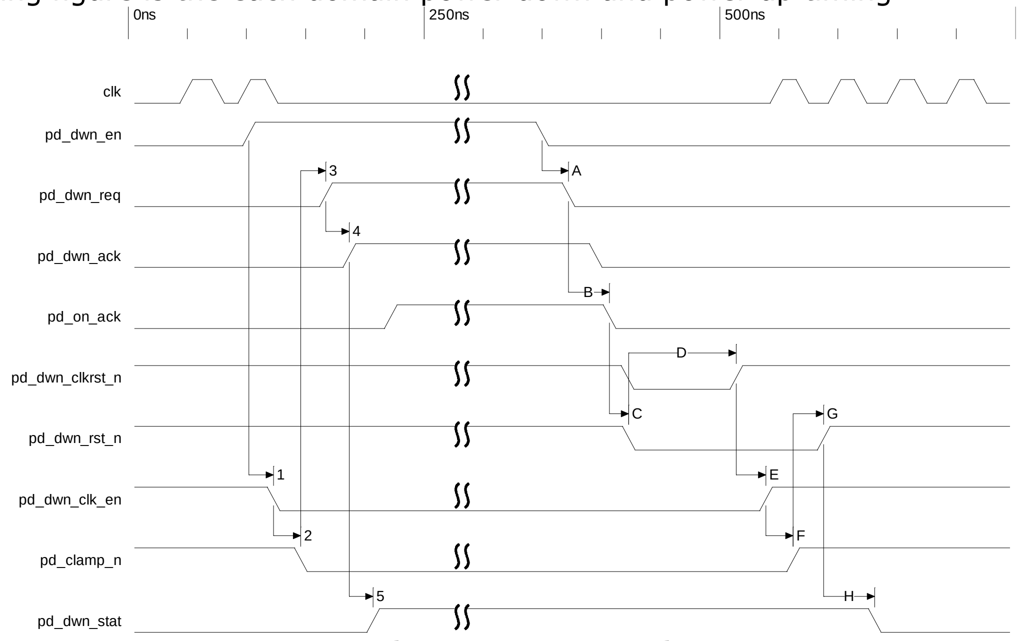


Fig. 8-3 Each Domain Power Switch Timing

8.5.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV, SIM detect wakeup, GPIO0 wakeup source and so on. All these external wakeup sources must meet the timing requirement (at least 200us) when the wakeup event is asserted. The following figure gives the timing information.

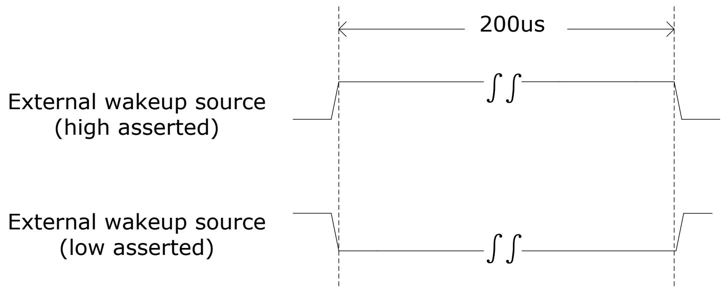

200us
External wakeup source (high asserted)
External wakeup source (low asserted)

Fig. 4-6 External Wakeup Source PAD Timing

8.6 Application Note

8.6.1 Debug IO

PX30 provide PMU Debug IO for FSM observation. Each IO corresponding with a bit of PMU power states [4:0].

Table 8-2 Low Power State

Module Pin	Direction	Pad Name	IOMUX Setting
power_state[0]	O	IO_UART0tx_PMUdebug0_GPIO0B2pmuio2	PMUGRF_GPIO0B_IOMUX[5:4]=2’b10
power_state[1]	O	IO_UART0rx_PMUdebug1_GPIO0B3pmuio2	PMUGRF_GPIO0B_IOMUX[7:6]=2’b10
power_state[2]	O	IO_UART0cts_PMUdebug2_PMUdebug_sout_GPIO0B4pmuio2	PMUGRF_GPIO0B_IOMUX[9:8]=2’b10
power_state[3]	O	IO_PWM1_UART3txm0_PMUdebug3_GPIO0C0pmuio2	PMUGRF_GPIO0C_IOMUX[1:0]=2’b11

Module Pin	Direction	Pad Name	IOMUX Setting
power_state[4]	O	IO_PWM3_UART3rxm0_PMUdebug4_GPIO0C1pmuio2	PMUGRF_GPIO0C_IOMUX[3:2]=2’b11
power_state[4]	O	IO_I2C1scl_UART3ctsm0_PMUdebug5_GPIO0C2pmuio2	PMUGRF_GPIO0C_IOMUX[5:4]=2’b11
debug_Sout	O	IO_UART0cts_PMUdebug2_PMUdebug_sout_GPIO0B4pmuio2	PMUGRF_GPIO0B_IOMUX[9:8]=2’b11

Chapter 9 Pulse Width Modulation (PWM)

9.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.
The PWM Module supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
  - Measures the high/low polarity effective cycles of this input waveform
  - Generates a single interrupt at the transition of input waveform polarity
  - 32-bit high polarity capture register
  - 32-bit low polarity capture register
  - 32-bit current value register
  - The capture result can be stored in a FIFO, and the depth of FIFO is 8. The data of FIFO can be read by CPU or DMA
  - Support 32-bit power key capture mode
  - Support a input filter to remove glitch
- Configurable to operate in continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - Configurable PWM output polarity in inactive state and duty period pulse polarity
  - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
  - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
  - 8-bit repeat counter for one-shot operation. One-shot operation will produce N + 1 periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
  - Continuous mode generates the waveform continuously, and does not generates any interrupts
- pre-scaled operation to clk_pwm and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

9.2 Block Diagram

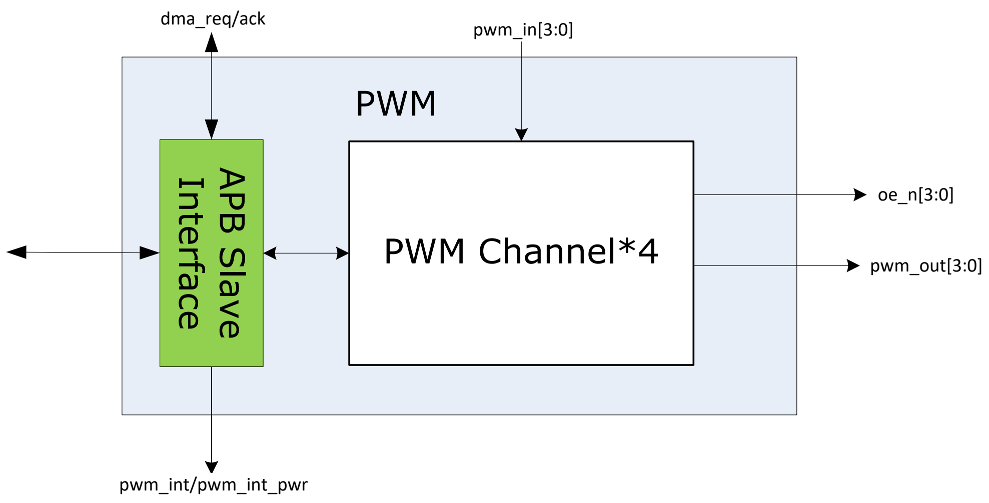


Fig. 9-1PWM Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface with 32-bit bus width, and asserts the active-high level interrupt. PWM only supports one interrupt output, please refer to interrupt register to know the raw interrupt status when an

interrupt is asserted.
PWM Channel is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

9.3 Function Description

The PWM supports three operation modes: capture mode, one-shot mode and continuous mode. For the one-shot mode and the continuous mode, the PWM output can be configured as the left-aligned mode or the center-aligned mode.

9.3.1 Capture mode

The capture mode is used to measure the PWM channel input waveform high/low effective cycles with the PWM channel clock, and asserts an interrupt when the polarity of the input waveform changes. The number of the high effective cycles is recorded in the PWMx_PERIOD_HPC register, while the number of the low effective cycles is recorded in the PWMx_DUTY_LPC register.
Notes: the PWM input waveform is doubled buffered when the PWM channel isworking in order to filter unexpected shot-time polarity transition, and therefore the interrupt is asserted several cycles after the input waveform polarity changes, and so does the change of the values of PWMx_PERIOD_HPC and PWMx_DUTY_LPC.

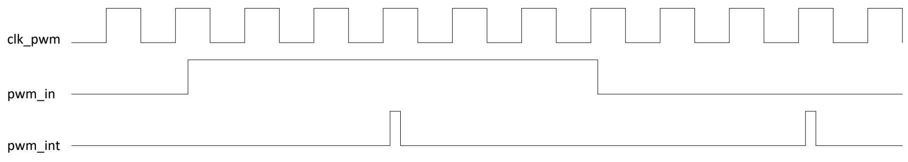


Fig. 9-2PWMCapture Mode

The capture result also can be stored in a FIFO. The FIFO has an almost full indicator. The indicator can chose to use as an interrupt or DMA request. When it is used as an interrupt, the data in FIFO can be read by CPU. When it is used as a DMA request, the data in FIFO can be read through DMA. It also supports timeout interrupt when the data in FIFO has not been read in a time threshold.
The PWM support 32-bit power key capture mode. User can set 10 power key to match, the interrupt will be asserted when the capture value match any one.

9.3.2 Continuous mode

The PWM channel generates a series of the pulses continuously as expected once the channel is enabled with continuous mode.
In the continuous mode, the PWM output waveforms can be in one form of the two output mode: left-aligned mode or center-aligned mode.
For the left-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. After the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

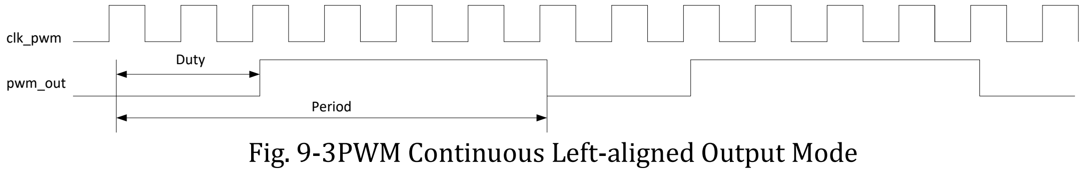


Fig. 9-3PWM Continuous Left-aligned Output Mode

For the center-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once one half of duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. Then if there is one half of duty cycle left for the whole period,the output is again switched to the opposite polarity. Finally after the period number (PWMx_PERIOD_HPC) is reached, the output starts another period of desired pulse.

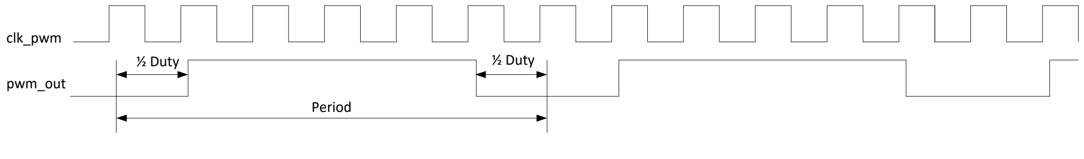


Fig. 9-4PWM Continuous Center-aligned Output Mode

Once disable the PWM channel, the channel stops generating the output waveforms and output polarity is fixed as the configured inactive polarity (PWMx_CTRL.inactive_pol).

9.3.3 One-shot mode

Unlike the continuous mode, the PWM channel generates the output waveforms within the configured periods (PWM_CTRL.rpt + 1), and then stops. At the same times, an interrupt is asserted to inform that the operation has been finished.
There are also two output modes for the one-shot mode: the left-aligned mode and the center-aligned mode.

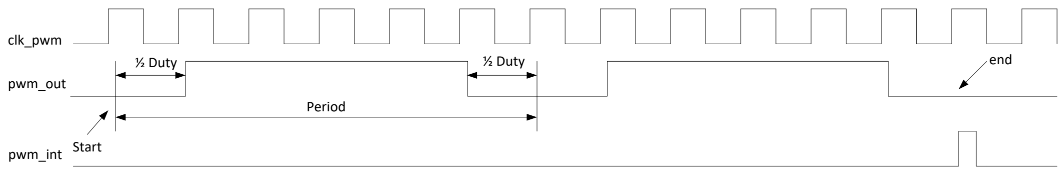


Fig. 9-5PWM One-shot Center-aligned Output Mode

9.4 Register Description

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register.
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register.
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register.
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register

Name	Offset	Size	Reset Value	Description
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003c	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register
PWM_FIFO_CTRL	0x0050	W	0x00000000	PWM Channel 3 FIFO Mode Control Register
PWM_FIFO_INTSTS	0x0054	W	0x00000010	FIFO Interrupts Status register
PWM_FIFO_TOUTTHR	0x0058	W	0x00000000	FIFO Timeout Threshold Register
PWM_FIFO	0x0060	W	0x00000000	FIFO Register
PWM_PWRMATCH_CTRL	0x0080	W	0x00000000	PWM power key match control
PWM_PWRMATCH_LPRE	0x0084	W	0x238c22c4	PWM power key match of low preload
PWM_PWRMATCH_HPRE	0x0088	W	0x11f81130	PWM power key match of high preload
PWM_PWRMATCH_LD	0x008c	W	0x029401cc	PWM power key match of low data
PWM_PWRMATCH_HD_ZERO	0x0090	W	0x029401cc	PWM power key match of high data for zero
PWM_PWRMATCH_HD_ONE	0x0094	W	0x06fe0636	PWM power key match of high data for one
PWM_PWRMATCH_VALUE0	0x0098	W	0x00000000	PWM power key match value 0
PWM_PWRMATCH_VALUE1	0x009c	W	0x00000000	PWM power key match value 1
PWM_PWRMATCH_VALUE2	0x00a0	W	0x00000000	PWM power key match value 2
PWM_PWRMATCH_VALUE3	0x00a4	W	0x00000000	PWM power key match value 3
PWM_PWRMATCH_VALUE4	0x00a8	W	0x00000000	PWM power key match value 4
PWM_PWRMATCH_VALUE5	0x00ac	W	0x00000000	PWM power key match value 5
PWM_PWRMATCH_VALUE6	0x00b0	W	0x00000000	PWM power key match value 6
PWM_PWRMATCH_VALUE7	0x00b4	W	0x00000000	PWM power key match value 7
PWM_PWRMATCH_VALUE8	0x00b8	W	0x00000000	PWM power key match value 8

Name	Offset	Size	Reset Value	Description
PWM_PWRMATCH_VALUE9	0x00bc	W	0x00000000	PWM power key match value 9
PWM_PWM0_PWRCAPTURE_VALUE	0x00c0	W	0x00000000	PWM Channel 0 power key capture value
PWM_PWM1_PWRCAPTURE_VALUE	0x00c4	W	0x00000000	PWM channel 1 power key capture value
PWM_PWM2_PWRCAPTURE_VALUE	0x00c8	W	0x00000000	PWM channel 2 power key capture value
PWM_PWM3_PWRCAPTURE_VALUE	0x00cc	W	0x00000000	PWM channel 3 power key capture value
PWM_FILTER_CTRL	0x00d0	W	0x00000000	PWM input filter control

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

9.4.2 Detail Register Description

PWM_PWM0_CNT
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter. The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM0_PERIOD_HPR
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM0_DUTY_LPR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM0_CTRL
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter. This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods
23:16	RW	0x00	scale Scale Factor. This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256)
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor. This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select. 1'b0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1'b1: scaled clock is selected as PWM clock source
8	RW	0x0	force_clk_en Force clock Enable 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.

Bit	Attr	Reset Value	Description
7	RW	0x0	ch_cnt_en Enable to read PWM Channel Counter Register 0: disabled 1: enabled
6	RW	0x0	conlock PWM configure lock. PWM period and duty lock to previous configuration. 1'b0: disable lock 1'b1: enable lock
5	RW	0x0	output_mode PWM Output Mode. 1'b0: left aligned mode 1'b1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity. This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: negative 1'b1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity. This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: negative 1'b1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode. 2'b00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt . 2'b01: Continuous mode. PWM produces the waveform continuously 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: reserved
0	RW	0x0	pwm_en PWM channel enable. 1'b0: disabled 1'b1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM1_CNT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CNT Timer Counter. The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock.The value ranges from 0 to (2^32-1)

PWM_PWM1_PERIOD_HPR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM1_DUTY_LPR
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM1_CTRL
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter. This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods
23:16	RW	0x00	scale Scale Factor. This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256)
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor. This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select. 1'b0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1'b1: scaled clock is selected as PWM clock source
8	RW	0x0	force_clk_en Force clock Enable 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en Enable to read PWM Channel Counter Register 0: disabled 1: enabled
6	RW	0x0	conlock PWM configure lock. pwm period and duty lock to previous configuration 1'b0: disable lock 1'b1: enable lock
5	RW	0x0	output_mode PWM Output Mode. 1'b0: left aligned mode 1'b1: center aligned mode

Bit	Attr	Reset Value	Description
4	RW	0x0	inactive_pol Inactive State Output Polarity. This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: negative 1'b1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity. This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: negative 1'b1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode. 2'b00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 2'b01: Continuous mode. PWM produces the waveform continuously 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: reserved
0	RW	0x0	pwm_en PWM channel enable. 1'b0: disabled 1'b1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM2_CNT
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter. The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM2_PERIOD_HPR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM2_DUTY_LPR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM2_CTRL
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter. This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods
23:16	RW	0x00	scale Scale Factor. This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256)
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	prescale Prescale Factor. This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select. 1'b0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1'b1: scaled clock is selected as PWM clock source
8	RW	0x0	force_clk_en Force clock Enable 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en Enable to read PWM Channel Counter Register 0: disabled 1: enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 1'b0: disable lock 1'b1: enable lock
5	RW	0x0	output_mode PWM Output mode. 1'b0: left aligned mode 1'b1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity. This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: negative 1'b1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity. This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: negative 1'b1: positive

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode PWM Operation Mode 2'b00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 2'b01: Continuous mode. PWM produces the waveform continuously 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: reserved
0	RW	0x0	pwm_en PWM channel enable 1'b0: disabled 1'b1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM3_CNT
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CNT Timer Counter. The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM3_PERIOD_HPR
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM3_DUTY_LPR
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle. If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1)

PWM_PWM3_CTRL
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter. This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods
23:16	RW	0x00	scale Scale Factor. This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256)
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor. This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select. 1'b0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1'b1: scaled clock is selected as PWM clock source
8	RW	0x0	force_clk_en Force clock Enable 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.

Bit	Attr	Reset Value	Description
7	RW	0x0	ch_cnt_en Enable to read PWM Channel Counter Register 0: disabled 1: enabled
6	RW	0x0	conlock PWM configure lock. PWM period and duty lock to previous configuration 1'b0: disable lock 1'b1: enable lock
5	RW	0x0	output_mode PWM Output mode. 1'b0: left aligned mode 1'b1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity. This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: negative 1'b1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity. This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: negative 1'b1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode. 2'b00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 2'b01: Continuous mode. PWM produces the waveform continuously 2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 2'b11: reserved
0	RW	0x0	pwm_en PWM channel enable. 1'b0: disabled 1'b1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_INTSTS
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	CH3_Pol Channel 3 Interrupt Polarity Flag. This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit
10	RO	0x0	CH2_Pol Channel 2 Interrupt Polarity Flag. This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit
9	RO	0x0	CH1_Pol Channel 1 Interrupt Polarity Flag. This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit
8	RO	0x0	CH0_Pol Channel 0 Interrupt Polarity Flag. This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit
7	RW	0x0	CH3_pwr_IntSts Channel 3 Raw power key Interrupt Status.. 1'b0: Channel 3 power key Interrupt not generated 1'b1: Channel 3 power key Interrupt generated
6	W1C	0x0	CH2_pwr_IntSts Channel 2 Raw power key Interrupt Status.. 1'b0: Channel 2 power key Interrupt not generated 1'b1: Channel 2 power key Interrupt generated

Bit	Attr	Reset Value	Description
5	W1C	0x0	CH1_pwr_IntSts Channel 1 Raw power key Interrupt Status.. 1'b0: Channel 1 power keyInterrupt not generated 1'b1: Channel 1 power key Interrupt generated
4	W1C	0x0	CH0_pwr_IntSts Channel 0 Raw power key Interrupt Status. 1'b0: Channel 0 power key Interrupt not generated 1'b1: Channel 0 power key Interrupt generated
3	R/WSC	0x0	CH3_IntSts Channel 3 Raw Interrupt Status. 1'b0: Channel 3 Interrupt not generated 1'b1: Channel 3 Interrupt generated
2	W1C	0x0	CH2_IntSts Channel 2 Raw Interrupt Status. 1'b0: Channel 2 Interrupt not generated 1'b1: Channel 2 Interrupt generated
1	W1C	0x0	CH1_IntSts Channel 1 Raw Interrupt Status. 1'b0: Channel 1 Interrupt not generated 1'b1: Channel 1 Interrupt generated
0	W1C	0x0	CH0_IntSts Channel 0 Raw Interrupt Status. 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated

PWM_INT_EN
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	CH3_pwr_Int_en Channel 3 Power Key Interrupt Enable. 1'b0: Channel 3 power key Interrupt disabled 1'b1: Channel 3 power key Interrupt enabled
6	RW	0x0	CH2_pwr_Int_en Channel 2 Power Key Interrupt Enable. 1'b0: Channel 2 power key Interrupt disabled 1'b1: Channel 2 power key Interrupt enabled
5	RW	0x0	CH1_pwr_Int_en Channel 1 Power Key Interrupt Enable. 1'b0: Channel 1 power key Interrupt disabled 1'b1: Channel 1 power key Interrupt enabled
4	RW	0x0	CH0_pwr_Int_en Channel 0 Power Key Interrupt Enable. 1'b0: Channel 0 power key Interrupt disabled 1'b1: Channel 0 power key Interrupt enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	CH3_Int_en Channel 3 Interrupt Enable. 1'b0: Channel 3 Interrupt disabled 1'b1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en Channel 2 Interrupt Enable. 1'b0: Channel 2 Interrupt disabled 1'b1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en Channel 1 Interrupt Enable. 1'b0: Channel 1 Interrupt disabled 1'b1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en Channel 0 Interrupt Enable. 1'b0: Channel 0 Interrupt disabled 1'b1: Channel 0 Interrupt enabled

PWM_FIFO_CTRL
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:12	RW	0x0	dma_ch_sel DMA channel select. 2'b00: Select PWM0 2'b01: Select PWM1 2'b10: Select PWM2 2'b11: Select PWM3
11	RO	0x0	reserved
10	RW	0x0	dma_ch_sel_en DMA channel select enable. 1'b1: Enable, use dma_ch_sel to select the channel to FIFO mode and DMA mode. 1'b0: Disable, select the channel PWM3 to FIFO mode and DMA mode
9	RW	0x0	timeout_en Fifo timeout enable
8	RW	0x0	dma_mode_en DMA mode enable. 1'b1: enable 1'b0: disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost full Watermark level

Bit	Attr	Reset Value	Description
3	RW	0x0	watermark_int_en Watermark full interrupt
2	RW	0x0	overflow_int_en FIFO Overflow Interrupt Enable. When high, an interrupt asserts when the fifo overflow
1	RW	0x0	full_int_en FIFO Full Interrupt Enable. When high, an interrupt asserts when the FIFO is full
0	RW	0x0	fifo_mode_sel FIFO MODE Sel. When high, PWM FIFO mode is activated

PWM_FIFO_INTSTS
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x1	fifo_empty_status FIFO empty Status. This bit indicates the FIFO is empty
3	W1C	0x0	timieout_intsts Timeout interrupt
2	W1C	0x0	fifo_watermark_full_intsts FIFO Watermark Full Interrupt Status. This bit indicates the FIFO is Watermark Full
1	W1C	0x0	fifo_overflow_intsts FIFO Overflow Interrupt Status. This bit indicates the FIFO is overflow
0	W1C	0x0	fifo_full_intsts FIFO Full Interrupt Status. This bit indicates the FIFO is full

PWM_FIFO_TOUTTHR
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	timeout_threshold FIFO Timeout value(unit pwm clock)

PWM_FIFO
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	pol Polarity.This bit indicates the polarity of the lower 31-bit counter. 1'b0: Low 1'b1: High
30:0	RO	0x00000000	cycle_cnt High/Low Cycle Counter. This 31-bit counter indicates the effective cycles of high/low waveform

PWM_PWRMATCH_CTRL
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	CH3_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
14	RW	0x0	CH2_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
13	RW	0x0	CH1_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
12	RW	0x0	CH0_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
11	RW	0x0	CH3_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
10	RW	0x0	CH2_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
9	RW	0x0	CH1_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
8	RW	0x0	CH0_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
7	RW	0x0	CH3_pwrkey_polarity 1'b0: pwm in polarity is positive 1'b1: pwm in polarity is negative
6	RW	0x0	CH2_pwrkey_polarity 1'b0: pwm in polarity is positive 1'b1: pwm in polarity is negative

Bit	Attr	Reset Value	Description
5	RW	0x0	CH1_pwrkey_polarity 1'b0: pwm in polarity is positive 1'b1: pwm in polarity is negative
4	RW	0x0	CH0_pwrkey_polarity 1'b0: pwm in polarity is positive 1'b1: pwm in polarity is negative
3	RW	0x0	CH3_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
2	RW	0x0	CH2_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
1	RW	0x0	CH1_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	CH0_pwrkey_enable 1'b0: Disabled 1'b1: Enabled

PWM_PWRMATCH_LPRE
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x238c	cnt_max The maximum counter value
15:0	RW	0x22c4	cnt_min The minimum counter value

PWM_PWRMATCH_HPRE
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x11f8	cnt_max The maximum counter value
15:0	RW	0x1130	cnt_min The minimum counter value

PWM_PWRMATCH_LD
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM_PWRMATCH_HD_ZERO
Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM_PWRMATCH_HD_ONE
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x06fe	cnt_max The maximum counter value
15:0	RW	0x0636	cnt_min The minimum counter value

PWM_PWRMATCH_VALUE0
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE1
Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE2
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE3
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE4
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE5
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE6
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE7
Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE8
Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE9
Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWM0_PWRCAPTURE_VALUE
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_PWM1_PWRCAPTURE_VALUE
Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_PWM2_PWRCAPTURE_VALUE
Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_PWM3_PWRCAPTURE_VALUE
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_FILTER_CTRL
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:4	RW	0x000	filter_number Filter window number
3	RW	0x0	CH3_input_filter_enable 1'b0: Disabled 1'b1: Enabled
2	RW	0x0	CH2_input_filter_enable 1'b0: Disabled 1'b1: Enabled
1	RW	0x0	CH1_input_filter_enable 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	CH0_input_filter_enable 1'b0: Disabled 1'b1: Enabled

9.5 Interface Description

Table 9-1PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
PWM0	I/O	IO_PWM0_OTGdrv_GPIO0B7pmuio2	PMUGRF_GPIO0B_IOMUX[15:14]=2'b1
PWM1	I/O	IO_PWM1_UART3txm0_PMUdebug3_GPIO0C0pmuio2	PMUGRF_GPIO0C_IOMUX[1:0]=2'b1
PWM2	I/O	IO_PWM2_GPIO2B5vccio3	GRF_GPIO2B_IOMUX_H[6:4]=3'b1

Module Pin	Direction	Pad Name	IOMUX Setting
PWM3	I/O	IO_PWM3_UART3rxm0_PMUdebug4_GPIO0C1pmuio2	PMUGRF_GPIO0C_IOMUX[3:2]=2'b1
PWM4	I/O	IO_LCDCd14_I2S08ch_lrcktx_PWM4_GPIO3C2vccio4	GRF_GPIO3C_IOMUX_L[10:8]=3'b11
PWM5	I/O	IO_LCDCd15_I2S08ch_sclktx_PWM5_GPIO3C3vccio4	GRF_GPIO3C_IOMUX_L[14:12]=3'b11
PWM6	I/O	IO_LCDCd16_I2S08ch_sdo0_PWM6_GPIO3C4vccio4	GRF_GPIO3C_IOMUX_H[2:0]=3'b11
PWM7	I/O	IO_LCDCd17_I2S08ch_sdi0_PWM7_GPIO3C5vccio4	GRF_GPIO3C_IOMUX_H[6:4]=3'b11

Notes: I=input, O=output, I/O=input/output.

9.6 Application Notes

9.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation.
5. Set PWM_FILTER_CTRL.filter_number, then Enable the PWM_FILTER_CTRL.CHx_input_filter_enable(Optional).
6. Enable the channel by writing '1' to PWM_PWMx_CTRL.pwm_en bit to start the channel.
7. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWM_PWMx_PERIOD_HPC register to know the effective high cycles of input waveforms, otherwise turn to PWM_PWMx_DUTY_LPC register to know the effective low cycles.
8. Write '0' to PWM_PWMx_CTRL.pwm_en to disable the channel.

9.6.2 PWM Capture DMA Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Configure the channel 3 to work in the capture mode.
4. Configure the PWM_FIFO_CTRL.dma_mode_enand PWM_FIFO_CTRL.fifo_mode_sel to enable the DMA mode.Configure PWM_FIFO_CTRL.almost_full_watermark at appropriate value.
5. Configure DMAC_BUS to tansfer data from PWM to DDR.
6. Set PWM_FILTER_CTRL.filter_number, then Enable the PWM_FILTER_CTRL.CHx_input_filter_enable(Optional).
7. Enable the channel by writing '1' to PWM_PWMx_CTRL.pwm_en bit to start the channel.
8. When a dma_req is asserted, DMAC_BUS transfer the data of effective high cycles and low cycles of input waveforms to DDR.
9. Write '0' to PWM_PWMx_CTRL.pwm_en to disable the channel.

9.6.3 PWM Power key Capture Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel. The clock should be 1 Mhz after division.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.CHx_int_pwr to enable the interrupt generation.

5. Set the PWM_PWRMATCH_VALUE0~9 registers for the 10 power key match value.
6.Set max_cnt and min_cnt of follow register:
PWM_PWRMATCH_LPRE,PWM_PWRMATCH_HPRE, PWM_PWRMATCH_LD,
PWM_PWRMATCH_HD_ZERO, PWM_PWRMATCH_HD_ONE. It doesn’t need to set these registers when the default value can meet the requirement.
7.Set PWM_PWRMATCH_CTRL.CHx_pwrkey_polarity for the polarity of power key signal, the default value is 0. Enable the PWM_PWRMATCH_CTRL.CHx_pwrkey_enable.
8. Set PWM_FILTER_CTRL.filter_number, then Enable the
PWM_FILTER_CTRL.CHx_input_filter_enable(Optional).
9. Enable the channel by writing ‘1’ to PWM_PWMx_CTRL.pwm_en bit to start the channel.
10. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status, and refer to PWM_PWMx_PWRCAPTURE_VALUE to know the power key capture value.
11. Write ‘0’ to PWM_PWMx_CTRL.pwm_en to disable the channel.

9.6.4 PWM One-shot Mode/ContinuousStandard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to ‘0’ to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWM_PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWM_PWMx_CTRL.duty_pol and PWM_PWMx_CTRL.inactive_pol.
4. Set the PWM_PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode.
5. Configure the channel to work in the one-shot mode or the continuous mode.
6. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation if if the channel is desired to work in the one-shot mode.
7. If the channel is working in the one-shot mode, an interrupt is asserted after the end of operation, and the PWM_PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working in, write ‘0’ to PWM_PWMx_CTRL.pwm_en bit to disable the PWM channel.

9.6.5 Low-power UsageFlow

The default value of PWM_PWMx_CTRL.force_clk_en is ‘0’ which make the channel enter the low-power mode. In low-power mode, When the PWM channel is inactive, the clk_pwm to the clock prescale module is gated in order to reduce the power consumption. User can set PWM_PWMx_CTRL.force_clk_en to ‘1’ which will make the channel quit the low-power mode. After the setting, the clk_pwm to the clock prescale module is always enable.

9.6.6 Other notes

When the channel is active to produce waveforms, it is free to program the PWM_PWMx_PERIOD_HPC and PWM_PWMx_DUTY_LPC register. User can use PWM_PWMx_CTRL.conlock to take period and duty effect at the same time. The usage flow is as follow:
1. Set PWM_PWMx_CTRL.conlock to ‘1’.
2. Set PWM_PWMx_PERIOD_HPC and PWM_PWMx_DUTY_LPC.
3. Set PWM_PWMx_CTRL.conlock to ‘0’, the other bits in PWM_PWMx_CTRL should be appropriate.
After above configuration, the change will not take effect immediately until the current period ends.
An active channel can be changed to another operation mode without disable the PWM channel. However, during the transition of the operation mode there may be some irregular output waveforms. So does changing the clock division factor when the channel is active.

Chapter 10 Generic Interrupt Controller (GIC)

10.1 Overview

There is a generic interrupt controller(GIC400) in PX30which generates physical interrupts to Cortex-A35. It has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A35. The details of CPU interface connectivity are shown in the following table.

Table 10-1CPU interface connectivity

CPU Interface Number	**Connectivity**
CPU interface 0	CPU0
CPU interface 1	CPU1
CPU interface 2	CPU2
CPU interface 3	CPU3

It supports the following features:
- Supports 128 hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A35 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

10.2 Block Diagram

The generic interrupt controller comprises with:

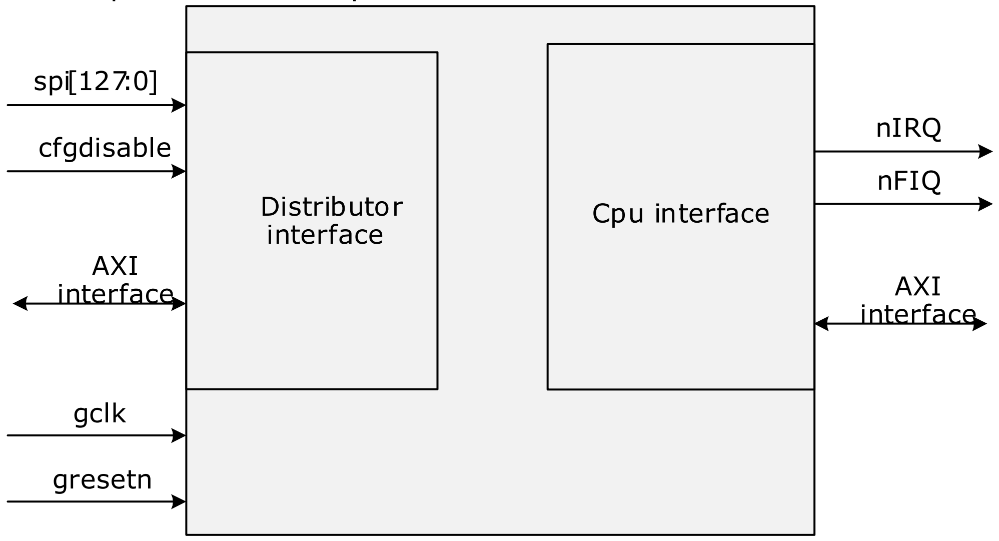


Fig. 10-1 Block Diagram

10.3 Function Description

Please refer to the document “IHI0048B_gic_architecture_specification.pdf” for the detailedfunction description.

Chapter 11 DMA Controller (DMAC)

11.1 Overview

This device supports 1 Direct Memory Access(DMA) Controllers.It (DMAC) supports transfers between memory and memory, peripheral and memory.DMACis under Non-secure state after reset, and the secure state can be changed by configurable SGRF module.
DMACsupports the following features:

- Supports Trustzone technology
- Supports 25 peripheral request
- Up to 64bits data size
- 8 channel at the same time
- Up to burst 16
- 16 interrupts output and 1 abort output
- Supports 128 MFIFO depth

Following table shows the DMACrequest mapping scheme.

Table 11-1DMAC Request Mapping Table

Req number	Source	Polarity
0	UART0_TX	High level
1	UART0_RX	High level
2	UART1_TX	High level
3	UART1_RX	High level
4	UART2_TX	High level
5	UART2_RX	High level
6	UART3_TX	High level
7	UART3_RX	High level
8	UART4_TX	High level
9	UART4_RX	High level
10	UART5_TX	High level
11	UART5_RX	High level
12	SPI0_TX	High level
13	SPI0_RX	High level
14	SPI1_TX	High level
15	SPI1_RX	High level
16	I2S0_8CH_TX	High level
17	I2S0_8CH_RX	High level
18	I2S1_2CH_TX	High level
19	I2S1_2CH_RX	High level
20	I2S2_8CH_TX	High level
21	I2S2_8CH_RX	High level
22	PWM0_TX	High level
23	PWM1_TX	High level
24	PDM	High level

DMAC supportincrementing-address burst and fixed-address burst. But in the case of access SPI and UART at byte or halfword size, DMAC only support fixed-address burst and the address must be aligned to word.

11.2 Block Diagram

Following figure shows the block diagram of DMAC.

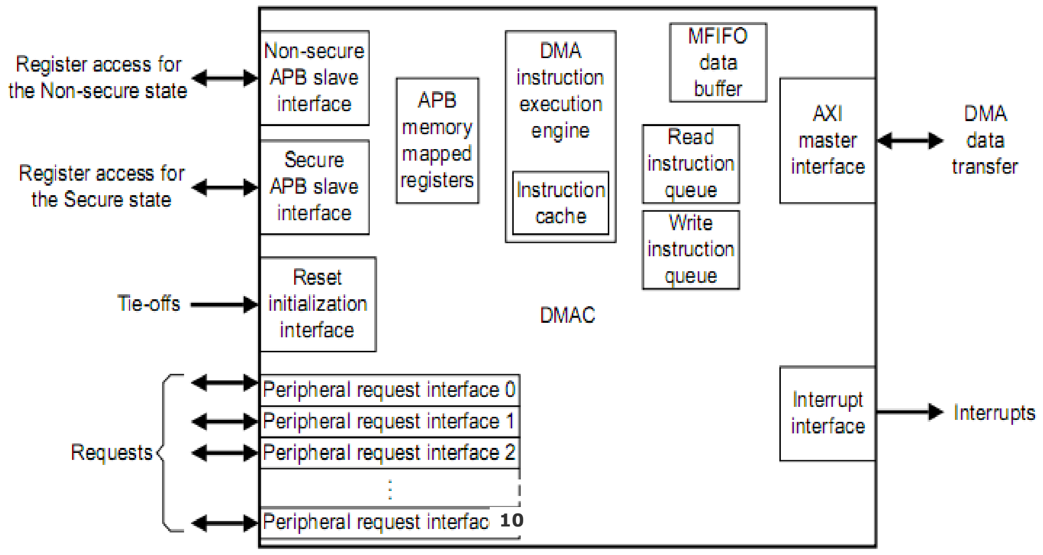

Register access for the Non-secure state
Register access for the Secure state
Tie-offs
Requests
Non-secure APB slave interface
Secure APB slave interface
Reset initialization interface
APB memory mapped registers
DMA instruction execution engine
Instruction cache
MFIFO data buffer
Read instruction queue
Write instruction queue
AXI master interface
DMA data transfer
DMAC
Peripheral request interface 0
Peripheral request interface 1
Peripheral request interface 2
Peripheral request interfac 10
Interrupt interface
Interrupts

Fig. 11-1Block diagram of DMAC

As the DMAC supports Trustzone technology,so dual APB interfaces enable the operation of the DMAC to be partitioned into the secure state and Non-secure state. You can use the APB interfaces to access status registers and also directly execute instructions in the DMAC. The default interface after reset is Non-secure apb interface.

11.3 Function Description

11.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache. It supports 8 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.
The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.
When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete.
When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

11.3.2 Operating states

Following figure shows the operating states for the DMA manager thread and DMA channel threads.

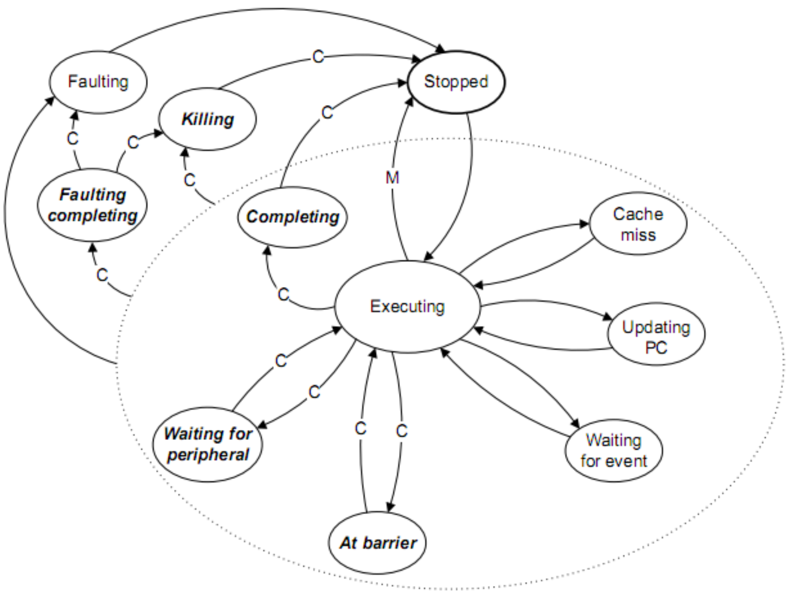


Fig. 11-2DMAC operation states

Notes:arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:
C DMA channel threads only.
M DMA manager thread only.

After the DMAC exits from reset, it sets all DMA channel threads to the stopped state, and DMA manager thread moves to the Stopped state.

11.4 Register Description

11.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

11.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
DMA_DSR	0x0000	W	0x00000000	DMA Manager Status Register
DMA_DPC	0x0004	W	0x00000000	DMA Program Counter Register
DMA_INTEN	0x0020	W	0x00000000	Interrupt Enable Register
DMA_EVENT_RIS	0x0024	W	0x00000000	Event-Interrupt Raw Status Register
DMA_INTMIS	0x0028	W	0x00000000	Interrupt Status Register
DMA_INTCLR	0x002c	W	0x00000000	Interrupt Clear Register
DMA_FSRD	0x0030	W	0x00000000	Fault Status DMA Manager Register
DMA_FSRC	0x0034	W	0x00000000	Fault Status DMA Channel Register
DMA_FTRD	0x0038	W	0x00000000	Fault Type DMA Manager Register
DMA_FTR0	0x0040	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel Register

Name	Offset	Size	Reset Value	Description
DMA_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel Register
DMA_CSR0	0x0100	W	0x00000000	Channel Status Registers
DMA_CPC0	0x0104	W	0x00000000	Channel Program Counter Registers
DMA_CSR1	0x0108	W	0x00000000	Channel Status Registers
DMA_CPC1	0x010c	W	0x00000000	Channel Program Counter Registers
DMA_CSR2	0x0110	W	0x00000000	Channel Status Registers
DMA_CPC2	0x0114	W	0x00000000	Channel Program Counter Registers
DMA_CSR3	0x0118	W	0x00000000	Channel Status Registers
DMA_CPC3	0x011c	W	0x00000000	Channel Program Counter Registers
DMA_CSR4	0x0120	W	0x00000000	Channel Status Registers
DMA_CPC4	0x0124	W	0x00000000	Channel Program Counter Registers
DMA_CSR5	0x0128	W	0x00000000	Channel Status Registers
DMA_CPC5	0x012c	W	0x00000000	Channel Program Counter Registers
DMA_SAR0	0x0400	W	0x00000000	Source Address Registers
DMA_DAR0	0x0404	W	0x00000000	DestinationAddress Registers
DMA_CCR0	0x0408	W	0x00000000	Channel Control Registers
DMA_LC0_0	0x040c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_0	0x0410	W	0x00000000	Loop Counter 1 Registers
DMA_SAR1	0x0420	W	0x00000000	Source Address Registers
DMA_DAR1	0x0424	W	0x00000000	DestinationAddress Registers
DMA_CCR1	0x0428	W	0x00000000	Channel Control Registers
DMA_LC0_1	0x042c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_1	0x0430	W	0x00000000	Loop Counter 1 Registers
DMA_SAR2	0x0440	W	0x00000000	Source Address Registers
DMA_DAR2	0x0444	W	0x00000000	DestinationAddress Registers
DMA_CCR2	0x0448	W	0x00000000	Channel Control Registers
DMA_LC0_2	0x044c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_2	0x0450	W	0x00000000	Loop Counter 1 Registers
DMA_SAR3	0x0460	W	0x00000000	Source Address Registers
DMA_DAR3	0x0464	W	0x00000000	DestinationAddress Registers
DMA_CCR3	0x0468	W	0x00000000	Channel Control Registers
DMA_LC0_3	0x046c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_3	0x0470	W	0x00000000	Loop Counter 1 Registers
DMA_SAR4	0x0480	W	0x00000000	Source Address Registers
DMA_DAR4	0x0484	W	0x00000000	DestinationAddress Registers

Name	Offset	Size	Reset Value	Description
DMA_CCR4	0x0488	W	0x00000000	Channel Control Registers
DMA_LC0_4	0x048c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_4	0x0490	W	0x00000000	Loop Counter 1 Registers
DMA_SAR5	0x04a0	W	0x00000000	Source Address Registers
DMA_DAR5	0x04a4	W	0x00000000	DestinationAddress Registers
DMA_CCR5	0x04a8	W	0x00000000	Channel Control Registers
DMA_LC0_5	0x04ac	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_5	0x04b0	W	0x00000000	Register0000 Description
DMA_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMA_DBGCMD	0x0d04	W	0x00000000	Debug Command Register
DMA_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMA_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMA_CR0	0x0e00	W	0x00047051	Configuration Register 0
DMA_CR1	0x0e04	W	0x00000057	Configuration Register 1
DMA_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMA_CR3	0x0e0c	W	0x00000000	Configuration Register 3
DMA_CR4	0x0e10	W	0x00000006	Configuration Register 4
DMA_CRDn	0x0e14	W	0x02094733	Configuration Register n
DMA_WD	0x0e80	W	0x00000000	DMA Watchdog Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

11.4.3 Detail Register Description

DMA_DSR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state

8:4	RO	0x00	b00000 = event[0]
			b00001 = event[1]
			b00010 = event[2]
			…
			b11111 = event[31]

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101-b1110 = reserved
			b1111 = Faulting

DMA_DPC
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA manager thread

DMA_INTEN
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request. Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interrupt request

DMA_EVENT_RIS
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Bit [N] = 0 Event N is inactive or irq[N] is LOW. Bit [N] = 1 Event N is active or irq[N] is HIGH

DMA_INTMIS
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW. Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH

DMA_INTCLR
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	Bit [N] = 0 The status of irq[N] does not change. Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change

DMA_FSRD
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	0 = the DMA manager thread is not in the Faulting state 1 = the DMA manager thread is in the Faulting state

DMA_FSRC

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Bit [N] = 0 No fault is present on DMA channel N. Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state

DMA_FTRD
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface
29:17	RO	0x0	reserved
16	RO	0x0	performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	0 = DMA manager has appropriate security to execute DMAWFE or DMASEV 1 = a DMA manager thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt
4	RO	0x0	0 = DMA manager has appropriate security to execute DMAGO 1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state
3:2	RO	0x0	reserved
1	RO	0x0	the configuration of the DMAC: 0 = valid operand 1 = invalid operand
0	RW	0x0	0 = defined instruction 1 = undefined instruction

DMA_FTR0
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort

Bit	Attr	Reset Value	Description
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort

Bit	Attr	Reset Value	Description
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_FTR1
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort

Bit	Attr	Reset Value	Description
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_FTR2
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort

Bit	Attr	Reset Value	Description
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_FTR3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort

Bit	Attr	Reset Value	Description
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort

Bit	Attr	Reset Value	Description
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_FTR4
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort

Bit	Attr	Reset Value	Description
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_FTR5
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31	RO	0x0	0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs
29:19	RO	0x0	reserved
18	RO	0x0	thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
16	RO	0x0	thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort
15:14	RO	0x0	reserved
13	RO	0x0	0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort
12	RO	0x0	DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort

Bit	Attr	Reset Value	Description
11:8	RO	0x0	reserved
7	RO	0x0	to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort
5	RO	0x0	0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt. This fault is a precise abort
4:2	RO	0x0	reserved
1	RO	0x0	valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort
0	RO	0x0	0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMA_CSR0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC0
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 0 thread

DMA_CSR1
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC1
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 1 thread

DMA_CSR2
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC2
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 2 thread

DMA_CSR3
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC3
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 3 thread

DMA_CSR4
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC4
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 4 thread

DMA_CSR5
Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31

3:0	RO	0x0	b0000 = Stopped
			b0001 = Executing
			b0010 = Cache miss
			b0011 = Updating PC
			b0100 = Waiting for event
			b0101 = At barrier
			b0110 = reserved
			b0111 = Waiting for peripheral
			b1000 = Killing
			b1001 = Completing
			b1010-b1101 = reserved
			b1110 = Faulting completing
			b1111 = Faulting

DMA_CPC5
Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 5 thread

DMA_SAR0
Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 0

DMA_DAR0
Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 0

DMA_CCR0
Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_0
Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_0
Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_SAR1
Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 1

DMA_DAR1
Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 1

DMA_CCR1
Address: Operational Base + offset (0x0428)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

Bit	Attr	Reset Value	Description
14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_1
Address: Operational Base + offset (0x042c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_1

Address: Operational Base + offset (0x0430)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_SAR2

Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 2

DMA_DAR2

Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 2

DMA_CCR2

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

Bit	Attr	Reset Value	Description

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

Bit	Attr	Reset Value	Description
0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_2
Address: Operational Base + offset (0x044c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_2
Address: Operational Base + offset (0x0450)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_SAR3
Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 3

DMA_DAR3
Address: Operational Base + offset (0x0464)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 3

DMA_CCR3
Address: Operational Base + offset (0x0468)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH

Bit	Attr	Reset Value	Description
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

Bit	Attr	Reset Value	Description

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_3
Address: Operational Base + offset (0x046c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_3
Address: Operational Base + offset (0x0470)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_SAR4
Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 4

DMA_DAR4
Address: Operational Base + offset (0x0484)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 4

DMA_CCR4
Address: Operational Base + offset (0x0488)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_4
Address: Operational Base + offset (0x048c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_4
Address: Operational Base + offset (0x0490)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_SAR5
Address: Operational Base + offset (0x04a0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 5

DMA_DAR5
Address: Operational Base + offset (0x04a4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destinationdata for DMA channel 5

DMA_CCR5
Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	RO	0x0	the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size

17:15	RO	0x0	b000 = writes 1 byte per beat
			b001 = writes 2 bytes per beat
			b010 = writes 4 bytes per beat
			b011 = writes 8 bytes per beat
			b100 = writes 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction
			is the product of dst_burst_len and dst_burst_size

Bit	Attr	Reset Value	Description
14	RO	0x0	0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH
13:11	RO	0x0	Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH
10:8	RO	0x0	Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH
7:4	RO	0x0	b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size

3:1	RO	0x0	b000 = reads 1 byte per beat
			b001 = reads 2 bytes per beat
			b010 = reads 4 bytes per beat
			b011 = reads 8 bytes per beat
			b100 = reads 16 bytes per beat
			b101-b111 = reserved.
			The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction
			is the product of src_burst_len and src_burst_size

0	RO	0x0	0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMA_LC0_5

Address: Operational Base + offset (0x04ac)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMA_LC1_5
Address: Operational Base + offset (0x04b0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMA_DBGSTATUS
Address: Operational Base + offset (0x0d00)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

1:0	RO	0x0	b00 = execute the instruction that the DBGINST [1:0] Registers contain
			b01 = reserved
			b10 = reserved
			b11 = reserved

DMA_DBGCMD
Address: Operational Base + offset (0x0d04)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

1:0	WO	0x0	b00 = execute the instruction that the DBGINST [1:0] Registers contain
			b01 = reserved
			b10 = reserved
			b11 = reserved

DMA_DBGINST0
Address: Operational Base + offset (0x0d08)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 1
23:16	WO	0x00	Instruction byte 0
15:11	RO	0x0	reserved

10:8	WO	0x0	b000 = DMA channel 0
			b001 = DMA channel 1
			b010 = DMA channel 2
			…
			b111 = DMA channel 7

7:1	RO	0x0	reserved
0	WO	0x0	0 = DMA manager thread 1 = DMA channel

DMA_DBGINST1
Address: Operational Base + offset (0x0d0c)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 5
23:16	WO	0x00	Instruction byte 4
15:8	WO	0x00	Instruction byte 3
7:0	WO	0x00	Instruction byte 2

DMA_CR0
Address: Operational Base + offset (0x0e00)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

21:17	RO	0x02	b00000 = 1 interrupt output, irq[0]
			b00001 = 2 interrupt outputs, irq[1:0]
			b00010 = 3 interrupt outputs, irq[2:0]
			.
			.
			.
			b11111 = 32 interrupt outputs, irq[31:0]

16:12	RO	0x07	b00000 = 1 peripheral request interface
			b00001 = 2 peripheral request interfaces
			b00010 = 3 peripheral request interfaces
			.
			.
			.
			b11111 = 32 peripheral request interfaces

11:7	RO	0x0	reserved

6:4	RO	0x5	b000 = 1 DMA channel
			b001 = 2 DMA channels
			b010 = 3 DMA channels
			.
			.
			.
			b111 = 8 DMA channels

3	RO	0x0	reserved
2	RO	0x0	0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH
1	RO	0x0	0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH
0	RO	0x1	0 = the DMAC does not provide a peripheral request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies

DMA_CR1
Address: Operational Base + offset (0x0e04)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

7:4	RO	0x5	b0000 = 1 i-cache line
			b0001 = 2 i-cache lines
			b0010 = 3 i-cache lines
			…
			b1111 = 16 i-cache lines

3	RO	0x0	reserved

2:0	RO	0x7	b000-b001 = reserved
			b010 = 4 bytes
			b011 = 8 bytes
			b100 = 16 bytes
			b101 = 32 bytes
			b110-b111 = reserved

DMA_CR2
Address: Operational Base + offset (0x0e08)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMA_CR3
Address: Operational Base + offset (0x0e0c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state

DMA_CR4
Address: Operational Base + offset (0x0e10)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000006	Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMA_CRDn
Address: Operational Base + offset (0x0e14)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

29:20	RO	0x020	b000000000 = 1 line
			b000000001 = 2 lines
			…
			b111111111 = 1024 lines

Bit	Attr	Reset Value	Description

19:16	RO	0x9	b0000 = 1 line
			b0001 = 2 lines
			.
			.
			.
			b1111 = 16 lines

15	RO	0x0	reserved

14:12	RO	0x4	b000 = 1
			b001 = 2
			…
			b111 = 8

11:8	RO	0x7	b0000 = 1 line
			b0001 = 2 lines
			…
			b1111 = 16 lines

7	RO	0x0	reserved

6:4	RO	0x3	b000 = 1
			b001 = 2
			…
			b111 = 8

3	RO	0x0	reserved

2:0	RO	0x3	b000 = reserved
			b001 = reserved
			b010 = 32-bit
			b011 = 64-bit
			b100 = 128-bit
			b101-b111 = reserved

DMA_WD
Address: Operational Base + offset (0x0e80)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH

11.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

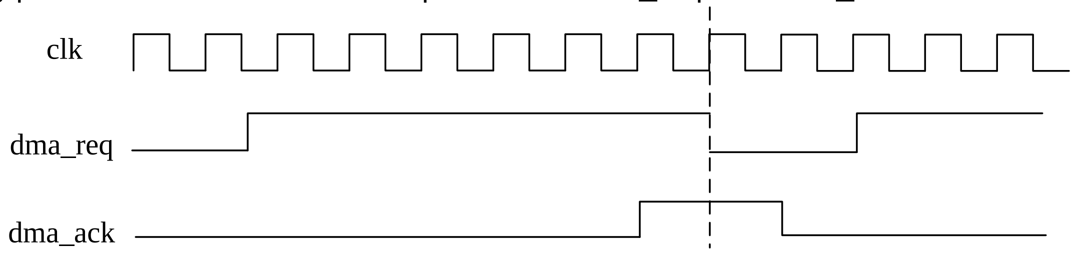


Fig.11-3DMAC request and acknowledge timing

11.6 Interface Description

DMAC has the following tie-off signals. It can be configured by SGRF register. (Please refer to the GRF chapter to find them)

Table 11-2DMAC boot interface

Interface	Reset value	Control source
boot_manager_ns	0x1	sgrf_dmac_con3[0]
boot_irq_ns	0xFFFF	sgrf_dmac_con0[15:0]
boot_periph_ns	0xFFFFFFFF	{sgrf_dmac_con2[15:0],sgrf_dmac_con1[15:0]}
grf_drtype_uart0_tx	0x1	sgrf_dmac_con4[1:0]
grf_drtype_uart0_rx	0x1	sgrf_dmac_con4[3:2]
grf_drtype_uart1_tx	0x1	sgrf_dmac_con4[5:4]
grf_drtype_uart1_rx	0x1	sgrf_dmac_con4[7:6]
grf_drtype_uart2_tx	0x1	sgrf_dmac_con4[9:8]
grf_drtype_uart2_rx	0x1	sgrf_dmac_con4[11:10]
grf_drtype_uart3_tx	0x1	sgrf_dmac_con4[13:12]
grf_drtype_uart3_rx	0x1	sgrf_dmac_con4[15:14]
grf_drtype_uart4_tx	0x1	sgrf_dmac_con5[1:0]
grf_drtype_uart4_rx	0x1	sgrf_dmac_con5[3:2]
grf_drtype_uart5_tx	0x1	sgrf_dmac_con5[5:4]
grf_drtype_uart5_rx	0x1	sgrf_dmac_con5[7:6]
grf_drtype_spi0_tx	0x1	sgrf_dmac_con5[9:8]
grf_drtype_spi0_rx	0x1	sgrf_dmac_con5[11:10]
grf_drtype_spi1_tx	0x1	sgrf_dmac_con5[13:12]
grf_drtype_spi1_rx	0x1	sgrf_dmac_con5[15:14]
grf_drtype_i2s0_8ch_tx	0x1	sgrf_dmac_con6[1:0]
grf_drtype_i2s0_8ch_rx	0x1	sgrf_dmac_con6[3:2]
grf_drtype_i2s1_2ch_tx	0x1	sgrf_dmac_con6[5:4]
grf_drtype_i2s1_2ch_rx	0x1	sgrf_dmac_con6[7:6]
grf_drtype_i2s2_8ch_tx	0x1	sgrf_dmac_con6[9:8]
grf_drtype_i2s2_8ch_rx	0x1	sgrf_dmac_con6[11:10]
grf_drtype_pwm0_tx	0x1	sgrf_dmac_con6[13:12]
grf_drtype_pwm1_tx	0x1	sgrf_dmac_con6[15:14]
grf_drtype_pdm	0x1	sgrf_dmac_con7[1:0]

boot_manager_ns
When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:
0 = assigns DMA manager to the Secure state
1 = assigns DMA manager to the Non-secure state.
boot_irq_ns
Controls the security state of an event-interrupt resource, when the DMAC exits from reset:
boot_irq_ns[x] is LOW
The DMAC assigns event<x> or irq[x] to the Secure state.
boot_irq_ns[x] is HIGH
The DMAC assigns event<x> or irq[x] to the Non-secure state.
boot_periph_ns

Controls the security state of a peripheral request interface, when the DMAC exits from reset:
boot_periph_ns[x] is LOW
The DMAC assigns peripheral request interface x to the Secure state.
boot_periph_ns[x] is HIGH
The DMAC assigns peripheral request interface x to the Non-secure state.
grf_drtype_<x>
The DMAC sets the state of the request_type flag:
grf_drtype_<x>[1:0]=b00: request_type<x> = Single.
grf_drtype_<x>[1:0]=b01: request_type<x> = Burst.

11.7 Application Notes

11.7.1 Using the APB slave interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.
The necessary steps to start a DMA channel thread using the debug instruction registers as following:
1. Create a program for the DMA channel.
2. Store the program in a region of system memory.
3. Poll the DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DBGINST0 Register and enter the:
- Instruction byte 0 encoding for DMAGO.
- Instruction byte 1 encoding for DMAGO.
- Debug thread bit to 0. This selects the DMA manager thread.

5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program, that was written to system memory in step 2.
6. Writing zero to the DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

11.7.2 Security usage

DMA manager thread is in the secure state
If the DNS bit is 0, the DMA manager thread operates in the secure state and it only performs secure instruction fetches. When a DMA manager thread in the secure state processes:
DMAGO
It uses the status of the ns bit, to set the security state of the DMA channel thread by writing to the CNS bit for that channel.
DMAWFE
It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.
DMASEV
It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.
DMA manager thread is in the Non-secure state
If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:
DMAGO
The DMAC uses the status of the ns bit, to control if it starts a DMA channel
thread. If:
ns = 0

The DMAC does not start a DMA channel thread and instead it:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
ns = 1
The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.
DMAWFE
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.
DMASEV
The DMAC uses the status of the corresponding INS bit, in the CR3Register, to control if it creates the event-interrupt. If:
INS = 0
The event-interrupt resource is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.
INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.
DMA channel thread is in the secure state
When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.
When a DMA channel thread in the secure state processes the following instructions:
DMAWFE
The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.
DMASEV
The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.
DMAWFP
The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
DMALDP, DMASTP
The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
DMAFLUSHP
The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses

DMA channel thread is in the Non-secure state
When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.
When a DMA channel thread in the Non-secure state processes the following instructions:
DMAWFE
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.
DMASEV
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event. If:
INS = 0
The event-interrupt resource is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers .
4. Moves the DMA channel to the Faulting completing state.
INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.
DMAWFP
The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it waits for the peripheral to signal a request. If:
PNS = 0
The peripheral is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1
The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.
DMALDP, DMASTP
The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends an acknowledgement to the peripheral. If:
PNS = 0
The peripheral is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1
The peripheral is in the Non-secure state. The DMAC sends a message to the peripheral to communicate when the data transfer is complete.
DMAFLUSHP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends a flush request to the peripheral. If:
PNS = 0
The peripheral is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1
The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.
When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:
1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

11.7.3 Programming restrictions

Fixed unaligned bursts
The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:
Unaligned read
- src_inc field is 0 in the CCRn Register
- the SARn Register contains an address that is not aligned to the size of data that the src_burst_size field contain

Unaligned write
- dst_inc field is 0 in the CCRn Register
- the DARn Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions
If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.
Updating DMA channel control registers during a DMA cycle restrictions
Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.
Resource sharing between DMA channels
DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

11.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.
Data corruption might occur if all of the following are true:
1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is splited across two lines in the data buffer (see Splitting data, below).
3. There is one idle cycle between the two read data beats.
4. Channel 0 performs an operation that updates channel control information during this idle

cycle (see Updates to channel control information, below)

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface need to be splited across two lines in the internal data buffer. This occurs when the read data beat contains data bytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.

Most applications of DMA-330 do not split data in this way, so are NOT vulnerable to data corruption from this defect.

The following cases are NOT vulnerable to data corruption because they do not split data:

- Byte lane offset between source and destination addresses is 0 when source and destination addresses have the same byte lane alignment, the offset is 0 and a wrap operation that splits data cannot occur.
- Byte lane offset between source and destination addresses is a multiple of source size

Table 11-3Source size in CCRn

Source size in CCRn	Allowed offset between SARn and DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

11.7.5 Interrupt shares between channel

As the DMAC does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help identify the interrupt source.

There are 4 likely information sources that can be used to determine the progress made by a program:

- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.

The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:

1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

11.7.6 Instruction sets

Table 11-4DMAC Instruction sets

Mnemonic	Instruction	Thread usage
DMAADDH	Add Halfword	C
DMAEND	End	M/C
DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C
DMAMOV	Move	C

DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wait For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

Notes:Thread usage: C=DMA channel, M=DMA manager

11.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled. For the other instructions, please refer to pl330_trm.pdf.

DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Registers and Destination Address Registers. The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two's complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register.
Following table shows the instruction encoding.

Table 11-5DMAC instruction encoding

Imm[15:8]	Imm[7:0]	0	1	0	1	1	1	ra	0

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>
where:
<address_register>
  Selects the address register to use. It must be either:
  SAR
    SARn Register and sets ra to 0.
  DAR
    DARn Register and sets ra to 1.
<16-bit immediate>
  The immediate value to be added to the <address_register>.

You should specify the 16-bit immediate as the number that is to be represented in the instruction encoding. For example, DMAADNH DAR, 0xFFF0 causes the value 0xFFFFFFF0 to be added to the current value of the Destination Address Register, effectively subtracting 16 from the DAR.
You can only use this instruction in a DMA channel thread.

Chapter 12 MAC Ethernet Interface

12.1 Overview

The MAC Ethernet Controller provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) compliant Ethernet PHY.
The MAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

12.1.1 Feature

- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
  - Supports CSMA/CD Protocol for half-duplex operation
  - Supports IEEE 802.3x flow control for full-duplex operation
  - Optional forwarding of received pause control frames to the user application in full-duplex operation
  - Back-pressure support for half-duplex operation
  - Automatic transmission of zero-quanta pause frame on de-assertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
  - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the MAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI Master interface
- Supports internal loopback on theRMII for debugging
- Debug status register that gives status of FSMs in Transmit and Receive data-paths and FIFO fill-levels.

12.2 Block Diagram

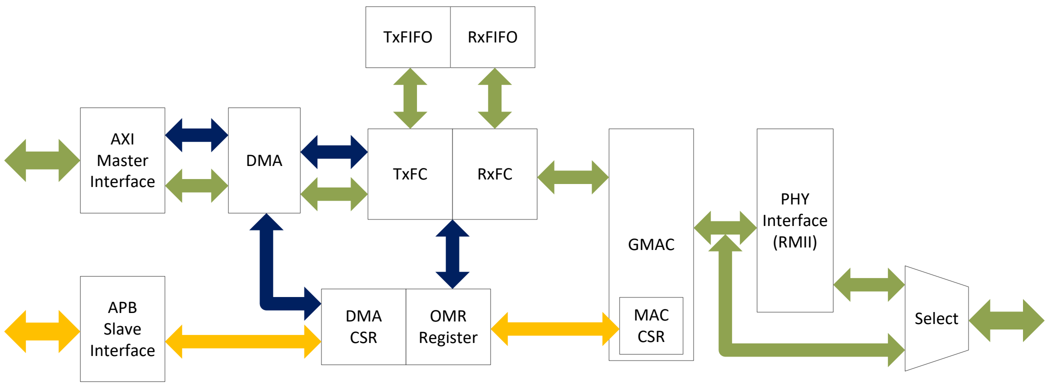


Fig.12-1 MACArchitecture

The MAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.
The MAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the MAC subsystem’s control and status registers (CSRs).
The MAC supports the PHY interfaces of reduced MII (RMII).
The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the MAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA.
These are asynchronous FIFOs, as they also transfer the data between the application clock and the MAC line clocks.

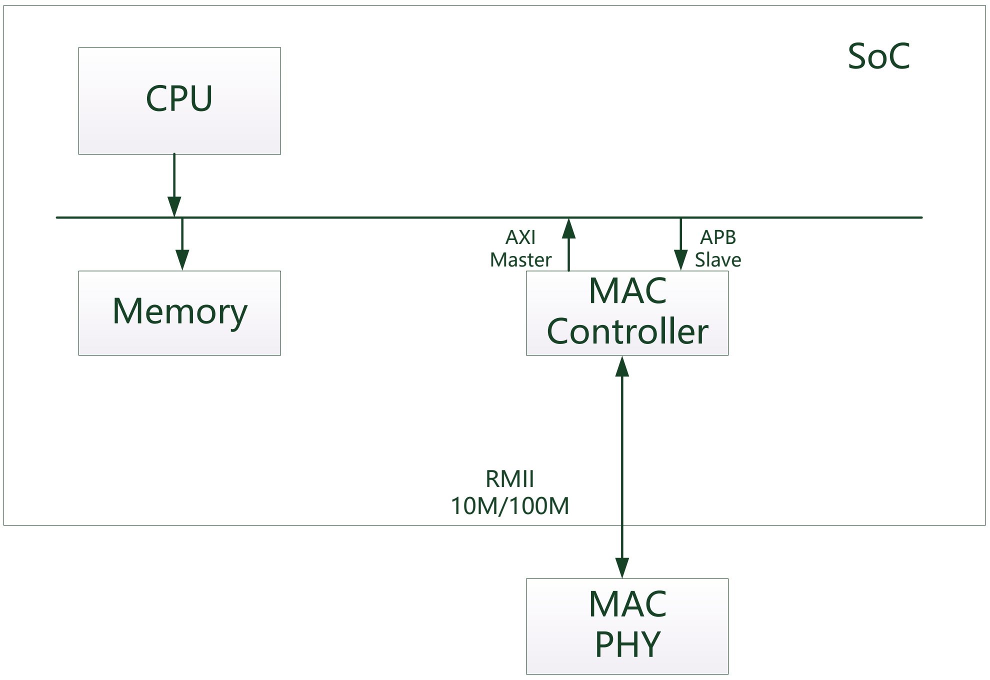


Fig.12-2 MAC Block Diagram

The MAC controller named MAC2IO:

- MAC2IO Supports 10/100-Mbps data transfer rates with the RMII interfaces

12.3 Function Description

12.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig. 1-3.

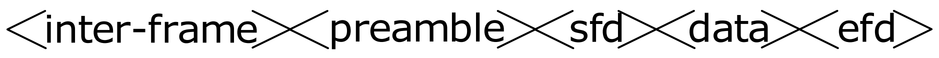


Fig. 12-3 MAC Frame Structure

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble.The bit value is 10101011.
The data in a well formed frame shall consist of N octet's data.

12.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the MAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
- Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

Transmit Bit Ordering
Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-4. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

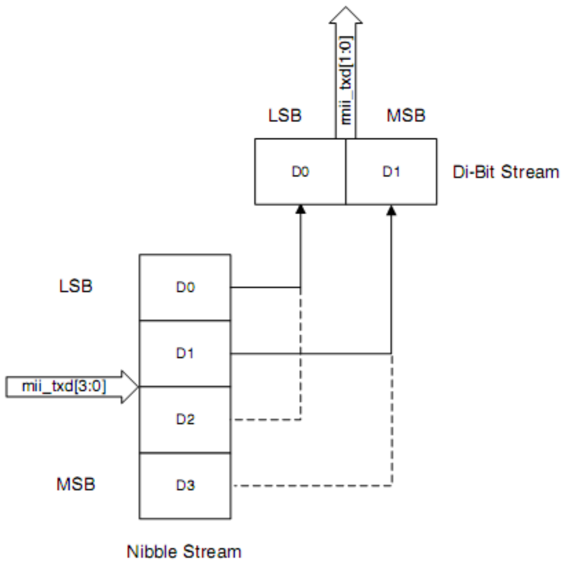


Fig. 12-4 RMII transmission bit ordering

RMII Transmit Timing Diagrams
Fig.1-5 through 1-8 show MII-to-RMII transaction timing.The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface.In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that TXD[1:0] may be sampled every 10th cycle,regard-less of the starting cycle within the group and yield the correct frame data.

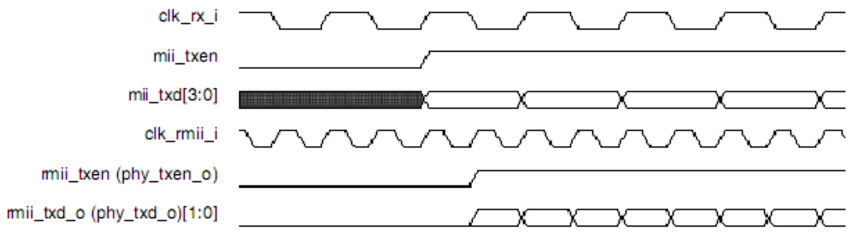


Fig. 12-5 Start of MII and RMII transmission in 100-Mbps mode

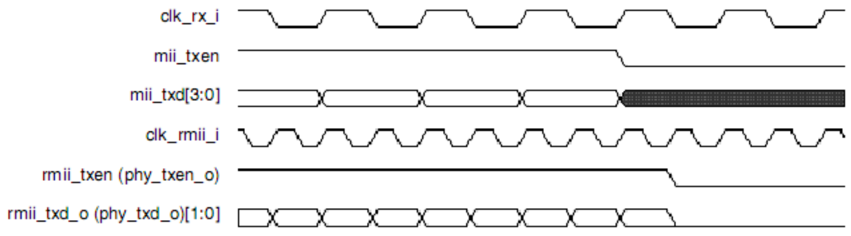


Fig. 12-6 End of MII and RMII Transmission in 100-Mbps Mode

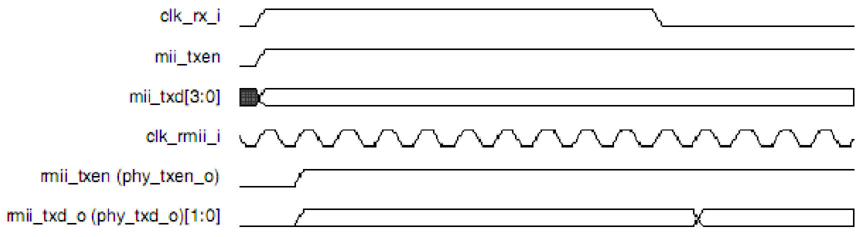


Fig. 12-7 Start of MII and RMII Transmission in 10-Mbps Mode

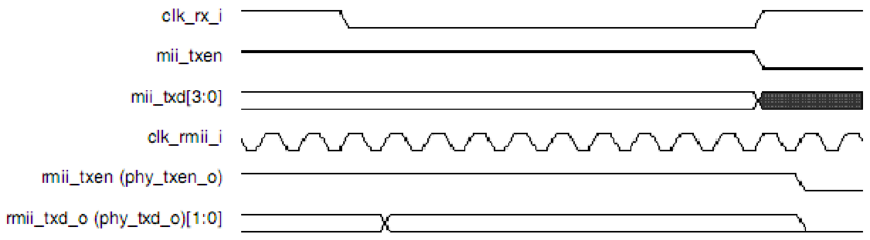


Fig. 12-8 End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-9. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

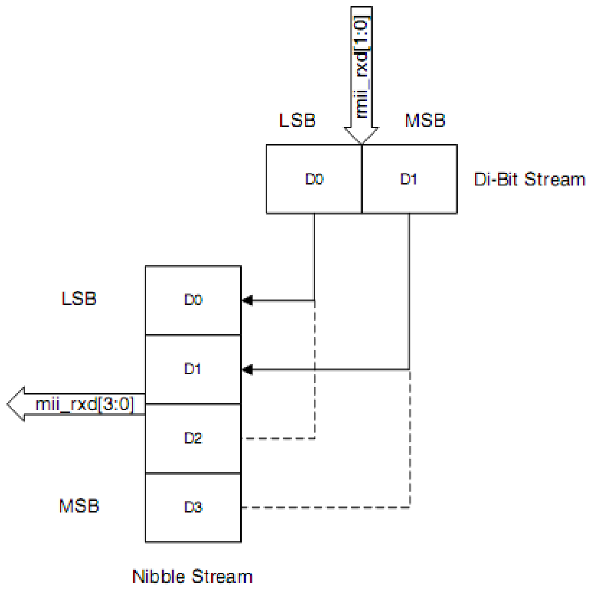


Fig. 12-9 RMII receive bit ordering

12.3.3 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the MAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.
The MAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_MAC. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	**pclk_MAC**	**MDC Clock**
0000	60-100 MHz	pclk_MAC/42
0001	100-150 MHz	pclk_MAC/62
0010	20-35 MHz	pclk_MAC/16
0011	35-60 MHz	pclk_MAC/26
0100	150-250 MHz	pclk_MAC/102
0101	250-300 MHz	pclk_MAC/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_MAC. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.
The frame structure on the MDIO line is shown below.

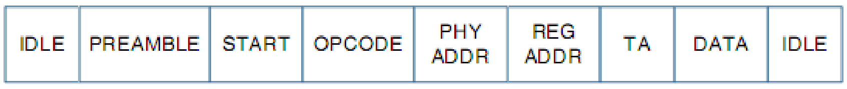


Fig. 12-10 MDIO frame structure

IDLE:	The mdio line is three-state; there is no clock on gmii_mdc_o
PREAMBLE:	32 continuous bits of value 1
START:	Start-of-frame is 2’b01
OPCODE:	2’b10 for read and 2’b01 for write
PHY ADDR:	5-bit address select for one of 32 PHYs
REG ADDR:	Register address in the selected PHY
TA:	Turnaround is 2’bZ0 for read and 2’b10 for Write
DATA:	Any 16-bit value. In a write operation, the MAC drives mdio; in a read operation, PHY drives it.

12.3.4 Power Management Block

Power management (PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts

for wake-up frames and Magic Packets received by the MAC. The PMT block sits on the receiver path of the MAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.
When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.
Remote Wake-Up Frame Detection
When the MAC is in sleep mode and the remote wake-up bit is enabled in register MAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register MAC_PMT_CTRL_STA.
PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received.
Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.
The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register MAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the MAC_PMT_CTRL_STA register to determine reception of a wake-up frame.
Magic Packet Detection
The Magic Packet frame is based on a method that uses Advanced Micro Device’s Magic Packet technology to power up the sleeping device on the network. The MAC receives a specific packet of information, called a Magic Packet, addressed to the node on the network. Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a MAC Address appearing 16 times.
The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register MAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48’hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48’hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48’hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the MAC address are detected.
If the MAC address of a node is 48'h00_11_22_33_44_55, then the MAC scans for the data sequence:
Destination Address Source Address ……………………………….. FF FFFFFFFFFF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC
Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

12.3.5 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.
The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

12.4 Register Description

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register This is the operation mode register for the MAC
MAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter Contains the frame filtering controls
MAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register Contains the higher 32 bits of the Multicast Hash table.This register is present only when the Hash filter function is selected in coreConsultant
MAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register Contains the lower 32 bits of the Multicast Hash table. This register is present only when the Hash filter function is selected in coreConsultant
MAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register Controls the management cycles to an external PHY
MAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register Contains the data to be written to or read from the PHY register
MAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register Controls the generation of control frames

Name	Offset	Size	Reset Value	Description
MAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register Identifies IEEE 802.1Q VLAN type frames
MAC_DEBUG	0x0024	W	0x00000000	Debug register This debug register gives the status of all the main modules of the transmit and receive data-paths and the FIFOs. An all-zero status indicates that the MAC core is in idle state (and FIFOs are empty) and no activity is going on in the data-paths
MAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register PMT Control and Status
MAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register Contains the interrupt status
MAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register Contains the masks for generating the interrupts
MAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register Contains the higher 16 bits of the first MAC address
MAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register Contains the lower 32 bits of the first MAC address
MAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register Enables and/or restarts auto-negotiation. It also enables PCS loopback
MAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register Indicates the link and auto-negotiation status
MAC_AN_ADV	0x00c8	W	0x000001e0	Auto Negotiation Advertisement Register This register is configured before auto-negotiation begins. It contains the advertised ability of the MAC

Name	Offset	Size	Reset Value	Description
MAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto Negotiation Link Partner Ability Register Contains the advertised ability of the link partner. Its value is valid after successful completion of auto-negotiation or when a new base page has been received (indicated in the Auto-Negotiation Expansion Register)
MAC_AN_EXP	0x00d0	W	0x00000000	Auto Negotiation Expansion Register Indicates whether a new base page has been received from the link partner
MAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register Indicates the status signals received from the PHY through the RGMII interface
MAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register The MMC Control register establishes the operating mode of the management counters
MAC_MMC_RX_INTR	0x0104	W	0x00000000	MMC Receive Interrupt Register The MMC Receive Interrupt register maintains the interrupts generated when the receive statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit

Name	Offset	Size	Reset Value	Description
MAC_MMC_TX_INTR	0x0108	W	0x00000000	MMC Transmit Interrupt Register The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Transmit Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit
MAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half their maximum value, and when they reach their maximum values
MAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when transmit statistic counters reach half their maximum value, and when they reach their maximum values
MAC_MMC_TXOCTETCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
MAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX OCTET Good and Bad Counter
MAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error

Name	Offset	Size	Reset Value	Description
MAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
MAC_MMC_TXOCTETCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
MAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
MAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
MAC_MMC_RXOCTETCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
MAC_MMC_RXOCTETCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter
MAC_MMC_RXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
MAC_MMC_RXCRCERR	0x0194	W	0x00000000	MMC RX Carrier
MAC_MMC_RXLENERR	0x01c8	W	0x00000000	MMC RX Length Error
MAC_MMC_RXFIFOOVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow
MAC_MMC_IPC_INT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register The MMC Receive Checksum Offload Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Offload) statistic counters reach half their maximum value , and when they reach their maximum values

Name	Offset	Size	Reset Value	Description
MAC_MMC_IPC_INTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register The MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counterís least-significant byte lane (bits[7:0]) must be read to clear the interrupt bit
MAC_MMC_RXIPV4GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
MAC_MMC_RXIPV4HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
MAC_MMC_RXIPV6GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
MAC_MMC_RXIPV6HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame
MAC_MMC_RXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
MAC_MMC_RXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
MAC_MMC_RXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
MAC_MMC_RXIPV4HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error
MAC_MMC_RXIPV6HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
MAC_MMC_RXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
MAC_MMC_RXTCPERROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
MAC_MMC_RXICMPERROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
MAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register

Name	Offset	Size	Reset Value	Description
MAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register Used by the host to instruct the DMA to poll the Transmit Descriptor List
MAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register Used by the Host to instruct the DMA to poll the Receive Descriptor list
MAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register Points the DMA to the start of the Receive Descriptor list
MAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register Points the DMA to the start of the Transmit Descriptor List
MAC_STATUS	0x1014	W	0x00000000	Status Register The Software driver (application) reads this register during interrupt service routine or polling to determine the status of the DMA
MAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register Establishes the Receive and Transmit operating modes and command
MAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register Enables the interrupts reported by the Status Register
MAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register Contains the counters for discarded frames because no host Receive Descriptor was available, and discarded frames because of Receive FIFO Overflow
MAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register Watchdog time-out for Receive Interrupt (RI) from DMA

Name	Offset	Size	Reset Value	Description
MAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register Controls AXI Master behavior (mainly controls burst splitting and number of outstanding requests)
MAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register Gives the idle status of the AXI master's read/write channels
MAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register Points to the start of current Transmit Descriptor read by the DMA
MAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register Points to the start of current Receive Descriptor read by the DMA
MAC_CUR_HOST_TX_BUF_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register Points to the current Transmit Buffer address read by the DMA
MAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Address Register Points to the current Receive Buffer address read by the DMA

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

12.4.2 Detail Register Description

MAC_MAC_CONF

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY

Bit	Attr	Reset Value	Description
23	RW	0x0	WD Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the MAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that
22	RW	0x0	JD Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission
21	RW	0x0	BE Frame Burst Enable When this bit is set, the MAC allows frame bursting during transmission in GMII Half-Duplex mode
20	RO	0x0	reserved
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)

Bit	Attr	Reset Value	Description
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps
13	RW	0x0	DO Disable Receive Own When this bit is set, the MAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting
12	RW	0x0	LM Loopback Mode When this bit is set, the MAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally
11	RW	0x0	DM Duplex Mode When this bit is set, the MAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration
10	RW	0x0	IPC Checksum Offload When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared

Bit	Attr	Reset Value	Description
9	RW	0x0	DR Disable Retry When this bit is set, the MAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the MAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC will attempt retries based on the settings of BL
8	RW	0x0	LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up
7	RW	0x0	ACS Automatic Pad/CRC Stripping When this bit is set, the MAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the MAC will pass all incoming frames to the Host unmodified
6:5	RW	0x0	BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 = r < 2^k

Bit	Attr	Reset Value	Description
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmit state machine of the MAC is enabled for transmission on the GMII/MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames
2	RW	0x0	RE Receiver Enable When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII
1:0	RO	0x0	reserved

MAC_MAC_FRM_FILT
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the MAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter
30:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter
9	RW	0x0	SAF Source Address Filter Enable The MAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the MAC drops the frame. When this bit is reset, then the MAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter
7:6	RW	0x0	PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register MAC_FLOW_CTRL[2]. 2'b00: MAC filters all control frames from reaching the application. 2'b01: MAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: MAC forwards all control frames to application even if they fail the Address Filter. 2'b11: MAC forwards control frames that pass the Address Filter
5	RW	0x0	DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames

Bit	Attr	Reset Value	Description
4	RW	0x0	PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set

MAC_HASH_TAB_HI
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

MAC_HASH_TAB_LO
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

MAC_GMII_ADDR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_MAC frequency used in your design. The suggested range of pclk_MAC frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. Selection pclk_MAC MDC Clock 0000 60-100 MHz pclk_MAC/42 0001 100-150 MHz pclk_MAC/62 0010 20-35 MHz pclk_MAC/16 0011 35-60 MHz pclk_MAC/26 0100 150-250 MHz pclk_MAC/102 0101 250-300 MHz pclk_MAC/124 0110, 0111 Reserved When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE 802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_MAC is of frequency 100 MHz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks. Selection MDC Clock 1000 pclk_MAC/4 1001 pclk_MAC/6 1010 pclk_MAC/8 1011 pclk_MAC/10 1100 pclk_MAC/12 1101 pclk_MAC/14 1110 pclk_MAC/16 1111 pclk_MAC/18
1	RW	0x0	GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register MAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register MAC_GMII_DATA

Bit	Attr	Reset Value	Description
0	W1C	0x0	GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the MAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the MAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared

MAC_GMII_DATA
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation

MAC_FLOW_CTRL
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain
15:8	RO	0x0	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted. Selection Threshold 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface
3	RW	0x0	UP Unicast Pause Frame Detect When this bit is set, the MAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the MAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard
2	RW	0x0	RFE Receive Flow Control Enable When this bit is set, the MAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled
1	RW	0x0	TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the MAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register MAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the MAC will reset this bit to 1'b0. The register MAC_FLOW_CTRL should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC Core. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function

MAC_VLAN_TAG
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames

MAC_DEBUG
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission
24	RW	0x0	TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission
23	RO	0x0	reserved
22	RW	0x0	TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO
3	RO	0x0	reserved
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state

MAC_PMT_CTRL_STA
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the MAC (DAF) address recognition to be a wake-up frame
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception
0	R/W SC	0x0	PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high

MAC_INT_STATUS
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared
6	RO	0x0	MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration
5	RO	0x0	MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration
4	RO	0x0	MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration

Bit	Attr	Reset Value	Description
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register MAC_PMT_CTRL_STA
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register

MAC_INT_MASK
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register MAC_INT_STATUS
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register MAC_INT_STATUS

MAC_MAC_ADDR0_HI
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames

MAC_MAC_ADDR0_LO
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames

MAC_AN_CTRL
Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the MAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation
11:10	RO	0x0	reserved
9	R/W SC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation
8:0	RO	0x0	reserved

MAC_AN_STATUS
Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the MAC supports auto-negotiation
2	R/W SC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down
1:0	RO	0x0	reserved

MAC_AN_ADV
Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the MAC does not support the next page
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the MAC is capable of configuring the PAUSE function as defined in IEEE 802.3x
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the MAC supports Half-Duplex. This bit is tied to low (and RO) when the MAC is configured for Full-Duplex-only operation
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the MAC supports Full-Duplex
4:0	RO	0x0	reserved

MAC_AN_LINK_PART_AB
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the MAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved

Bit	Attr	Reset Value	Description
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner
11:9	RO	0x0	reserved
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode
4:0	RO	0x0	reserved

MAC_AN_EXP
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the MAC does not support next page function
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the MAC. This bit will be cleared when read
0	RO	0x0	reserved

MAC_INTF_MODE_STA
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)

Bit	Attr	Reset Value	Description
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

MAC_MMC_CTRL
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2K Bytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2K Bytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W SC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read

Bit	Attr	Reset Value	Description
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/W SC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

MAC_MMC_RX_INTR
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value
3	RO	0x0	reserved
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TX_INTR
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_RX_INT_MSK
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value
20:19	RO	0x0	reserved
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value
17:6	RO	0x0	reserved
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TX_INT_MSK
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value
18:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TXOCTETCNT_GB
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames

MAC_MMC_TXFRMCNT_GB
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames

MAC_MMC_TXUNDFLWERR
Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error

MAC_MMC_TXCARERR
Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier)

MAC_MMC_TXOCTETCNT_G
Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only

MAC_MMC_TXFRMCNT_G
Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted

MAC_MMC_RXFRMCNT_GB
Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received

MAC_MMC_RXOCTETCNT_GB
Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames

MAC_MMC_RXOCTETCNT_G
Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames

MAC_MMC_RXMCFRMCNT_G
Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received

MAC_MMC_RXCRCERR
Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error

MAC_MMC_RXLENERR
Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠frame size), for all frames with valid length field

MAC_MMC_RXFIFOOVRFLW
Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow

MAC_MMC_IPC_INT_MSK
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
28	RO	0x0	reserved
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
26	RO	0x0	reserved
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_IPC_INTR
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value
4:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_RXIPV4GFRM
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

MAC_MMC_RXIPV4HDERRFRM
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

MAC_MMC_RXIPV6GFRM
Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads

MAC_MMC_RXIPV6HDERRFRM
Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch)

MAC_MMC_RXUDPERRFRM
Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error

MAC_MMC_RXTCPERRFRM
Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error

MAC_MMC_RXICMPERRFRM
Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error

MAC_MMC_RXIPV4HDERROCT
Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter

MAC_MMC_RXIPV6HDERROCT
Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter

MAC_MMC_RXUDPERROCT
Address: Operational Base + offset (0x0274)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors

MAC_MMC_RXTCPERROCT
Address: Operational Base + offset (0x027c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors

MAC_MMC_RXICMPERROCT
Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors

MAC_BUS_MODE
Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value

Bit	Attr	Reset Value	Description
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x01	PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly. For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less. For RxFIFO, valid PBL range in full duplex mode is all
7	RO	0x0	reserved
6:2	RW	0x00	DSL Descriptor Skip Length This bit specifies the number of dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode
1	RO	0x0	reserved
0	R/W SC	0x1	SWR Software Reset When this bit is set, the MAC DMA Controller resets all MAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion

MAC_TX_POLL_DEMAND
Address: Operational Base + offset (0x1004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register MAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register MAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes

MAC_RX_POLL_DEMAND
Address: Operational Base + offset (0x1008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register MAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register MAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state

MAC_RX_DESC_LIST_ADDR
Address: Operational Base + offset (0x100c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only

MAC_TX_DESC_LIST_ADDR
Address: Operational Base + offset (0x1010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only

MAC_STATUS

Address: Operational Base + offset (0x1014)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI MAC PMT Interrupt This bit indicates an interrupt event in the MAC core's PMT module. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
27	RO	0x0	GMI MAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the MAC core. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
26	RO	0x0	GLI MAC Line interface Interrupt This bit reflects an interrupt event in the MAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
25:23	RO	0x0	EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register MAC_STATUS[13]) set. This field does not generate an interrupt. Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access

Bit	Attr	Reset Value	Description
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory
16	W1 C	0x0	NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register MAC_STATUS[0]: Transmit Interrupt Register MAC_STATUS[2]: Transmit Buffer Unavailable Register MAC_STATUS[6]: Receive Interrupt Register MAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared

Bit	Attr	Reset Value	Description
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register MAC_STATUS[1]: Transmit Process Stopped Register MAC_STATUS[3]: Transmit Jabber Timeout Register MAC_STATUS[4]: Receive FIFO Overflow Register MAC_STATUS[5]: Transmit Underflow Register MAC_STATUS[7]: Receive Buffer Unavailable Register MAC_STATUS[8]: Receive Process Stopped Register MAC_STATUS[9]: Receive Watchdog Timeout Register MAC_STATUS[10]: Early Transmit Interrupt Register MAC_STATUS[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared
14	W1C	0x0	ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register MAC_STATUS[6] automatically clears this bit
13	W1C	0x0	FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses
12:11	RO	0x0	reserved
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state

Bit	Attr	Reset Value	Description
7	W1C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register MAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA
6	W1C	0x0	RI Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state
5	W1C	0x0	UNF Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set
4	W1C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11]
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped

Bit	Attr	Reset Value	Description
0	W1C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor

MAC_OP_MODE
Address: Operational Base + offset (0x1018)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset
23:22	RO	0x0	reserved
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register MAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped

Bit	Attr	Reset Value	Description
20	W1C	0x0	FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active
19:17	RO	0x0	reserved
16:14	RW	0x0	TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16

Bit	Attr	Reset Value	Description
13	RW	0x0	ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register MAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register MAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state
12:11	RW	0x0	RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is de-asserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the de-assertion is effective only after flow control is asserted
10:9	RW	0x0	RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high

Bit	Attr	Reset Value	Description
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01)
5	RO	0x0	reserved
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained

Bit	Attr	Reset Value	Description
1	RW	0x0	SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register MAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register MAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register MAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state
0	RO	0x0	reserved

MAC_INT_ENA
Address: Operational Base + offset (0x101c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register MAC_STATUS[0]: Transmit Interrupt Register MAC_STATUS[2]: Transmit Buffer Unavailable Register MAC_STATUS[6]: Receive Interrupt Register MAC_STATUS[14]: Early Receive Interrupt

Bit	Attr	Reset Value	Description
15	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register MAC_STATUS[1]: Transmit Process Stopped Register MAC_STATUS[3]: Transmit Jabber Timeout Register MAC_STATUS[4]: Receive Overflow Register MAC_STATUS[5]: Transmit Underflow Register MAC_STATUS[7]: Receive Buffer Unavailable Register MAC_STATUS[8]: Receive Process Stopped Register MAC_STATUS[9]: Receive Watchdog Timeout Register MAC_STATUS[10]: Early Transmit Interrupt Register MAC_STATUS[13]: Fatal Bus Error
14	RW	0x0	ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled
13	RW	0x0	FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled
12:11	RO	0x0	reserved
10	RW	0x0	ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled
9	RW	0x0	RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled

Bit	Attr	Reset Value	Description
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled

MAC_OVERFLOW_CNT
Address: Operational Base + offset (0x1020)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1

MAC_REC_INT_WDT_TIMER
Address: Operational Base + offset (0x1024)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame

MAC_AXI_BUS_MODE
Address: Operational Base + offset (0x1028)

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the MAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller

Bit	Attr	Reset Value	Description
30	RW	0x0	UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received
29:22	RO	0x0	reserved
21:20	RW	0x1	WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1
19:18	RO	0x0	reserved
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit Register0 (register MAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4

Bit	Attr	Reset Value	Description
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register MAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1

MAC_AXI_STATUS
Address: Operational Base + offset (0x102c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data

MAC_CUR_HOST_TX_DESC
Address: Operational Base + offset (0x1048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_RX_DESC
Address: Operational Base + offset (0x104c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_TX_BUF_ADDR
Address: Operational Base + offset (0x1050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_RX_BUF_ADDR
Address: Operational Base + offset (0x1054)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation

12.5 Interface Description

Table 12-1 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	IO_CIFclkinm0_RMIIclk_GPIO2B2vccio3	GPIO2B_IOMUX_SEL_L[10:8]=3'b10
mac_txen	O	IO_CIFd2m0_RMIItxen_GPIO2A0vccio3	GPIO2A_IOMUX_SEL_L[2:0]=3'b10
mac_txd1	O	IO_CIFd3m0_RMIItxd1_GPIO2A1vccio3	GPIO2A_IOMUX_SEL_L[6:4]=3'b10
mac_txd0	O	IO_CIFd4m0_RMIItxd0_GPIO2A2vccio3	GPIO2A_IOMUX_SEL_L[10:8]=3'b10
mac_rxdv	I	IO_CIFd8m0_RMIIrxdv_GPIO2A6vccio3	GPIO2A_IOMUX_SEL_H[10:8]=3'b10
mac_rxer	I	IO_CIFd7m0_RMIIrxer_GPIO2A5vccio3	GPIO2A_IOMUX_SEL_H[6:4]=3'b10
mac_rxd1	I	IO_CIFd6m0_RMIIrxd1_GPIO2A4vccio3	GPIO2A_IOMUX_SEL_H[2:0]=3'b10
mac_rxd0	I	IO_CIFd5m0_RMIIrxd0_GPIO2A3vccio3	GPIO2A_IOMUX_SEL_L[14:12]=3'b10
Management interface			
mac_mdio	I/O	IO_CIFd9m0_RMIImdio_GPIO2A7vccio3	GPIO2A_IOMUX_SEL_H[14:12]=3'b10
mac_mdc	O	IO_CIFhrefm0_RMIImdc_GPIO2B1vccio3	GPIO2B_IOMUX_SEL_L[6:4]=3'b10

Notes: I=input, O=output, I/O=input/output, bidirectional

12.6 Application Notes

12.6.1 Descriptors

The DMA in MAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.
There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.
A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers.

However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled
The descriptor ring and chain structure is shown in following figure.

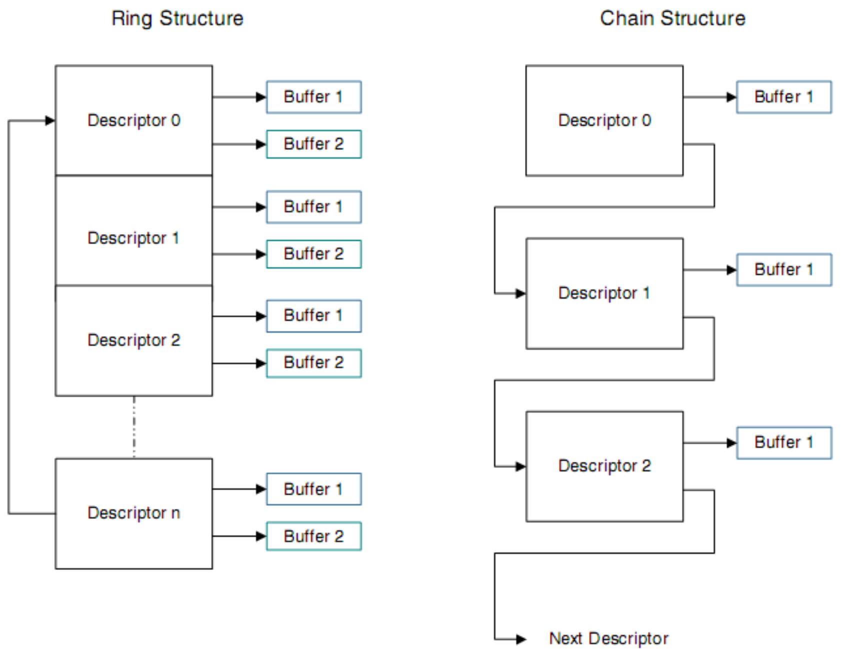


Fig. 12-11 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

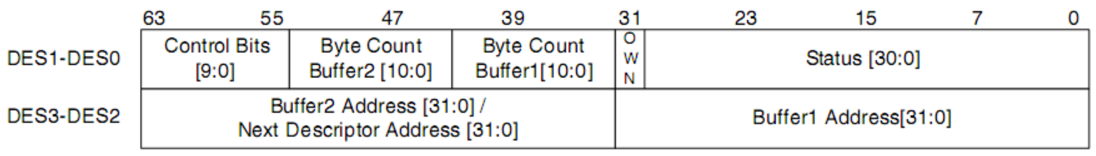


Fig. 12-12 Rx/Tx Descriptors definition

12.6.2 Receive Descriptor

The MAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMAalways attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.
In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)
RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 12-2 Receive Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the MAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception

Bit	Description
	or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the MAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: • RDES0[0]: Payload Checksum Error • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: IPC Checksum • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the MAC Core.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the MAC Core.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
7	IPC Checksum Error/Giant Frame When IP Checksum Engine is enabled, this bit, when set, indicates that the 16-bit IPv4 Header checksum calculated by the core did not match the received checksum bytes. The Error Summary bit[15] is NOT set when this bit is set in this

Bit	Description
	mode.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error (rxd≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.
1	CE: CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Table 12-3 Receive Descriptor 1

Bit	Description
31	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the MAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].

Bit	Description
23:22	Reserved.
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Receive Descriptor 2 (RDES2)
RDES2 contains the address pointer to the first data buffer in the descriptor.

Table 12-4 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)
RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 12-5 Receive Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

12.6.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width used (64). Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Transmit Descriptor 0 (TDES0)

TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 12-6 Transmit Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:17	Reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.
15	ES: Error Summary Indicates the logical OR of the following bits: • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the MAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the MAC operates in Half-Duplex Mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII Mode and 512 byte times including Preamble and Carrier Extension in RGMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision

Bit	Description
	When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the MAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the MAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the MAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register MAC_STATUS[5]) and Transmit Interrupt (Register MAC_STATUS [0]).
0	DB: Deferred Bit When set, this bit indicates that the MAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 12-7 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.

Bit	Description
26	DC: Disable CRC When set, the MAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)
TDES2 contains the address pointer to the first buffer of the descriptor.

Table 12-8 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)
TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 12-9 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

12.6.4 Programming Guide

DMA Initialization – Descriptors
The following operations must be performed to initialize the DMA.
1. Provide a software reset. This will reset all of the MAC internal registers and logic. (MAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll MAC_OP_MODE[0], which is only cleared after the reset operation is completed).

3. Program the following fields to initialize the Bus Mode Register by setting values in register MAC_BUS_MODE
   a. Mixed Burst and AAL
   b. Fixed burst or undefined burst
   c. Burst length values and burst mode values.
   d. Descriptor Length (only valid if Ring Mode is used)
   e. Tx and Rx DMA Arbitration scheme
4. Program the AXI Interface options in the register MAC_BUS_MODE
   a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])
5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.
6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.
7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register MAC_RX_DESC_LIST_ADDR and MAC_TX_DESC_LIST_ADDR).
8. Program the following fields to initialize the mode of operation by setting values in register MAC_OP_MODE
   a. Receive and Transmit Store And Forward
   b. Receive and Transmit Threshold Control (RTC and TTC)
   c. Hardware Flow Control enable
   d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
   e. Error Frame and undersized good frame forwarding enable
   f. OSF Mode
9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register MAC_STATUS).
10. Enable the interrupts by programming the interrupt enable register MAC_INT_ENA.
11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register MAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow.
1. Program the register MAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of (MAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in registerMAC_GMII_ADDR (bits 15-11).
3. Provide the MAC address registers (MAC_MAC_ADDR0_HI and MAC_MAC_ADDR0_LO).
4. If Hash filtering is enabled in your configuration, program the Hash filter register (MAC_HASH_TAB_HI and MAC_HASH_TAB_LO).
5. Program the following fields to set the appropriate filters for the incoming frames in register MAC_MAC_FRM_FILT
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, Multicast, broad cast and control frames filter settings etc.
6. Program the following fields for proper flow control in register MAC_FLOW_CTRL.
   a. Pause time and other pause frame control bits
   b. Receive and Transmit Flow control bits

c. Flow Control Busy/Backpressure Activate
7. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
8. Program the appropriate fields in register MAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
9. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register MAC_MAC_CONF.

Normal Receive and Transmit Operation
For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (MAC_TX_POLL_DEMAND and MAC_RX_POLL_DEMAND).
- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (MAC_CUR_HOST_TX_DESC and MAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (MAC_CUR_HOST_TX_Buf_ADDR and MAC_CUR_HOST_RX_BUF_ADDR).

Stop and Start Operation
When the transmission is required to be paused for some time then the following steps can be followed.
1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register MAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register MAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register MAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

12.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.
The mux select is CRU_CLKSEL23_CON[6].

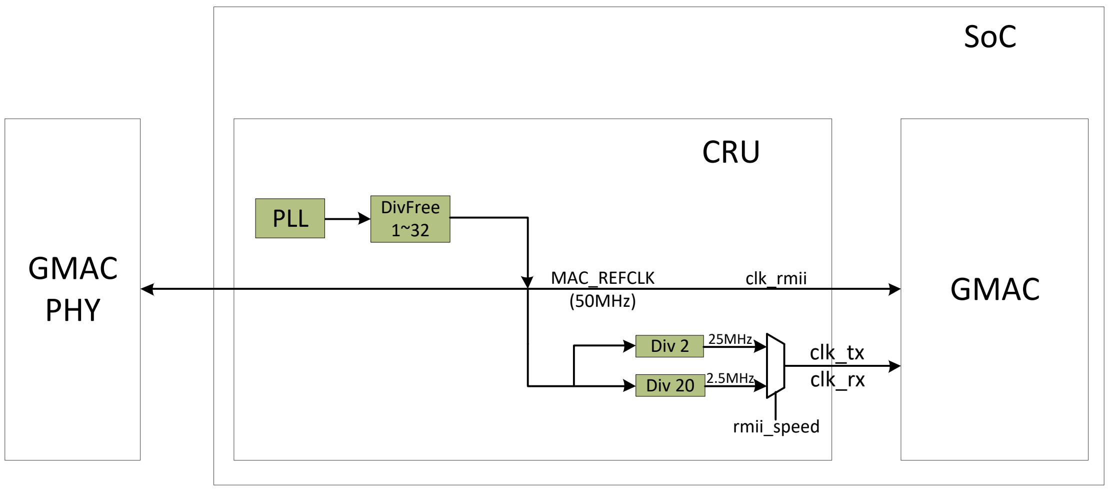


Fig. 12-13 RMII clock architecture when clock source from CRU

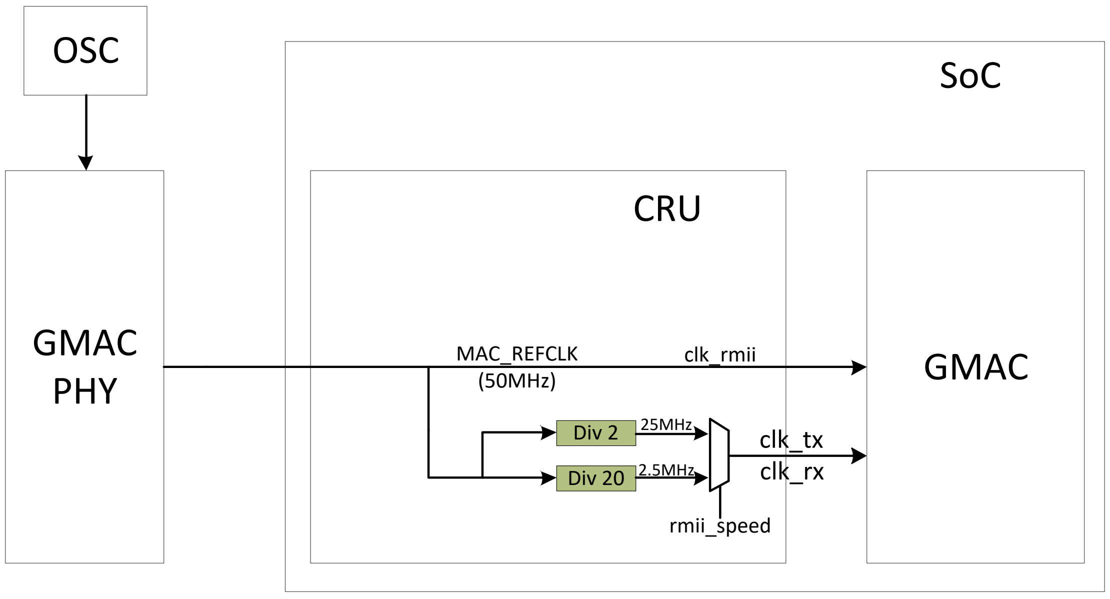
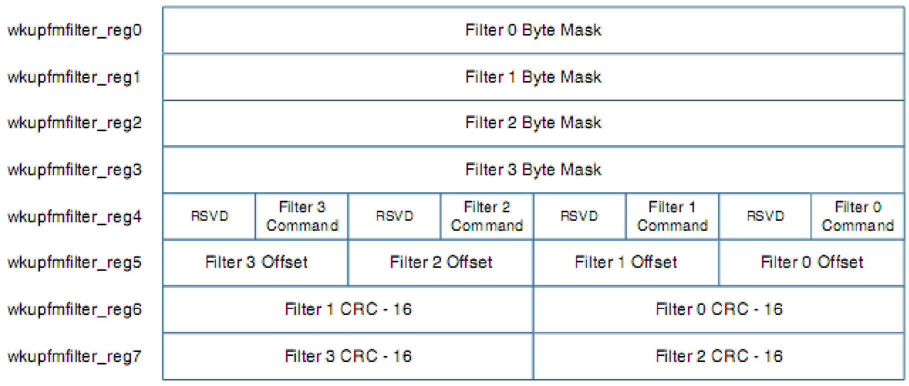


Fig. 12-14 RMII clock architecture when clock source from external OSC

12.6.6 Remote Wake-Up Frame Filter Register

The register wkupfmfilter_reg, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (wkupfmfilter_reg) must be written. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values in address (028) for wkupfmfilter_reg0, wkupfmfilter_reg1, ..., wkupfmfilter_reg7, respectively. Wkupfmfilter_reg is read in the same way.
The internal counter to access the appropriate wkupfmfilter_reg is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.



wkupfmfilter_reg0	Filter 0 Byte Mask							
wkupfmfilter_reg1	Filter 1 Byte Mask							
wkupfmfilter_reg2	Filter 2 Byte Mask							
wkupfmfilter_reg3	Filter 3 Byte Mask							
wkupfmfilter_reg4	RSVD	Filter 3 Command	RSVD	Filter 2 Command	RSVD	Filter 1 Command	RSVD	Filter 0 Command
wkupfmfilter_reg5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
wkupfmfilter_reg6	Filter 1 CRC - 16				Filter 0 CRC - 16			
wkupfmfilter_reg7	Filter 3 CRC - 16				Filter 2 CRC - 16			

Fig. 12-1Wake-Up Frame Filter Register

Filter i Byte Mask
This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.

Filter i Command
This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.

Filter i Offset
This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).

Filter i CRC-16
This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

12.6.7 System Consideration During Power-Down

MAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.
The PMT interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.
The recommended power-down and wake-up sequence is as follows.
1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register MAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.
6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the MAC asserts the PMT interrupt signal and exits Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to

the core.
9. Read the register MAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

12.6.8 GRF Register Summary

MAC2IO	
GRF Register	**Register Description**
GRF_MAC_CON1[2]	MACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON1[3]	MAC transmit flow control When set high, instructs the MAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the MAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON1[6:4]	PHY interface select 3'b001: RGMII(useless) 3'b100: RMII All others: Reserved
CRU_CLKSEL23_CON[6]	rmii_extclk_sel 1’b1:from CRU 1’b0:from IO
CRU_CLKSEL23_CON[7]	rmii_clk_sel 1’b1:100M 1’b0:10M

Chapter 13 Timer

13.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device.There are 6 non-secure timers and 2 secure timers.
Timer5 and STimer0~1 count up from zero to a programmed value and generate an interrupt when the counter reaches the programmed value.
Timer0~4 count down from a programmed value to zero and generate an interrupt when the counter reaches zero.
Timer supports the following features:

- Timer0~Timer5 is used for no-secure, STimer0~STimer1 is used for secure.
- Two operation modes: free-running and user-defined count.

13.2 Block Diagram

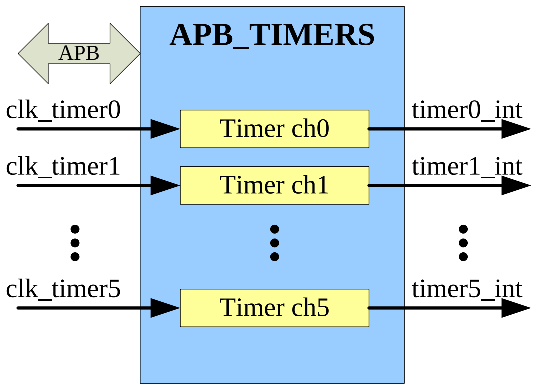


Fig. 13-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channels) that in the bus subsystem. The Stimers that in the bus subsystem only include two programmable timer channels.

13.3 Function Description

13.3.1 Timer clock

TIMER0~ TIMER5 and STIMER0~1 are in the pd_bus subsystem. The timer clock is 24MHz OSC.

13.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG (0≤n≤5) register:

- Disable the timer by writing a "0" to the timer enable bit (bit 0). Accordingly, the timer_en output signal is de-asserted.
- Program the timer mode—user-defined or free-running—by writing a "0" or "1" respectively, to the timer mode bit (bit 1).
- Set the interrupt mask as either masked or not masked by writing a "0" or "1" respectively, to the timer interrupt mask bit (bit 2).

2. Load the timer count value into the TIMERn_LOAD_COUNT1 (0≤n≤5) and TIMERn_LOAD_COUNT0 (0≤n≤5) register.
3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG (0≤n≤5).

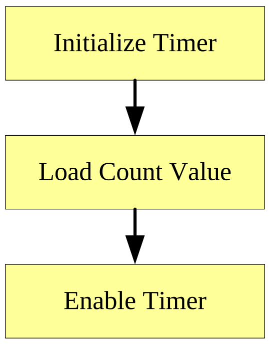


Fig. 13-2 Timer Usage Flow

13.3.3 Loading a timer count value

For the descending Timers(Timer0~4).The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the load count register (TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0). Two events can cause a timer to load the initial value from its load count register:

- Timer is enabled after reset or disabled.
- Timer counts down to 0, when timer is configured into free-running mode.

For the incremental Timers(Timer5 and STimer0~1).The initial value for each timer is zero. The count register will count up to the value loaded in the register TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0. Two events can cause a timer to load zero:

- Timer is enabled after reset or disabled.
- Timer counts up to the value stored in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0, when timer is configured into free-running mode.

13.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 registers (for descending timers) or zero (for incremental timers) as initial value. When the timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 (for incremental timers),it will not automatically reload the count register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0(for descending timers) or zero (for incremental timers)register as initial value. Timer will automatically reload the count register, when timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for incremental timers).

13.4 Register Description

13.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_LOAD_COUNT0	0x0000	W	0x00000000	Timern Load Count Register 0
TIMER_TIMERn_LOAD_COUNT1	0x0004	W	0x00000000	Timern Load Count Register 1.High 32 bits Value to be loaded into Timer n. This is the value from which counting commences
TIMER_TIMERn_CURRENT_VALUE0	0x0008	W	0x00000000	Timern Current Value Register 0

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_CURRENT_VALUE1	0x000c	W	0x00000000	Timern Current Value Register 1.High 32 bits of Current Value of Timer n
TIMER_TIMERn_CONTROLREG	0x0010	W	0x00000000	Timern Control Register
TIMER_TIMERn_INTSTATUS	0x0018	W	0x00000000	Timern Interrupt Status Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

13.4.2 Detail Register Description

TIMER_TIMERn_LOAD_COUNT0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_0 Low 32 bits Value to be loaded into Timer n. This is the value from which counting commences

TIMER_TIMERn_LOAD_COUNT1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_0 Low 32 bits Value to be loaded into Timer n. This is the value from which counting commences

TIMER_TIMERn_CURRENT_VALUE0
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value0 Low 32 bits of Current Value of Timer n

TIMER_TIMERn_CURRENT_VALUE1
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value0 Low 32 bits of Current Value of Timer n

TIMER_TIMERn_CONTROLREG
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	timer_int_mask Timer interrupt mask. 0: mask 1: not mask
1	RW	0x0	timer_mode Timer mode. 0: free-running mode 1: user-defined count mode
0	RW	0x0	timer_en Timer enable. 0: disable 1: enable

TIMER_TIMERn_INTSTATUS
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	timern_int This register contains the interrupt status for timern

13.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (0≤n≤5)), the timeren output signal is de-asserted, and timer_clk will stop. When user enables the timer, the timer_en signal is asserted and timer_clk will start running.
The application is only allowed to re-config registers when timer_en is low.

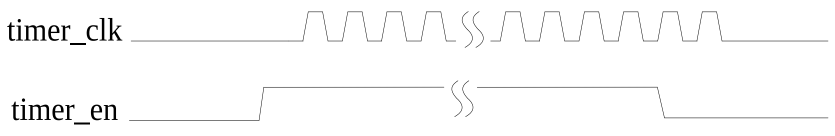


Fig. 13-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 14 System Debug

14.1 Overview

The chip uses the DAPLITE Technology to support real-time debug.

14.1.1 Features

- Invasive debug with core halted
- SW-DP

14.1.2 Debug components address map

The following table shows the debug components address in memory map:

Module	Base Address
DAP_ROM	0xff680000

14.2 Block Diagram

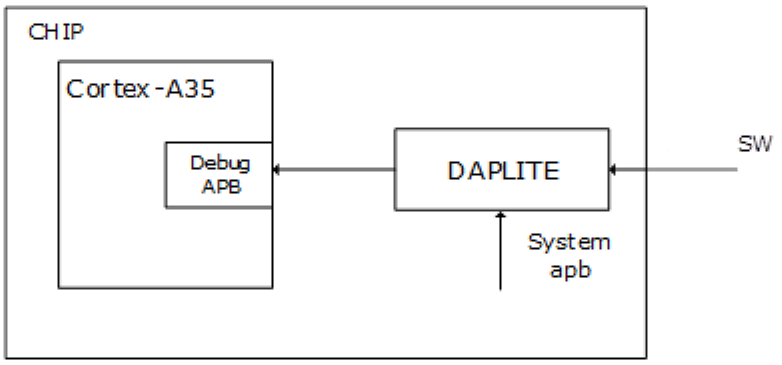


Fig. 14-1Debug system structure

14.3 Function Description

14.3.1 DAP

The DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)
- ROM table

The debug port is the host tools interface to access the DAP-Lite. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite supports a combined debug port which includes both JTAG and Serial Wire Debug(SWD), with a mechanism that supports switching between them.
The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.
The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit[31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.
More information please refer to the documentCoreSight_DAPLite_TRM.pdf for the debug detail description.

14.4 Register Description

Please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

14.5 Interface Description

14.5.1 DAP SW-DP Interface

This implementation is taken from ADIv5.1 and operates with a synchronous serial interface.This uses a single bidirectional data signal, and a clock signal.
The figure below describes the interaction between the timing of transactions on the serialwire interface, and the DAP internal bus transfers. It shows when the target respondswith a WAIT acknowledgement.

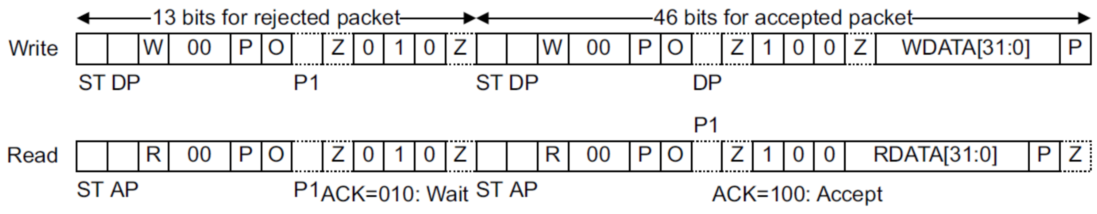


Fig. 14-2SW-DP acknowledgement timing

Table 14-1 SW-DP Interface Description

Module pin	Direction	Pad name	IOMUX
jtag_tck	I	IO_SDMMC0d2_UART4rx_JTAGtck_GPIO1D4vccio2	GRF_GPIO1D_IOMUX_H[2:0]=3'b 011
jtag_tms	I/O	IO_SDMMC0d3_UART4tx_JTAGtms_GPIO1D5vccio2	GRF_GPIO1D_IOMUX_h[6:4]=3'b 011

Chapter 15 WatchDog

15.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may becaused by conflicting parts or programs .The WDT would generate interrupt or reset signal when it's counter reaches zero, then a reset controller would reset the system. there are a Non-secure WDT(WDT_NS) and a Secure WDT(WDT_S);
WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
  - Generate a system reset
  - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period
- Support two WTD, one is used for non-secure application, the other is used for secure application

15.2 Block Diagram

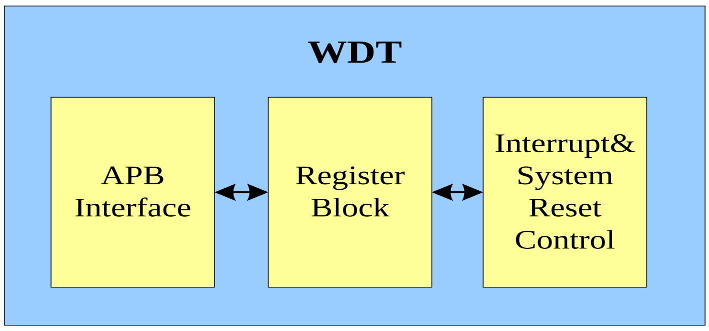


Fig. 15-1WDT block diagram

Block Descriptions:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & system reset control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

15.3 Function Description

15.3.1 Operation

Counter
The WDT counts from a preset (timeout) value in descending order to zero. When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

Interrupts
The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated.
System Resets
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.
Reset Pulse Length
The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset. A counter restart has no effect on the system reset once it has been asserted.

15.4 Register Description

This section describes the control/status registers of the design.

15.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x0000ffff	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

15.4.2 Detail Register Description

WDT_CR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. 000: 2 pclk cycles 001: 4 pclk cycles 010: 8 pclk cycles 011: 16 pclk cycles 100: 32 pclk cycles 101: 64 pclk cycles 110: 128 pclk cycles 111: 256 pclk cycles

Bit	Attr	Reset Value	Description
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 0: Generate a system reset. 1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset
0	RW	0x0	wdt_en WDT enable: 0: WDT disabled; 1: WDT enabled

WDT_TORR
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 0000: 0x0000ffff 0001: 0x0001ffff 0010: 0x0003ffff 0011: 0x0007ffff 0100: 0x000fffff 0101: 0x001fffff 0110: 0x003fffff 0111: 0x007fffff 1000: 0x00ffffff 1001: 0x01ffffff 1010: 0x03ffffff 1011: 0x07ffffff 1100: 0x0fffffff 1101: 0x1fffffff 1110: 0x3fffffff 1111: 0x7fffffff

WDT_CCVR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt Current counter value. This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	cnt_restart Counter restart. This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero

WDT_STAT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1: Interrupt is active regardless of polarity; 0: Interrupt is inactive

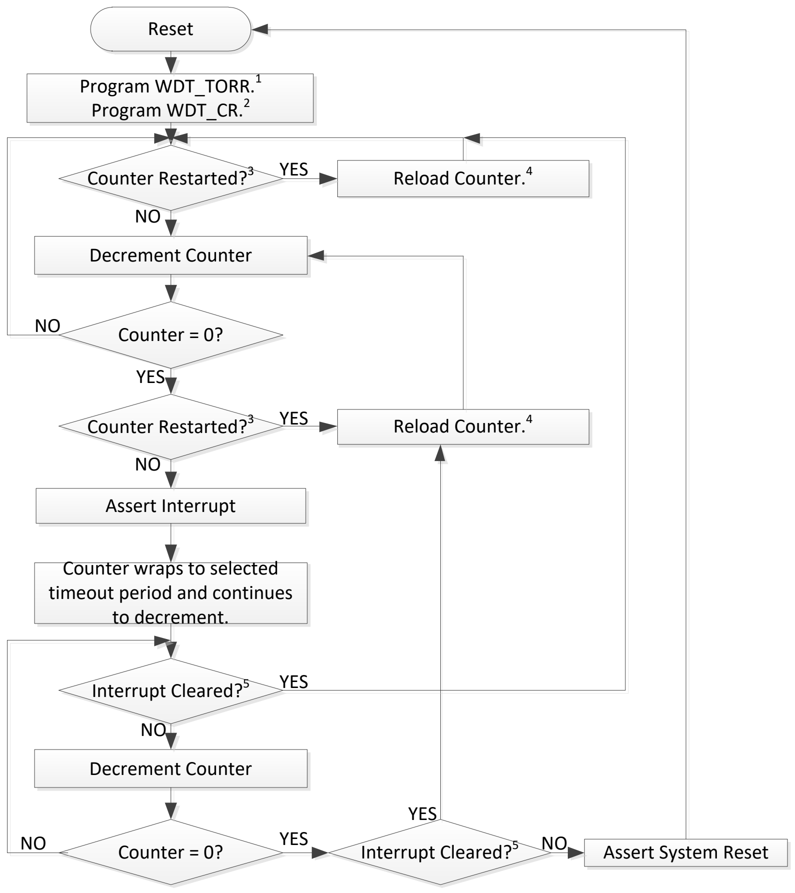
WDT_EOI
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter

15.5 Application Notes

15.5.1 Programming sequence

The following figure show the operation flow chart (Response mode=1).



1. Select required timeout period.
2. Set reset pulse length, response mode, and enable WDT.
3. Write 0x76 to WDT_CRR.
4. Starts back to selected timeout period.
5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 15-2WDT Operation Flow

Chapter 16 Serial Flash Controller (SFC)

16.1 Overview

The serial flash controller (SFC) is used to control the data transfer between the chip system and the serial nor/nand flash device.
The SFC supports the following features:

- Support AHB slave interface to configure register and read/write serial flash
- Support AHB master interface to transfer data from/to SPIflash device
- Support AHB burst with incr4x32bits, or incr x32bits
- Support two independent clock domain: AHB clock and SPI clock
- Support x1,x2,x4 data bits mode
- Support up to 4 chip select
- Support interrupt output, interrupt maskable
- Support Spansion, MXIC,Gigadevice…vendor's nor flash memory.

16.2 Block Diagram

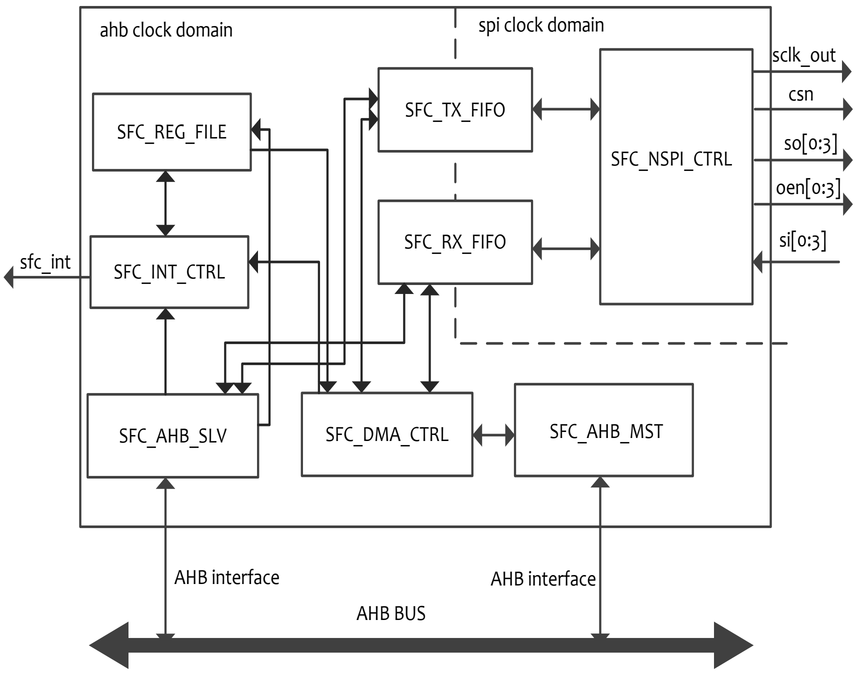


Fig.16-1SFC architecture

16.3 Function Description

16.3.1 SFC slave

The AHB slave is used to configure the register, and also write to/read from the serial nor/nand flash device.
The SFC_CTRL register is a global control register, when the controller is in busy state(SFC_SR), SFC_CTRL cannot be set. The field sclk_idle_level_cycles of this register is used to configure the idle level cycles of sclk before read the first bit of the read command. Like the following picture shows: the red line of the sclk is the idle cycles, during these cycles, the chip pad is switched to output. When sclk_idle_level_cycles=0, it means there will be not such idle level.

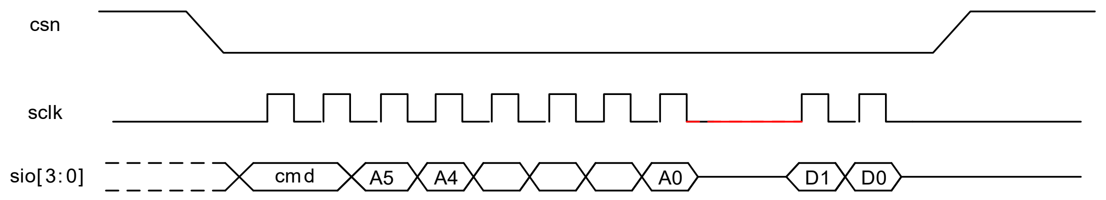


Fig.16-2idle cycles

When the field spi mode is set, the transfer waveform will like following, and switch to mode3.

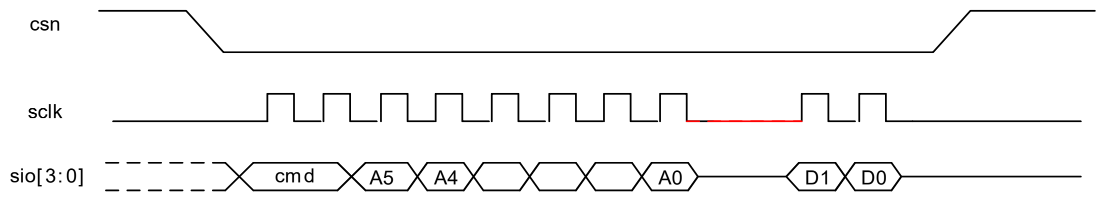


Fig.16-3SPI mode

16.4 Register Description

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SFC_CTRL	0x0000	W	0x00000000	Control Register
SFC_IMR	0x0004	W	0x00000000	Interrupt Mask
SFC_ICLR	0x0008	W	0x00000000	Interrupt Clear
SFC_FTLR	0x000c	W	0x00000000	FIFO Threshold Level
SFC_RCVR	0x0010	W	0x00000000	SFC Recover
SFC_AX	0x0014	W	0x00000000	SFC AX Value
SFC_ABIT	0x0018	W	0x00000000	Flash Address bits
SFC_ISR	0x001c	W	0x00000000	Interrupt Status
SFC_FSR	0x0020	W	0x00000001	FIFO Status
SFC_SR	0x0024	W	0x00000000	SFC Status
SFC_RISR	0x0028	W	0x00000000	Raw Interrupt Status
SFC_VER	0x002c	W	0x0a340003	Version Register
SFC_QOP	0x0030	W	0x00000000	quad line operation io level preset
SFC_DMATR	0x0080	W	0x00000000	DMA Trigger
SFC_DMAADDR	0x0084	W	0x00000000	DMA Address
SFC_CMD	0x0100	W	0x00000000	SFC CMD
SFC_ADDR	0x0104	W	0x00000000	Address the flash
SFC_DATA	0x0108	W	0x00000000	DATA that write to /read from the flash

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

16.4.2 Detail Register Description

SFC_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	DATB Data bits width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: reserved
11:10	RW	0x0	ADRB Address bits width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: reserved
9:8	RW	0x0	CMDB Command bits width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11:reserved
7:4	RW	0x0	IDLE_CYCLE 4'b0: idle hold is disable 4'b1: hold the sclk_out in idle for two cycles when switch to shift in ....
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE 1'b0: shift in the data at posedge sclk_out 1'b1: shift in the data at negedge sclk_out
0	RW	0x0	SPIM SPI MODE Select 1'b0: mode 0 1'b1: mode 3

SFC_IMR
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	DMAM 1'b0: dma_intr interrupt is not masked 1'b1: dma_intr interrupt is masked
6	RW	0x0	NSPIM 1'b0: nspi_intr interrupt is not masked 1'b1: nspi_intr interrupt is masked
5	RW	0x0	AHBM 1'b0: ahb_intr interrupt is not masked 1'b1: ahb_intr interrupt is masked

Bit	Attr	Reset Value	Description
4	RW	0x0	TRANSM 1'b0: transf_intr interrupt is not masked 1'b1: transf_intr interrupt is masked
3	RW	0x0	TXEM 1'b0: txe_intr interrupt is not masked 1'b1: txe_intr interrupt is masked
2	RW	0x0	TXOM 1'b0: txo_intr interrupt is not masked 1'b1: txo_intr interrupt is masked
1	RW	0x0	RXUM 1'b0: rxu_intr interrupt is not masked 1'b1: rxu_intr interrupt is masked
0	RW	0x0	RXFM 1'b0: rxf_intr interrupt is not masked 1'b1: rxf_intr interrupt is masked

SFC_ICLR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	DMAC DMA finish Interrupt Clear
6	W1C	0x0	NSPIC SPI Error Interrupt Clear
5	W1C	0x0	AHBC AHB Error Interrupt Clear
4	W1C	0x0	TRANSC Transfer finish Interrupt Clea
3	W1C	0x0	TXEC Transmit FIFO Empty Interrupt Clear
2	W1C	0x0	TXOC Transmit FIFO Overflow Interrupt Clear
1	W1C	0x0	RXUC Receive FIFO Underflow Interrupt Clear
0	W1C	0x0	RXFC Receive FIFO Full Interrupt Clear

SFC_FTLR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	RXFTLR When the number of receive FIFO entries is bigger than or equal to this value, the receive FIFO full interrupt is triggered.
7:0	RW	0x00	TXFTLR When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SFC_RCVR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	RCVR SFC Recover Write 1 to recover the SFC State Machine, FIFO state and other logic state.

SFC_AX
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	AX The AX Value when doing the continuous read(enhance mode).

SFC_ABIT
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	ABIT Flash Address bits

SFC_ISR
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: not active 1'b1: active
6	RO	0x0	NSPIS SPI Error Interrupt Statu 1'b0: not active 1'b1: active

Bit	Attr	Reset Value	Description
5	RO	0x0	AHBS AHB Error Interrupt Status 1'b0: not active 1'b1: active
4	RO	0x0	TRANSS Transfer finish Interrupt Status 1'b0: not active 1'b1: active
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: not active 1'b1: active
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: not active 1'b1: active
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: not active 1'b1: active
0	RW	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: not active 1'b1: active

SFC_FSR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:16	RO	0x00	RXWLVL RX FIFO Water Level 0x0: fifo is empty 0x1: 1 entry is taken ... 0x10:16 entry is taken, fifo is full
15:13	RO	0x0	reserved
12:8	RO	0x00	TXWLVL TX FIFO Water Level 0x0: fifo is full 0x1: left 1 entry ... 0x10:left 16 entry, fifo is empty
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RO	0x0	RXFS Receive FIFO Full Status 1'b0: rx fifo is not full 1'b1: rx fifo is full
2	RO	0x0	RXES Receive FIFO Empty Status 1'b0: rx fifo is not empty 1'b1: rx fifo is empty
1	RO	0x0	TXES Transmit FIFO Empty Status 1'b0: tx fifo is not empty 1'b1: tx fifo is empty
0	RO	0x1	TXFS Transmit FIFO Full Status 1'b0: tx fifo is not full 1'b1: tx fifo is full

SFC_SR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	SR 0: SFC is idle 1: SFC is busy When busy, don't set the control register.

SFC_RISR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: not active 1'b1: active
6	RO	0x0	NSPIS SPI Error Interrupt Statu 1'b0: not active 1'b1: active
5	RO	0x0	AHBS AHB Error Interrupt Status 1'b0: not active 1'b1: active

Bit	Attr	Reset Value	Description
4	RO	0x0	TRANSS Transfer finish Interrupt Status 1'b0: not active 1'b1: active
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: not active 1'b1: active
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: not active 1'b1: active
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: not active 1'b1: active
0	RO	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: not active 1'b1: active

SFC_VER
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	RW	0x0a340003	VER the version id of sfc

SFC_QOP
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	SO123 the value of SO1,SO2 and SO3 during command and address bits input

SFC_DMATR
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	DMATR Write 1 to start the dma transfer.

SFC_DMAADDR
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAADDR DMA Address

SFC_CMD
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	WO	0x0	CS Flash chip select 2'b00: chip select 0 2'b01: chip select 1 2'b10: chip select 2 2'b11: chip select 3
29:16	WO	0x0000	TRB Total Data Bytes number that will write to /read from the flash.
15:14	WO	0x0	ADDRB Address bits number select, if there is not address command to send, set to zero 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register
13	WO	0x0	CONT Continuous read mode 1'b0: disable continuous read mode 1'b1: enable continuous read mode
12	WO	0x0	WR Flash Write or Read 1'b0:read 1'b1:write
11:8	WO	0x0	DUMM Dummy Bits Number
7:0	WO	0x00	CMD Flash Command

SFC_ADDR
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ADDR Flash's address

SFC_DATA
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DATA Flash's Data

16.5 Interface Description

Table 16-11SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
sfc_clk	O	IO_FLASHrdy_EMMCclkout_SFCclk_GPIO1B1vccio0	GRF_GPIO1B_IOMUX_SEL_L[6:4]=3’b011
sfc_csn0	O	IO_FLASHd4_EMMCd4_SFCcsn0_GPIO1A4vccio0	GRF_GPIO1A_IOMUX_SEL_H[2:0]=3’b011
sfc_sio0	I/O	IO_FLASHd0_EMMCd0_SFCsio0_GPIO1A0vccio0	GRF_GPIO1A_IOMUX_SEL_L[2:0]=3’b011
sfc_sio1	I/O	IO_FLASHd1_EMMCd1_SFCsio1_GPIO1A1vccio0	GRF_GPIO1A_IOMUX_SEL_L[6:4]=3’b011
sfc_sio2	I/O	IO_FLASHd2_EMMCd2_SFCsio2_GPIO1A2vccio0	GRF_GPIO1A_IOMUX_SEL_L[10:8]=3’b011
sfc_sio3	I/O	IO_FLASHd3_EMMCd3_SFCsio3_GPIO1A3vccio0	GRF_GPIO1A_IOMUX_SEL_L[14:12]=3’b011

Notes: I=input, O=output, I/O=input/output, bidirectional.

16.6 Application Notes

16.6.1 AHB Slave write flash flow

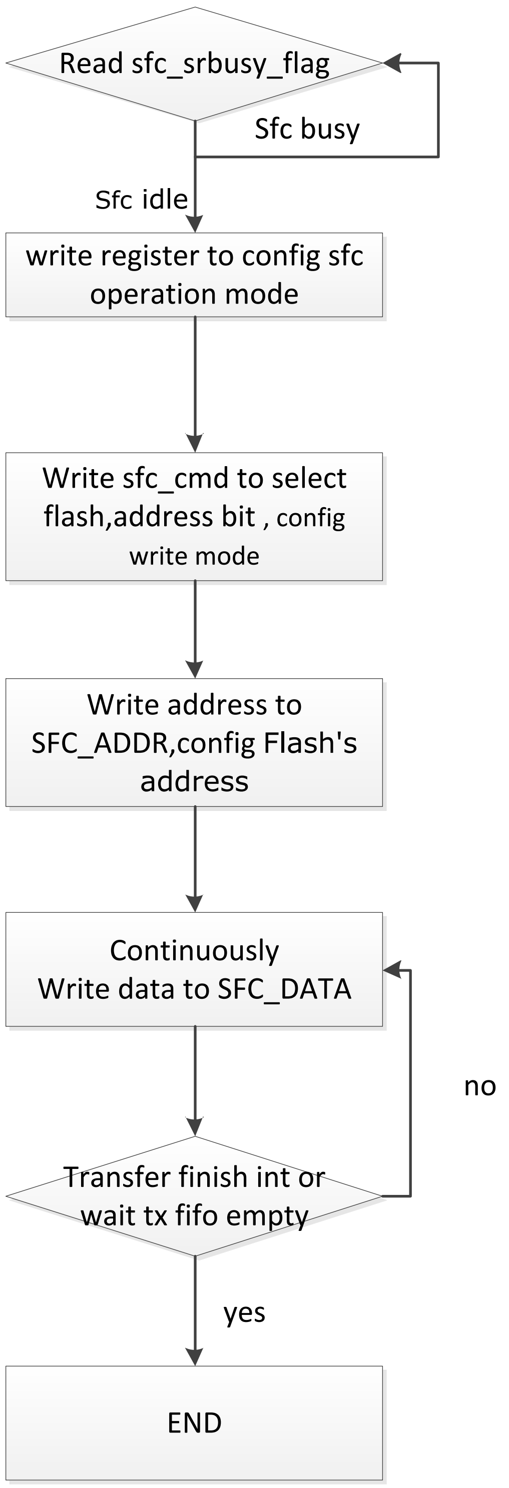

Read sfc_srbusy_flag
Sfc busy
Sfc idle
write register to config sfc operation mode
Write sfc_cmd to select flash,address bit , config write mode
Write address to SFC_ADDR,config Flash's address
Continuously Write data to SFC_DATA
no
Transfer finish int or wait tx fifo empty
yes
END

Fig.16-4slave mode write

16.6.2 AHB Slave read flash flow

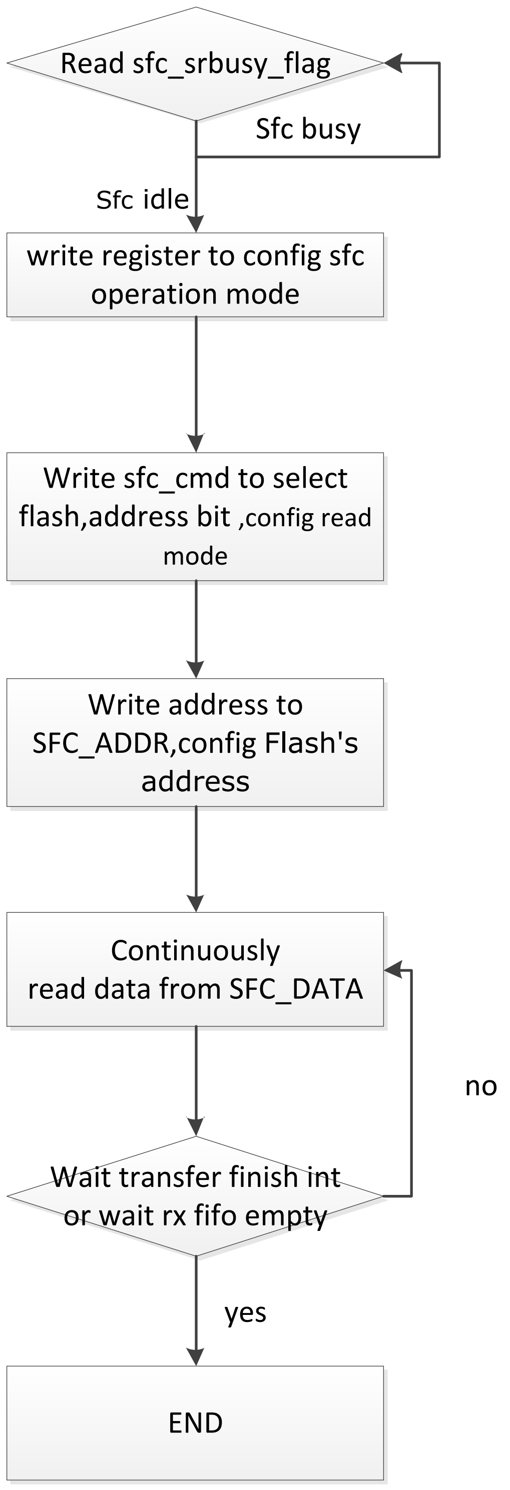

Read sfc_srbusy_flag
Sfc busy
Sfc idle
write register to config sfc operation mode
Write sfc_cmd to select flash,address bit ,config read mode
Write address to SFC_ADDR,config Flash's address
Continuously read data from SFC_DATA
no
Wait transfer finish int or wait rx fifo empty
yes
END

Fig.16-5slave mode read

16.6.3 AHB DMA transfer flow

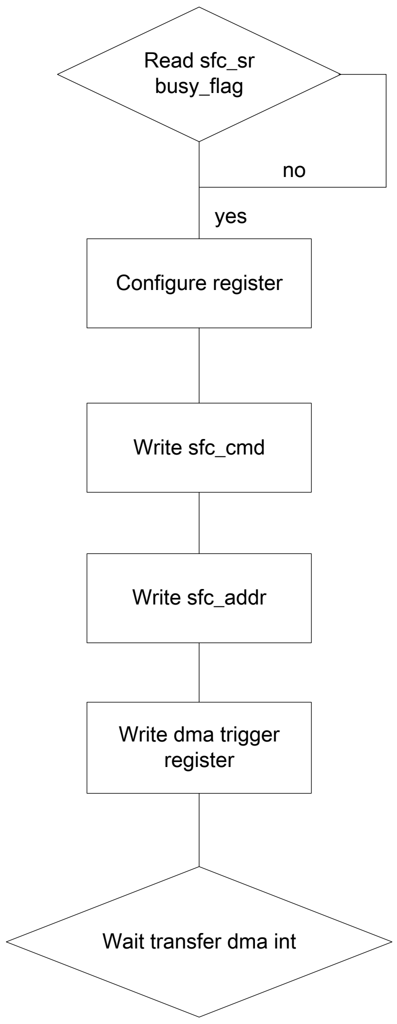


Fig.16-6master mode flow

16.6.4 Other Notes

The SFC_clk need to be kept under 150MHZ. It’s better to soft reset the SFC before data transfer.

Chapter 17 Serial Peripheral Interface (SPI)

17.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

17.2 Block Diagram

The SPI Controller comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt

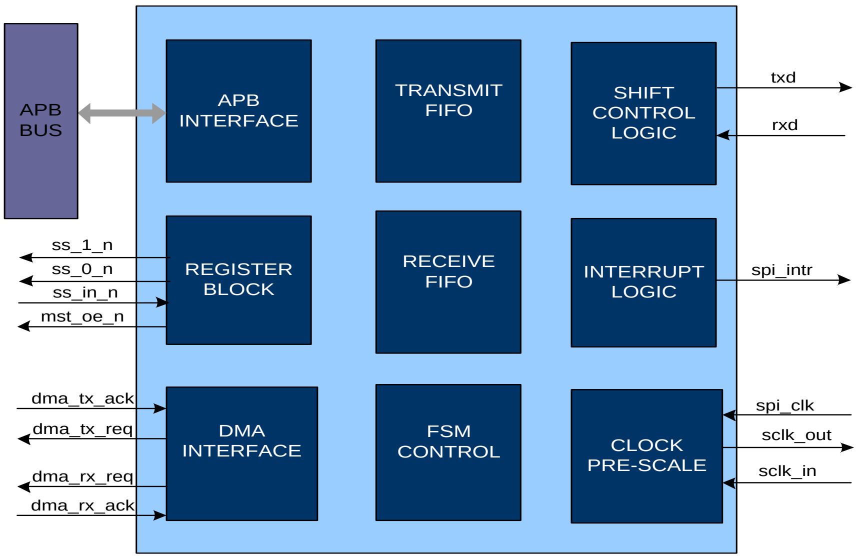


Fig. 17-1SPI Controller Block diagram

APB INTERFACE
The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits and 8 or 16 bits when reading or writing internal FIFO if data frame size(SPI_CTRL0[1:0]) is set to 8 bits.

DMA INTERFACE
This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC
For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL
Control the state's transformation of the design.

REGISTER BLOCK
All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL
Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL
The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

17.3 Function Description

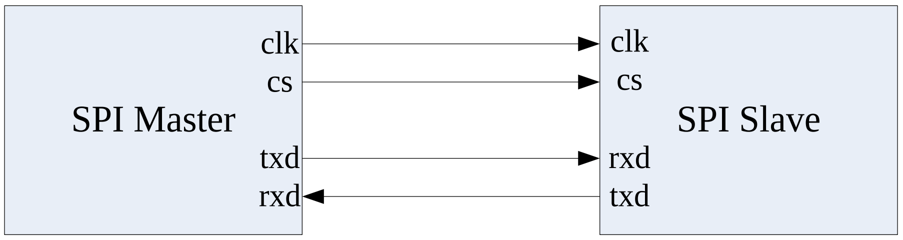


Fig. 17-2SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes
The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes
The SPI operates in the following three modes when transferring data on the serial bus.
1). Transmit and Receive
When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.
2).Transmit Only
When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.
3).Receive Only
When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.
Clock Ratios
A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,
When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})
When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.
When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

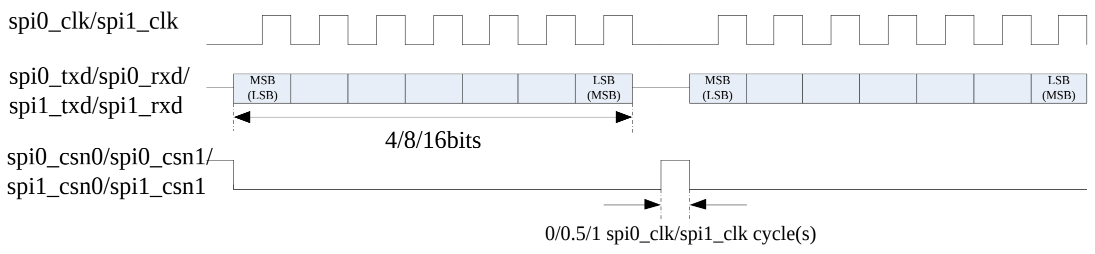


Fig. 17-3SPI Format (SCPH=0 SCPOL=0)

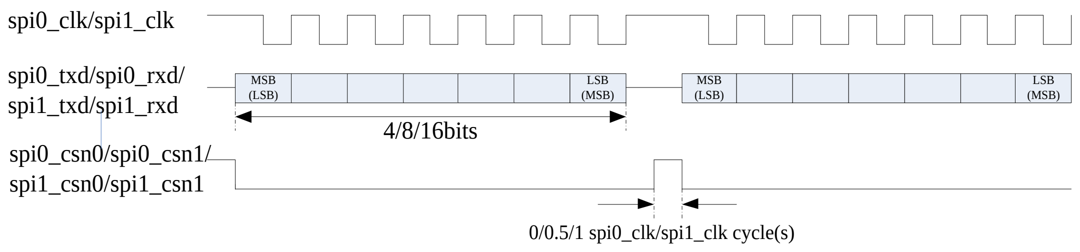


Fig. 17-4SPI Format (SCPH=0 SCPOL=1)
When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. The following two figures show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

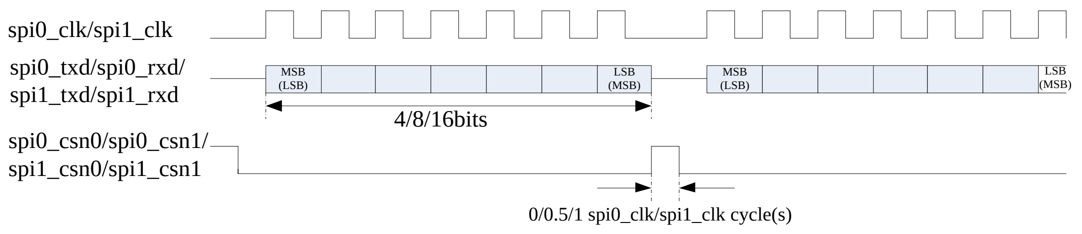


Fig. 17-5SPI Format (SCPH=1 SCPOL=0)

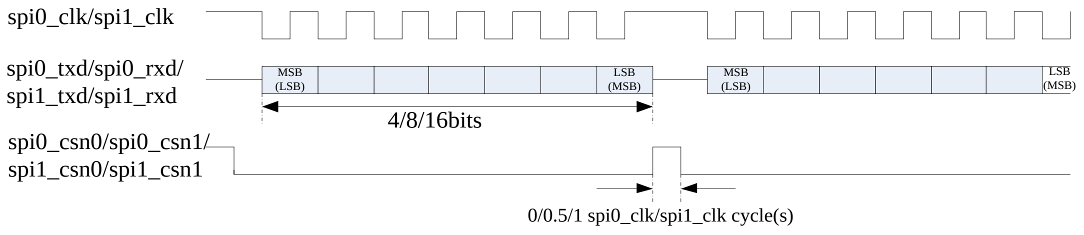


Fig. 17-6SPI Format (SCPH=1 SCPOL=1)

17.4 Register Description

17.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable Register
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x00000004	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

17.4.2 Detail Register Description

SPI_CTRLR0
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Microwire transfer mode. Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM Operation mode. 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	XFM Transfer mode. 2'b00 :Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 : reserved
17:16	RW	0x0	FRF Frame format. 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd sample delay. When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform. Valid when data frame size is 8bit. 1'b0:apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First bit mode. 1'b0:first bit is MSB 1'b1:first bit is LSB

Bit	Attr	Reset Value	Description
11	RW	0x0	EM Endian mode. Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0:little endian 1'b1:big endian
10	RW	0x0	SSD ss_n to sclk_out delay. Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle
9:8	RW	0x0	CSM Chip select mode. Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01:ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11:reserved
7	RW	0x0	SCPOL Serial Clock Polarity. Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low 1'b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase. Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit 1'b1: Serial clock toggles at start of first data bit

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CFS Control Frame Size.Selects the length of the control word for the Microwire frame format. 4'b0000~0010:reserved 4'b0011:4-bit serial data transfer 4'b0100:5-bit serial data transfer 4'b0101:6-bit serial data transfer 4'b0110:7-bit serial data transfer 4'b0111:8-bit serial data transfer 4'b1000:9-bit serial data transfer 4'b1001:10-bit serial data transfer 4'b1010:11-bit serial data transfer 4'b1011:12-bit serial data transfer 4'b1100:13-bit serial data transfer 4'b1101:14-bit serial data transfer 4'b1110:15-bit serial data transfer 4'b1111:16-bit serial data transfer
1:0	RW	0x2	DFS Data frame size,selects the data frame length. 2'b00:4bit data 2'b01:8bit data 2'b10:16bit data 2'b11:reserved

SPI_CTRLR1
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM Number of Data Frames.When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer

SPI_ENR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ENR Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled

SPI_SER
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	SER Slave Select Enable.This register is valid only when SPI is configured as a master device

SPI_BAUDR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	BAUDR SPI Clock Divider. Baud Rate Select. This register is valid only when the SPI is configured as a master device.The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register.If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TXFTLR Transmit FIFO Threshold Level.When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered

SPI_RXFTLR
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RXFTLR Receive FIFO Threshold Level.When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered

SPI_TXFLR
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TXFLR Transmit FIFO Level.Contains the number of valid data entries in the transmit FIFO

SPI_RXFLR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Reveive FIFO Level.Contains the number of valid data entries in the receive FIFO

SPI_SR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF Receive FIFO Full. 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RW	0x0	RFE Receive FIFO Empty. 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE Transmit FIFO Empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF Transmit FIFO Full. 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RO	0x0	BSF SPI Busy Flag.When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	IPR Interrupt Polarity Register. 1'b0:Active Interrupt Polarity Level is HIGH 1'b1: Active Interrupt Polarity Level is LOW

SPI_IMR
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM Receive FIFO Full Interrupt Mask. 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM Receive FIFO Overflow Interrupt Mask. 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM Receive FIFO Underflow Interrupt Mask. 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM Transmit FIFO Overflow Interrupt Mask. 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM Transmit FIFO Empty Interrupt Mask. 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS Receive FIFO Full Interrupt Status. 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS Receive FIFO Overflow Interrupt Status. 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking

Bit	Attr	Reset Value	Description
2	RO	0x0	RFUIS Receive FIFO Underflow Interrupt Status. 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS Transmit FIFO Overflow Interrupt Status. 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS Transmit FIFO Empty Interrupt Status. 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFRIS Receive FIFO Full Raw Interrupt Status. 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS Receive FIFO Overflow Raw Interrupt Status. 1'b0 = spi_rxo_intr interrupt is not active prior to masking 1'b1 = spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS Receive FIFO Underflow Raw Interrupt Status. 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking
1	RO	0x0	TFORIS Transmit FIFO Overflow Raw Interrupt Status. 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking
0	RO	0x1	TFERIS Transmit FIFO Empty Raw Interrupt Status. 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	CTFOI Clear Transmit FIFO Overflow Interrupt. Write 1 to Clear Transmit FIFO Overflow Interrupt
2	RW	0x0	CRFOI Clear Receive FIFO Overflow Interrupt. Write 1 to Clear Receive FIFO Overflow Interrupt
1	WO	0x0	CRFUI Clear Receive FIFO Underflow Interrupt. Write 1 to Clear Receive FIFO Underflow Interrupt
0	WO	0x0	CCI Clear Combined Interrupt. Write 1 to Clear Combined Interrupt

SPI_DMACR
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE Transmit DMA Enable. 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE Receive DMA Enable. 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1

SPI_DMARDLR
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	RDL Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1

SPI_TXDR
Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR Transimt FIFO Data Register. When it is written to, data are moved into the transmit FIFO

SPI_RXDR
Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR Receive FIFO Data Register. When the register is read, data in the receive FIFO is accessed

17.5 Interface Description

Table 17-11SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_clk	I/O	IO_FLASHrdn_UART3rxm1_SPI0clk_GPIO1B7vccio0	GRF_GPIO1B_IOMUX_H[2:0]=3'b11
spi0_rxd	I	IO_FLASHcle_UART3ctsm1_SPI0mosi_I2C3sda_GPIO1B4vccio0	SPI Slave mode: GRF_GPIO1B_IOMUX_H[6:4]=3'b11
	I	IO_FLASHwrn_UART3rtsm1_SPI0miso_I2C3scl_GPIO1B5vccio0	SPI Master mode: GRF_GPIO1B_IOMUX_H[10:8]=3'b11
spi0_txd	O	IO_FLASHcle_UART3ctsm1_SPI0mosi_I2C3sda_GPIO1B4vccio0	SPI Master mode: GRF_GPIO1B_IOMUX_H[6:4]=3'b11
	O	IO_FLASHwrn_UART3rtsm1_SPI0miso_I2C3scl_GPIO1B5vccio0	SPI Slave mode: GRF_GPIO1B_IOMUX_H[10:8]=3'b11
spi0_csn0	I/O	IO_FLASHcs1_UART3txm1_SPI0csn_GPIO1B6vccio0	GRF_GPIO1B_IOMUX_H[14:12]=3'b11
spi1_clk	I/O	IO_LCDCd11m0_I2S08ch_sdo2_CIFd9m1_SPI1clk_GPIO3B7vccio4	GRF_GPIO3B_IOMUX_H[14:12]=3'b11
spi1_rxd	I	IO_LCDCd8m0_I2S08ch_sclkrx_C	SPI Slave mode:

Module Pin	Direction	Pad Name	IOMUX Setting
		IFd7m1_SPI1mosi_GPIO3B4vccio4	GRF_GPIO3B_IOMUX_H[2:0]=3'b11
	I	IO_LCDCd10m0_I2S08ch_sdo3_CIFd8m1_SPI1miso_GPIO3B6vccio4	SPI Master mode: GRF_GPIO3B_IOMUX_H[10:8]=3'b11
spi1_txd	O	IO_LCDCd8m0_I2S08ch_sclkrx_CIFd7m1_SPI1mosi_GPIO3B4vccio4	SPI Master mode: GRF_GPIO3B_IOMUX_H[2:0]=3'b11
	O	IO_LCDCd10m0_I2S08ch_sdo3_CIFd8m1_SPI1miso_GPIO3B6vccio4	SPI Slave mode: GRF_GPIO3B_IOMUX_H[10:8]=3'b11
spi1_csn0	O	IO_LCDCd5m0_I2S08ch_sdi2_CIFd6m1_SPI1csn_GPIO3B1vccio4	GRF_GPIO3B_IOMUX_L[6:4]=3'b11
spi1_csn1	O	IO_LCDCd6_SPI1csn1_GPIO3B2vccio4	GRF_GPIO3B_IOMUX_L[10:8]=3'b11

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

17.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,
When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)
When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.
When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

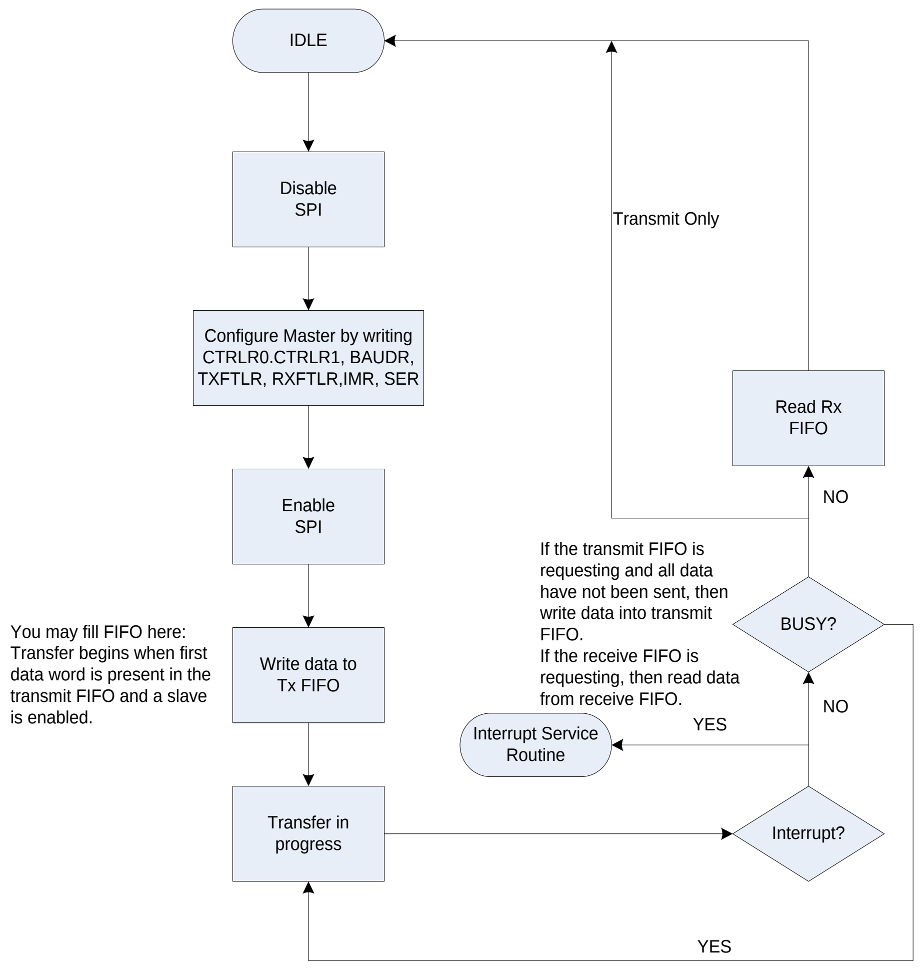

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, BAUDR, TXFTLR, RXFTLR,IMR, SER
Enable SPI
You may fill FIFO here: Transfer begins when first data word is present in the transmit FIFO and a slave is enabled.
Write data to Tx FIFO
Transfer in progress
Interrupt?
YES
Interrupt Service Routine
NO
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.
BUSY?
YES
NO
Read Rx FIFO
Transmit Only

Fig. 17-7SPI Master transfer flow diagram

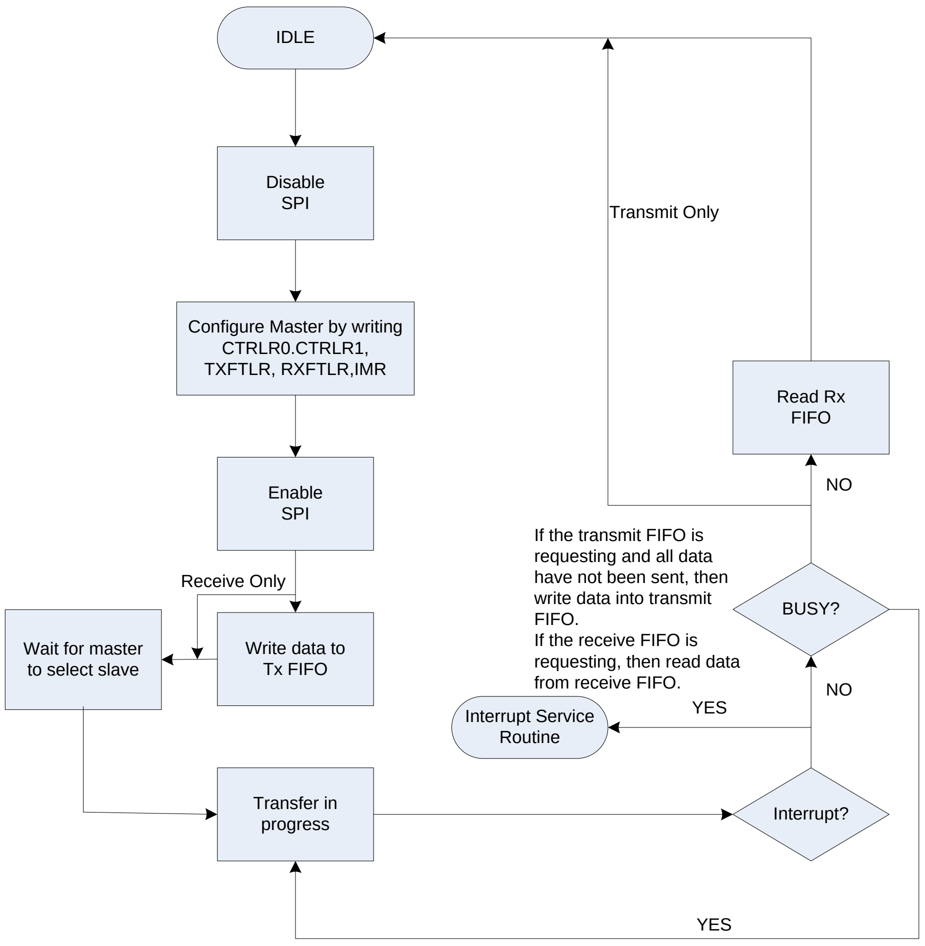

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, TXFTLR, RXFTLR,IMR
Enable SPI
Receive Only
Wait for master to select slave
Write data to Tx FIFO
Transfer in progress
Interrupt?
YES
Interrupt Service Routine
NO
BUSY?
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.
NO
Read Rx FIFO
Transmit Only
YES

Fig. 17-8SPI Slave transfer flow diagram

Chapter 18 UART

18.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.
UART Controller supports the following features:

- Support 6 independent UART controller: UART0~UART5
- contain two 64Bytes FIFOs for data receive and transmit
- support auto flow-control
- Support bit rates 115.2Kbps,460.8Kbps,921.6Kbps,1.5Mbps,3Mbps, 4Mbps
- Support programmable baud rates, even with non-integer clock divider
- Standard asynchronous communication bits (start, stop and parity)
- Support interrupt-based or DMA-based mode
- Support 5-8 bits width transfer

18.2 Block Diagram

This section provides a description about the functions and behavior under various conditions. The UART Controller comprises with:

- AMBA APB interface
- FIFO controllers
- Register block
- Modem synchronization block and baud clock generation block
- Serial receiver and serial transmitter

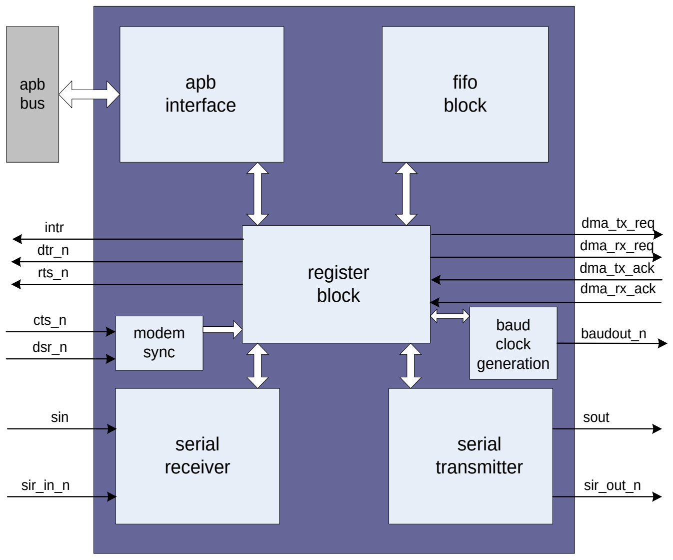


Fig. 18-1UART Architecture

APB INTERFACE
The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.
Register block
Be responsible for the main UART functionality including control, status and interrupt generation.
Modem Synchronization block
Synchronizes the modem input signal.
FIFO block
Be responsible for FIFO control and storage (when using internal RAM) or signaling to

control external RAM (when used).

Baud Clock Generator

Generates the transmitter and receiver baud clockalong with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.

Serial Receiver

Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

18.3 Function Description

UART (RS232) Serial Protocol

Because the serial communication is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to perform simple error checking on the received data, as shown in Figure.

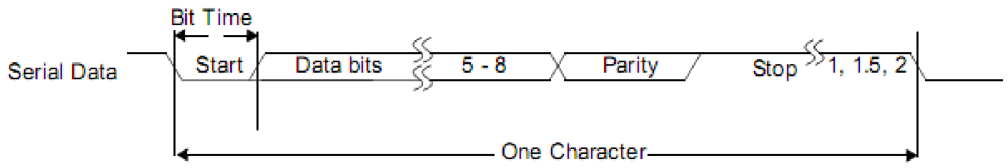


Fig. 18-2UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional datacommunications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

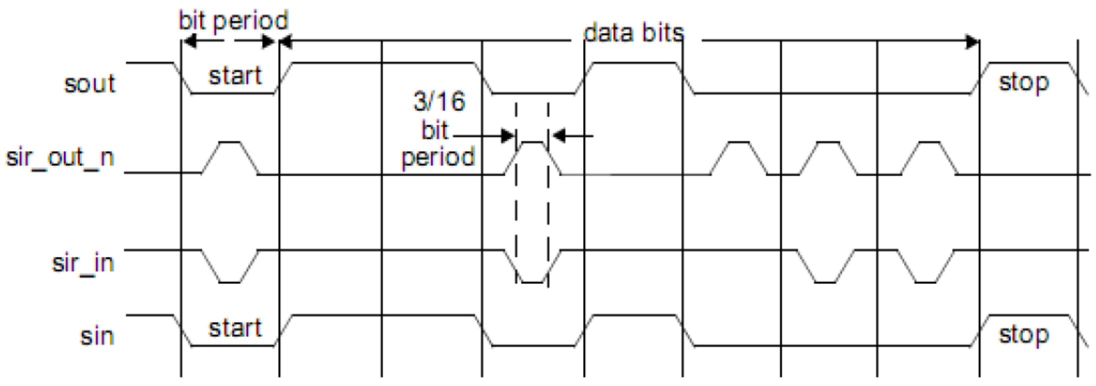


Fig. 18-3IrDA 1.0

Baud Clock

The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

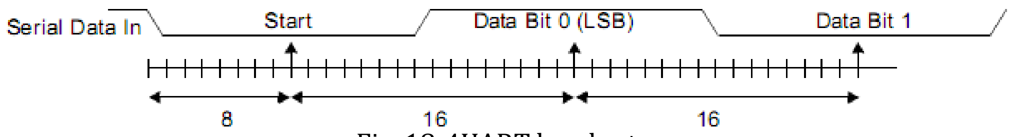


Fig. 18-4UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART0/UART1/UART2/UART3/UART4/UART5 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The UART supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

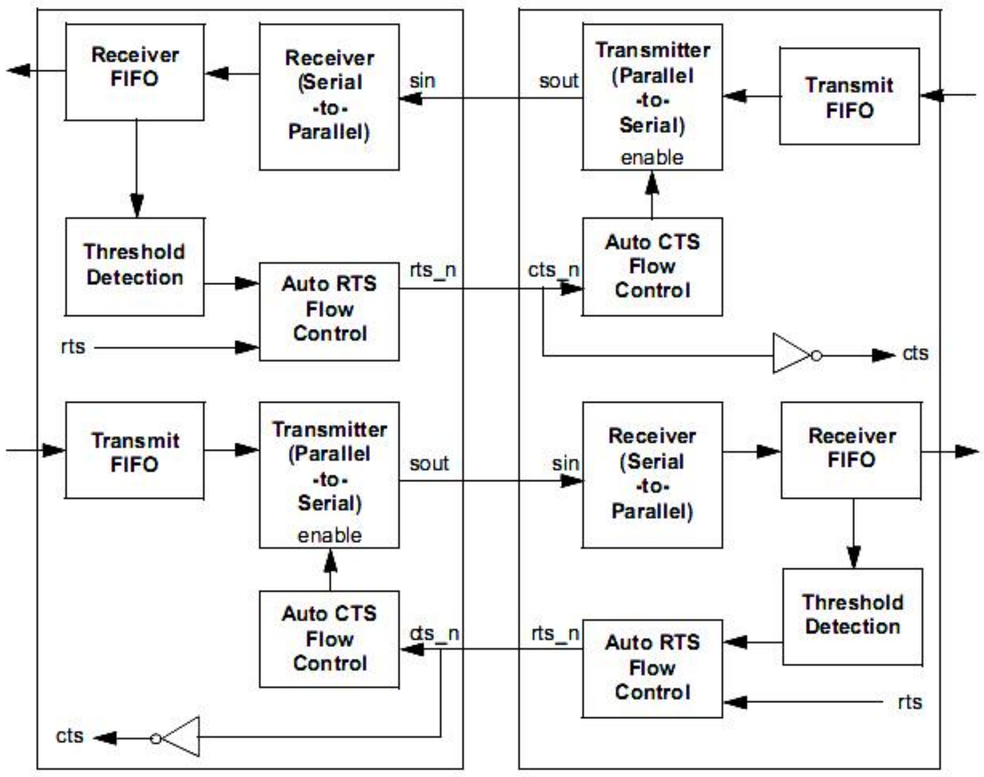


Fig. 18-5UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

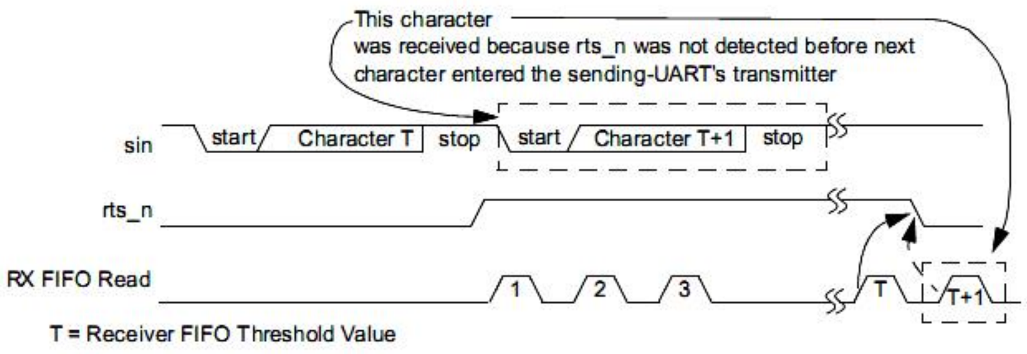

This character was received because rts_n was not detected before next character entered the sending-UART's transmitter
sin
start
Character T
stop
start
Character T+1
stop
rts_n
RX FIFO Read
1
2
3
T
T+1
T = Receiver FIFO Threshold Value

Fig. 18-6UART AUTO RTS TIMING

Auto CTS – becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit is set)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit is not set)

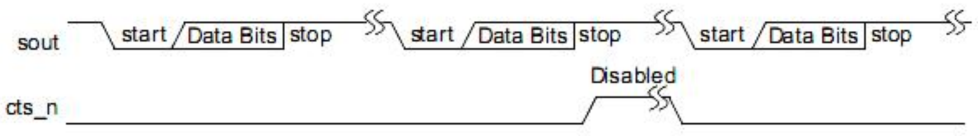


Fig. 18-7UART AUTO CTS TIMING

18.4 Register Description

This section describes the control/status registers of the design

18.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000001	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000060	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000006	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x3330382a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

18.4.2 Detail Register Description

UART_RBR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs

UART_THR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost

UART_DLL
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest Uart clock should be allowed to pass before transmitting or receiving data

UART_DLH
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART

UART_IER
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
6:4	RO	0x0	reserved
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

UART_IIR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
5:4	RO	0x0	reserved
3:0	RO	0x1	int_id Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

UART_FCR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than ful
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full
3	WO	0x0	dma_mode DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 0 = mode 0 1 = mode 11100 = character timeout
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit

Bit	Attr	Reset Value	Description
0	WO	0x0	fifo_en FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset

UART_LCR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers
6	RW	0x0	break_ctrl Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low
5	RO	0x0	reserved
4	RW	0x0	even_parity_sel Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked
3	RW	0x0	parity_en Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	stop_bits_num Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, twostop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
1:0	RW	0x0	data_length_sel Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART_MCR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	sir_mode_en SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode . 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes

Bit	Attr	Reset Value	Description
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)
2	RW	0x0	out1 OUT1
1	RW	0x0	req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data
0	RW	0x0	data_terminal_ready Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

UART_LSR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	RO	0x1	trans_empty Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty

Bit	Attr	Reset Value	Description
5	RO	0x1	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

UART_MSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	data_carrior_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read

UART_SCR
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space

UART_SRBR
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs

UART_STHR
Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR

UART_FAR
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled

UART_TFR
Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	trans_fifo_read Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO

UART_RFW
Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one)
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one)
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR

UART_USR
Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full

Bit	Attr	Reset Value	Description
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty
2	RO	0x1	trasn_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	RO	0x1	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full
0	RO	0x0	uart_busy UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the uart is idle or inactive. 0 = Uart is idle or inactive 1 = Uart is busy (actively transferring data)

UART_TFL
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO

UART_RFL
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This is indicates the number of data entries in the receive FIFO

UART_SRR
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2])
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1])
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the Uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset

UART_SRTS
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR

UART_SBCR
Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR

UART_SDMAM
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3])

UART_SFE
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0])

UART_SRT
Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6])

UART_STET
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4])

UART_HTX
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled

UART_DMASA
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition

UART_CPR
Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	RO	0x0	reserved
13	RO	0x0	DMA_EXTRA 0 = FALSE 1 = TRUE
12	RO	0x0	UART_ADD_ENCODED_PARAMS 0 = FALSE 1 = TRUE
11	RO	0x0	SHADOW 0 = FALSE 1 = TRUE
10	RO	0x0	FIFO_STAT 0 = FALSE 1 = TRUE
9	RO	0x0	FIFO_ACCESS 0 = FALSE 1 = TRUE
8	RO	0x0	NEW_FEAT 0 = FALSE 1 = TRUE
7	RO	0x0	SIR_LP_MODE 0 = FALSE 1 = TRUE
6	RO	0x0	SIR_MODE 0 = FALSE 1 = TRUE
5	RO	0x0	THRE_MODE 0 = FALSE 1 = TRUE
4	RO	0x0	AFCE_MODE 0 = FALSE 1 = TRUE
3:2	RO	0x0	reserved
1:0	RO	0x0	APB_DATA_WIDTH 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

UART_UCV
Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:0	RO	0x3330382a	ver ASCII value for each number in the version

UART_CTR
Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code

18.5 Interface Description

Table 18-1UART Interface Description

Modulepin	Dir	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	IO_UART0rx_PMUdebug1_GPIO0B3pmuio2	GRF_GPIO0B_IOMUX[7:6]=2’b01
uart0_sout	O	IO_UART0tx_PMUdebug0_GPIO0B2pmuio2	GRF_GPIO0B_IOMUX[5:4]=2’b01
uart0_cts_n	I	IO_UART0cts_PMUdebug2_PMUdebug_sout_GPIO0B4pmuio2	GRF_GPIO0B_IOMUX[9:8]=2’b01
uart0_rts_n	O	IO_UART0rts_TESTclk1_GPIO0B5pmuio2	GRF_GPIO0B_IOMUX[11:10]=2’b01
UART1 Interface			
uart1_sin	I	IO_UART1rx_GPIO1C0vccio1	GRF_GPIO1C_IOMUX_L[2:0]=3’b001
uart1_sout	O	IO_UART1tx_GPIO1C1vccio1	GRF_GPIO1C_IOMUX_L[6:4]= 3’b001
uart1_cts_n	I	IO_UART1cts_GPIO1C2vccio1	GRF_GPIO1C_IOMUXL[10:8]=3’b001
uart1_rts_n	O	IO_UART1rts_GPIO1C3vccio1	GRF_GPIO1C_IOMUX_L[14:12]=3’b001
UART2m0 Interface			
uart2m0_sin	I	IO_SDMMC0d1_UART2rxm0_GPIO1D3vccio2	GRF_GPIO1D_IOMUX_L[14:12]=3’b010
uart2m0_sout	O	IO_SDMMC0d0_UART2txm0_GPIO1D2vccio2	GRF_GPIO1D_IOMUX_L[10:8]= 3’b010
UART2m1 Interface			
uart2m1_sin	I	IO_CIFd1m0_UART2rxm1_GPIO2B6vccio3	GRF_GPIO2B_IOMUX_H[10:8]=3’b010
uart2m1_sout	O	IO_CIFd0m0_UART2txm1_GPIO2B4vccio3	GRF_GPIO2B_IOMUX_H[2:0]=3’b010
UART3m0 Interface			
uart3m0_sin	I	IO_PWM3_UART3rxm0_PMUdebug4_GPIO0C1pmuio2	GRF_GPIO0C_IOMUX[3:2]=2’b10

Modulepin	Dir	Pad name	IOMUX
uart3m0_sout	O	IO_PWM1_UART3txm0_PMUdebug3_GPIO0C0pmuio2	GRF_GPIO0C_IOMUX[1:0]= 2'b10
uart3m0_cts_n	I	IO_I2C1scl_UART3ctsm0_PMUdebug5_GPIO0C2pmuio2	GRF_GPIO0C_IOMUX[5:4]=2'b10
uart3m0_rts_n	O	IO_I2C1sda_UARTrtsm0_GPIO0c3pmuio2	GRF_GPIO0C_IOMUX[7:6]= 2'b10
UART3m1 Interface			
uart3m1_sin	I	IO_FLASHrdn_UART3rxm1_SPI0clk_GPIO1B7vccio0	GRF_GPIO1B_IOMUX_H[14:12]=3'b010
uart3m1_sout	O	IO_FLASHcs1_UART3txm1_SPI0csn_GPIO1B6vccio0	GRF_GPIO1B_IOMUX_H[10:8]= 3'b010
uart3m1_cts_n	I	IO_FLASHcle_UART3ctsm1_SPI0mosi_I2C3sda_GPIO1B4vccio0	GRF_GPIO1B_IOMUX_H[2:0]=3'b010
uart3m1_rts_n	O	IO_FLASHwrn_UART3rtsm1_SPI0miso_I2C3scl_GPIO1B5vccio0	GRF_GPIO1B_IOMUX_H[6:4]= 3'b010
UART4 Interface			
uart4_sin	I	IO_SDMMC0d2_UART4rx_JTAGtck_GPIO1D4vccio2	GRF_GPIO1D_IOMUX_H[2:0]=3'b010
uart4_sout	O	IO_SDMMC0d3_UART4tx_JTAGtms_GPIO1D5vccio2	GRF_GPIO1D_IOMUX_H[6:4]= 3'b010
uart4_cts_n	I	IO_SDMMC0clkout_UART4cts_TESTclk0_GPIO1D6vccio2	GRF_GPIO1D_IOMUX_H[10:8]=3'b010
uart4_rts_n	O	IO_SDMMC0cmd_UART4rts_GPIO1D7vccio2	GRF_GPIO1D_IOMUX_H[14:12]=3'b010
UART5 Interface			
uart5_sin	I	IO_LCDChsyncm0_I2S22ch_mclk_CIFd0m1_UART5rx_GPIO3A1vccio4	GRF_GPIO3A_IOMUX_L[6:4]=3'b100
uart5_sout	O	IO_LCDCvsyncm0_I2S22ch_sclk_CIFd1m1_UART5tx_GPIO3A2vccio4	GRF_GPIO3A_IOMUX_L[10:8]= 3'b100
uart5_cts_n	I	IO_LCDDdenm0_I2S22ch_lrck_CIFd2m1_UART5cts_GPIO3A3vccio4	GRF_GPIO3A_IOMUX_L[14:12]=3'b100
uart5_rts_n	O	IO_LCDCd1m0_I2S22ch_sdi_CIFd3m1_UART5rts_GPIO3A5vccio4	GRF_GPIO3A_IOMUX_H[6:4]= 3'b100

The I/O interface of UART2 can be chosen by setting GRF_IOFUNC_SEL0[10]bit, if this bit is set to 1, UART2 uses the UART2m1 I/O interface. The I/O interface of UART3 can be sen by setting GRF_IOFUNC_SEL0[9]bit, if this bit is set to 1, UART3 uses the UART3m1 I/O interface.

18.6 Application Notes

18.6.1 None FIFO Mode Transfer Flow

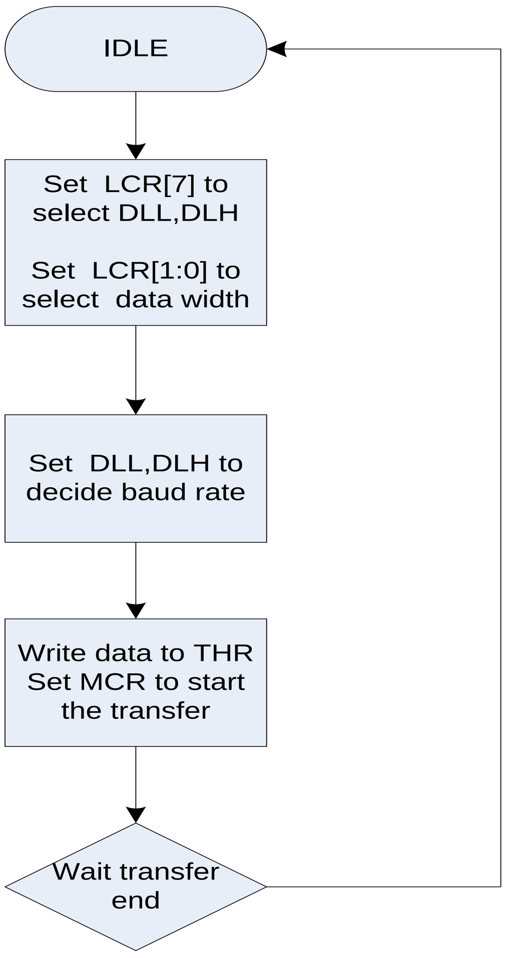

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set DLL,DLH to decide baud rate
Write data to THR Set MCR to start the transfer
Wait transfer end

Fig. 18-8UART none fifo mode

18.6.2 FIFO Mode Transfer Flow

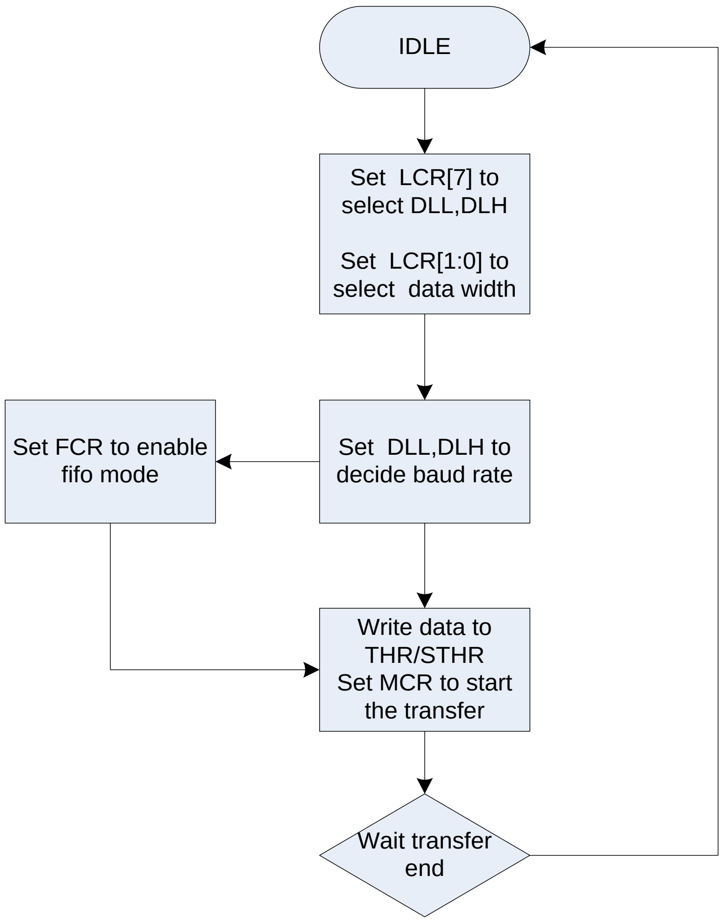

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set FCR to enable fifo mode
Set DLL,DLH to decide baud rate
Write data to THR/STHR
Set MCR to start the transfer
Wait transfer end

Fig. 18-9UART fifo mode

The UART is an APB slave performing:
Serial-to-parallel conversion on data received from a peripheral device.
Parallel-to-serial conversion on data transmitted to the peripheral device.
The CPU reads and writes data and control/status information through the APB interface.
The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 64-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

18.6.3 Baud Rate Calculation

UART clock generation
The following figures shows the UART clock generation.UART0~5 source clocks can be selected from four PLL outputs (XIN_OSC0_FUNC/GPLL_CLK_MUX/USBPHY480M_MUX/ NPLL_CLK_MUX).
UART clocks can be generated by 1 to 32 division of its source clock, or can be fractionally divided again.

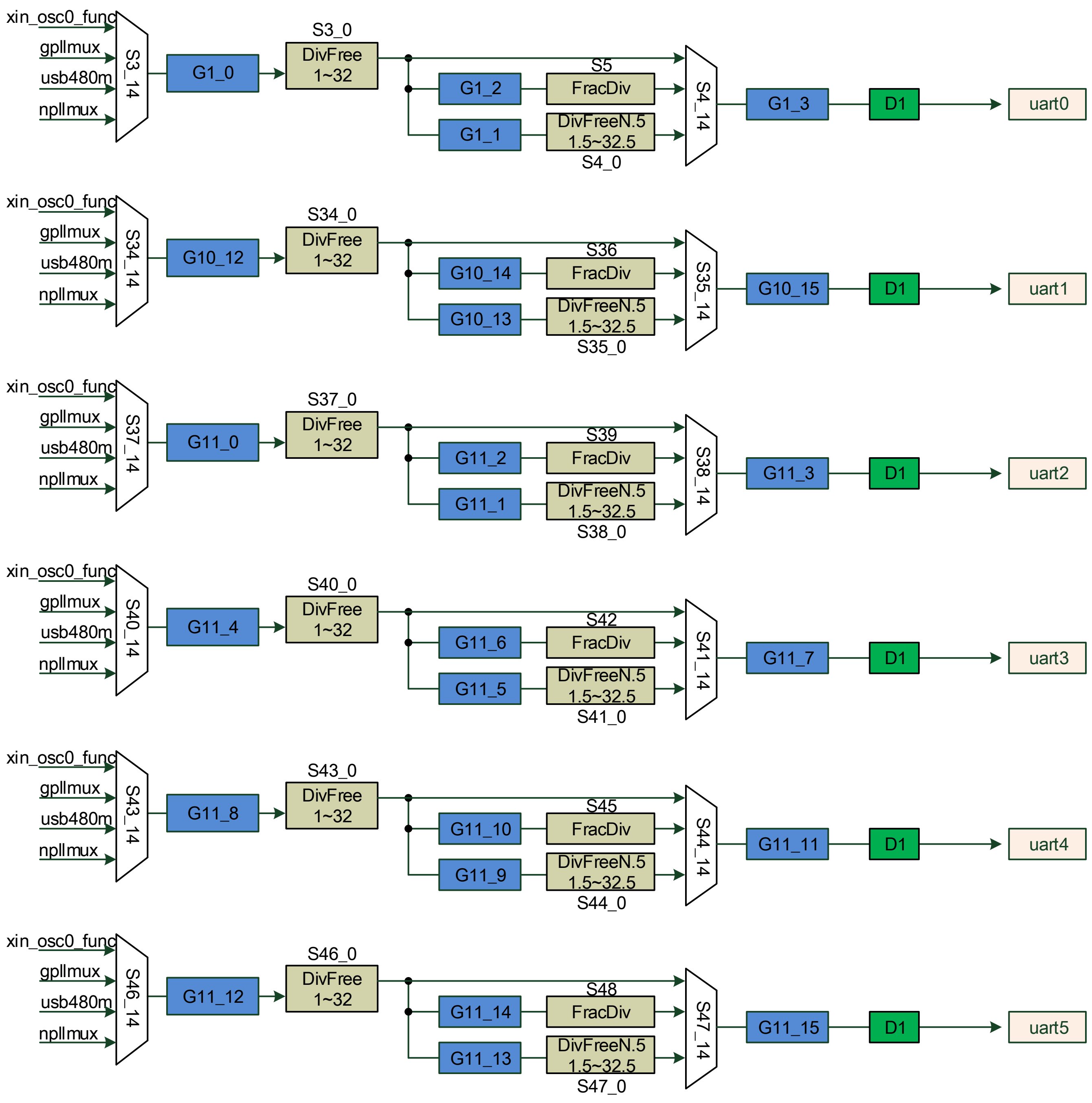


Fig. 18-10UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 18-2 UART baud rate configuration

Baud Rate	Reference Configuration

115.2 Kbps	Configure GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/72 to get 1.8432MHz clock;
	Configure UART_DLL to 1.

460.8 Kbps	Configure GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/288 to get 7.3728MHz clock;

Baud Rate	Reference Configuration
	Configure UART_DLL to 1.

921.6 Kbps	Configure GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/576 to get 14.7456MHz clock;
	Configure UART_DLL to 1.

1.5 Mbps	Choose GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 50 to get 24MHz clock;
	Configure UART_DLL to 1.

3 Mbps	Choose GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 1200/48 to get 48MHz clock;
	Configure UART_DLL to 1.

4 Mbps	Configure GENERAL PLL to get 1200MHz clock output;
	Divide 1200MHz clock by 480/7.5 to get 64MHz clock;
	Configure UART_DLL to 1.

18.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

- When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, lowactive.
- When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, lowactive.

Table 18-3 UART cts_n and rts_n polarity configuration

UART	GRF_UART_CTS_SEL	GRF_UART_RTS_SEL
UART0	PMUGRF_SOC_CON0[6]	PMUGRF_SOC_CON0[5]
UART1	GRF_SOC_CON2[2]	GRF_SOC_CON2[3]
UART2	GRF_SOC_CON2[4]	GRF_SOC_CON2[5]
UART3	GRF_SOC_CON2[6]	GRF_SOC_CON2[7]
UART4	GRF_SOC_CON2[8]	GRF_SOC_CON2[9]
UART5	GRF_SOC_CON2[10]	GRF_SOC_CON2[11]

Chapter 19 SAR-ADC

19.1 Overview

The SAR-ADC is a 3-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 13MHz A/D converter clock.

19.2 Block Diagram

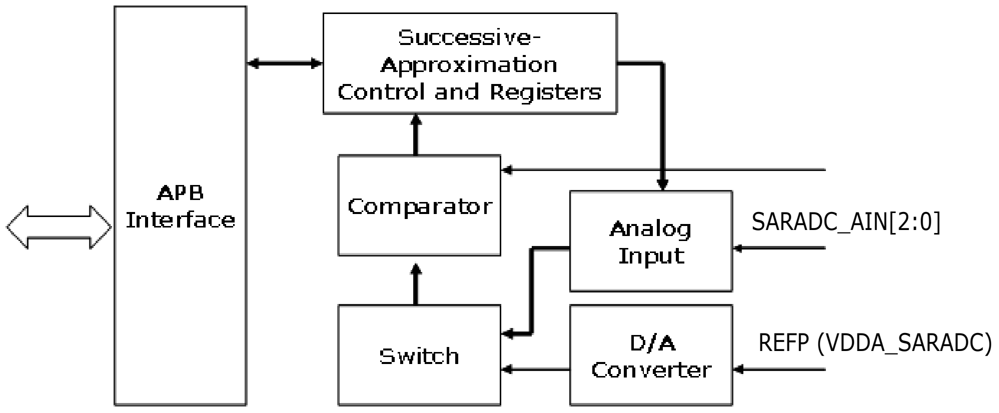


Fig.19-1SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block
This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.
Comparator Block
This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

19.3 Function Description

19.3.1 APB Interface

In PX30, SAR-ADC works at single-sample operation mode.
This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC.The detailed timing diagram will be shown in the following.

19.4 Register description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

19.4.2 Detail Register Description

SARADC_DATA
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data

SARADC_STAS
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status 0: ADC stop 1: Conversion in progress

SARADC_CTRL
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt
5	RW	0x0	int_en 0: Disable 1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl 0: ADC power down; 1: ADC power up and reset. start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up
2:0	RW	0x0	adc_input_src_sel 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) 010 : Input source 2 (SARADC_AIN[2]) 011 : Input source 3 (SARADC_AIN[3]) 100 : Input source 4 (SARADC_AIN[4]) 101 : Input source 5 (SARADC_AIN[5]) Others : Reserved

SARADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

19.5 Timing Diagram

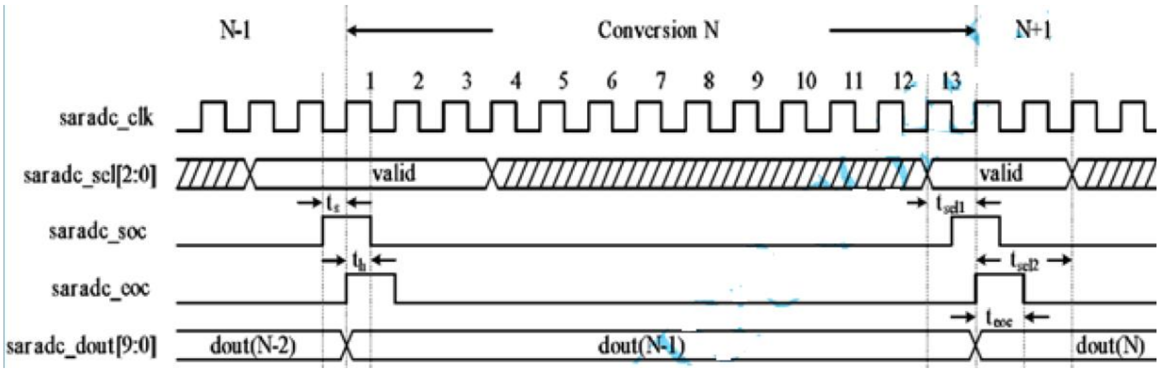


Fig.19-2SAR-ADC timing diagram in single-sample conversion mode

19.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]

Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 20 Temperature-Sensor ADC(TS-ADC)

20.1 Overview

TS-ADC Controller module supports user-defined mode and automatic mode.User-defined mode refers, TSADC all the control signals entirely by software writing to register for direct control.Automatic mode refers to the module automatically poll TSADC output, and the results were checked. If you find that the temperatureHigh in a period of time, an interrupt is generated to the processor down-measures taken; if the temperature over a period of timeHigh, the resulting TSHUT gave CRU module, let it reset the entire chip, or via GPIO give PMIC.
TS-ADC Controller supports the following features:

- Support User-Defined Mode and Automatic Mode
- In User-Defined Mode, start_of_conversion can be controlled completely by software, and also can be generated by hardware.
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 2 channel TS-ADC, the temperature criteria can be configurable
- In Automatic Mode, the time interval of temperature detection can be configurable
- In Automatic Mode, when detecting a high temperature, the time interval of temperature detection can be configurable
- High temperature denounce can be configurable
- 10-bit SARADC up to 50KS/s sampling rate

20.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

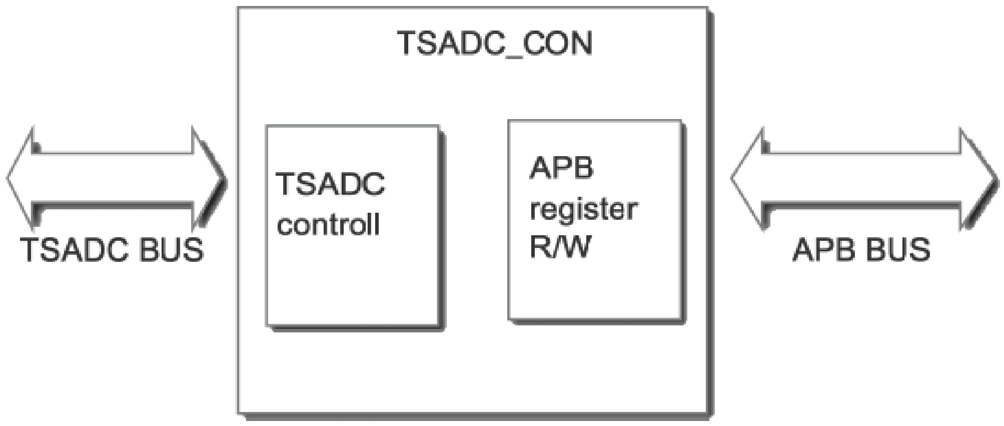


Fig.20-1 TS-ADC Controller Block Diagram

20.3 Function Description

20.3.1 APB Interface

There is an APB Slave interface in TS-ADC Controller, which is used to configure the TS-ADC Controller registers and look up the temperature from the temperature sensor.

20.3.2 TS-ADC Controller

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.This block compares the analog input with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

20.4 Register description

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000208	The control register of A/D converter
TSADC_AUTO_CON	0x0004	W	0x00000000	ADC auto mode control register
TSADC_INT_EN	0x0008	W	0x00000000	interrupt enable register
TSADC_INT_PD	0x000c	W	0x00000000	int_pd register
TSADC_DATA0	0x0020	W	0x00000000	This register contains the data after A/D conversion
TSADC_DATA1	0x0024	W	0x00000000	This register contains the data after A/D conversion
TSADC_COMP0_INT	0x0030	W	0x00000000	ADC high level for source 0 interrupt
TSADC_COMP1_INT	0x0034	W	0x00000000	ADC high level for source 1 interrupt
TSADC_COMP0_SHUT	0x0040	W	0x00000000	ADC high level for source 0 shut
TSADC_COMP1_SHUT	0x0044	W	0x00000000	ADC high level for source 1 shut
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	high temperature / voltage debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	high temperature / voltage debounce
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	ADC auto access period
TSADC_AUTO_PERIOD_HT	0x006c	W	0x00010000	ADC auto access period when ADC result is high
TSADC_COMP0_LOW_INT	0x0080	W	0x00000000	ADC low level for source 0
TSADC_COMP1_LOW_INT	0x0084	W	0x00000000	ADC low level for source 1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

20.4.2 Detail Register Description

TSADC_USER_CON
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	adc_status 0: ADC stop; 1: Conversion in progress
11:6	RW	0x08	inter_pd_soc interleave between power down and start of conversion

Bit	Attr	Reset Value	Description
5	RW	0x0	start When software write 1 to this bit , start_of_conversion will be assert. This bit will be cleared after ADC access finishing. When ADC_USER_CON[4] = 1'b1 take effect
4	RW	0x0	start_mode start mode. 0: adc controller will asert start_of_conversion after "inter_pd_soc" cycles. 1: the start_of_conversion will be controlled by ADC_USER_CON[5]
3	RW	0x1	adc_power_ctrl 0: ADC power down; 1: ADC power up and reset
2:0	RW	0x0	adc_input_src_sel 000 : Input source 0 (ADC_AIN[0]) 001 : Input source 1 (ADC_AIN[1]) 010 : Input source 2 (ADC_AIN[2]) 011 : Input source 3 (ADC_AIN[3]) Others : Reserved

TSADC_AUTO_CON
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	last_tshut_2cru TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset
24	RW	0x0	last_tshut_2gpio TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset
23:18	RO	0x0	reserved
17	RO	0x0	sample_dly_sel 0: AUTO_PERIOD is used. 1: AUTO_PERIOD_HT is used
16	RO	0x0	auto_status 0: auto mode stop; 1: auto mode in progress
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	src1_lt_en 0: do not care low level of source 1 1: enable the low level monitor of source 1
12	RW	0x0	src0_lt_en 0: do not care low level of source 0 1: enable the low level monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_prolarity 0: low active 1: high active
7:6	RO	0x0	reserved
5	RW	0x0	src1_en channel 1 enable. 0: do not monitor channel 1 result 1: monitor channel 1 in turn
4	RW	0x0	src0_en channel 0 enable. 0: do not monitor channel 0 result 1: monitor channel 0 in turn
3:2	RO	0x0	reserved
1	RW	0x0	adc_q_sel adc data select 0: adc_q 1: 4096 - adc_q
0	RW	0x0	auto_en 0: ADC controller works at user-define mode 1: ADC controller works at auto mode

TSADC_INT_EN
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_en eoc interrupt enable in user defined mode 0: disable; 1: enable
15:14	RO	0x0	reserved
13	RW	0x0	lt_inten_src1 low temperature interrupt enable for src1 0: disable 1: enable

Bit	Attr	Reset Value	Description
12	RW	0x0	lt_inten_src0 low temperature interrupt enable for src0 0: disable 1: enable
11:10	RO	0x0	reserved
9	RW	0x0	tshut_2cru_en_src1 0: TSHUT output to cru disabled. TSHUT output will always keep low. 1: TSHUT output works
8	RW	0x0	tshut_2cru_en_src0 0: TSHUT output to cru disabled. TSHUT output will always keep low. 1: TSHUT output works
7:6	RO	0x0	reserved
5	RW	0x0	tshut_2gpio_en_src1 0: TSHUT output to gpio disabled. TSHUT output will always keep low. 1: TSHUT output works
4	RW	0x0	tshut_2gpio_en_src0 0: TSHUT output to gpio disabled. TSHUT output will always keep low. 1: TSHUT output works
3:2	RO	0x0	reserved
1	RW	0x0	ht_inten_src1 high temperature interrupt enable for src1 0: disable 1: enable
0	RW	0x0	ht_inten_src0 high temperature interrupt enable for src0 0: disable 1: enable

TSADC_INT_PD
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_pd This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	lt_irq_src1 When ADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared
12	RW	0x0	lt_irq_src0 When ADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared
11:6	RO	0x0	reserved
5	RW	0x0	tshut_o_src1 TSHUT output status When ADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared
4	RW	0x0	tshut_o_src0 TSHUT output status When ADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared
3:2	RO	0x0	reserved
1	RW	0x0	ht_irq_src1 When ADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared
0	RW	0x0	ht_irq_src0 When ADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared

TSADC_DATA0
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[9:0])

TSADC_DATA1
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 1 last conversion (DOUT[9:0])

TSADC_COMP0_INT
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src0 ADC high level for channel 0. ADC output is bigger than adc_comp, means the temperature is high. ADC_HT_INT will be valid

TSADC_COMP1_INT
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src1 ADC high level for channel 1. ADC output is bigger than adc_comp, means the temperature is high. ADC_HT_INT will be valid

TSADC_COMP0_SHUT
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src0 ADC high level for channel 0 to generate TSHUT. ADC output is bigger than adc_comp, TSHUT will be valid

TSADC_COMP1_SHUT
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src1 ADC high level for channel 1 to generate TSHUT. ADC output is bigger than adc_comp, TSHUT will be valid

TSADC_HIGHT_INT_DEBOUNCE
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce ADC controller will only generate interrupt when temperature / voltage is higher than COMP_INT for "debounce" times

TSADC_HIGHT_TSHUT_DEBOUNCE
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce ADC controller will only generate TSHUT when temperature / voltage is higher than COMP_SHUT for "debounce" times

TSADC_AUTO_PERIOD
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period when auto mode is enabled, this register controls the interleave between every two accessing of ADC

TSADC_AUTO_PERIOD_HT
Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every two accessing of ADC after the temperature is higher than COMP_SHUT or COMP_INT

TSADC_COMP0_LOW_INT
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src0 ADC low level. ADC output is lower than adc_comp, means the temperature is low. ADC_LOW_INT will be valid

TSADC_COMP1_LOW_INT
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	adc_comp_src1 ADC low level. ADC output is lower than adc_comp, means the temperature is low. ADC_LOW_INT will be valid

20.5 Application Notes

20.5.1 Single-sample conversion

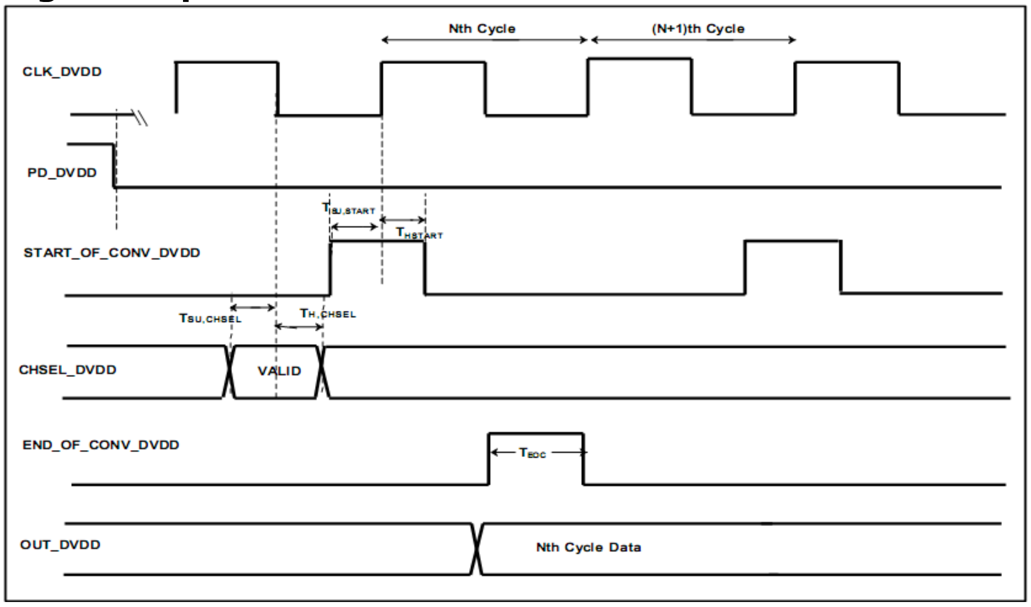


Fig. 20-2the start flow to enable the sensor and adc

20.5.2 Temperature-to-code mapping

Table 20-1 Temperature Code Mapping

temp (C)	Code
-40	3800
-35	3792
-30	3783
-25	3774
-20	3765
-15	3756
-10	3747
-5	3737
0	3728
5	3718
10	3708
15	3698
20	3688
25	3678
30	3667
35	3656
40	3645
45	3634
50	3623
55	3611
60	3600
65	3588
70	3575
75	3563
80	3550
85	3537
90	3524
95	3510
100	3496
105	3482
110	3467
115	3452
120	3437
125	3421

Note:
Code to Temperature mapping of the Temperature sensor is a piece wise linear curve. Any temperature, code faling between to 2 give temperatures can be linearly interpolated.
Code to Temperature mapping should be updated based on sillcon results.

20.5.3 User-Define Mode

- In user-define mode, the PD_DVDD and CHSEL_DVDD are generate by setting register TSADC_USER_CON, bit[3] and bit[2:0]. In order to ensure timing between PD_DVDD and CHSEL_DVDD, the CHSEL_DVDD must be set before the PD_DVDD.
- In user-define mode, you can choose the method to control the START_OF_CONVERSION by setting bit[4] of TSADC_USER_CON. If set to 0, the start_of_conversion will be assert after "inter_pd_soc" cycles, which could be set by bit[11:6] of TSADC_USER_CON. And if start_mode was set 1, the start_of_conversion will be controlled by bit[5] of TSADC_USER_CON.
- Software can get the four channel temperature from TSADC_DATAn (n=0,1,2,3).

20.5.4 Automatic Mode

You can use the automatic mode with the following step:

- Set TSADC_AUTO_PERIOD，configure the interleave between every two accessing of TSADC in normal operation.
- Set TSADC_AUTO_PERIOD_HT. configure the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.
- Set TSADC_COMPn_INT(n=0,1), configure the high temperature level, if tsadc output is smaller than the value, means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0,1), configure the super high temperature level, if tsadc output is smaller than the value, means the temperature is too high, TSHUT will be asserted.

- Set TSADC_INT_EN, you can enable the high temperature interrupt for all channel; and you can also set TSHUT output to gpio to reset the whole chip; and you can set TSHUT output to cru to reset the whole chip.
- Set TSADC_HIGHT_INT_DEBOUNCE and TSADC_HIGHT_TSHUT_DEBOUNCE, if the temperature is higher than COMP_INT or COMP_SHUT for “debounce” times, TSADC controller will generate interrupt or TSHUT.
- Set TSADC_AUTO_CON, enable the TSADC controller.

Chapter 21 GPIO

21.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is an APB slave device.GPIO controls the output data and direction of external I/O pads. It also can read back thedata on external pads using memory-mapped registers.
GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode

21.2 Block Diagram

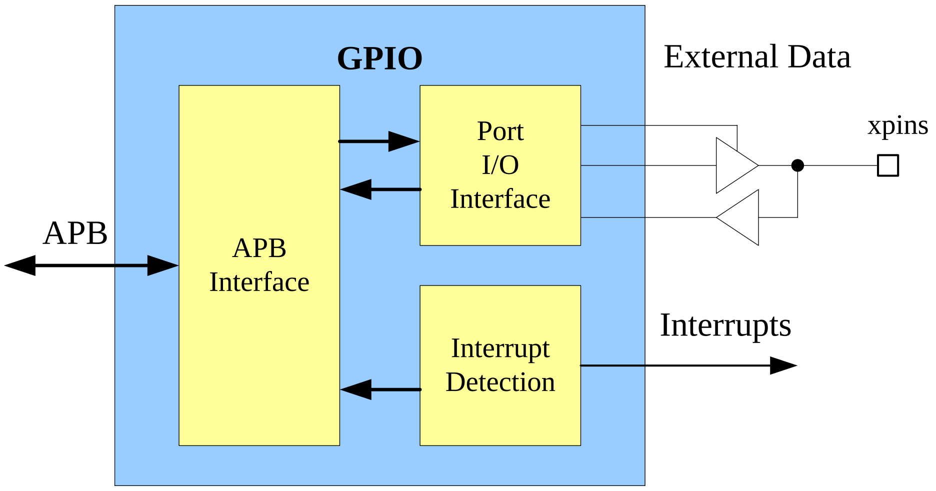


Fig. 21-1GPIO block diagram

Block descriptions:
APB Interface
The APB Interface implements the APB slave operation. Its data bus width is 32 bits.
Port I/O Interface
External data Interface to or from I/O pads.
Interrupt Detection
Interrupt interface to or from interrupt controller.

21.3 Function Description

21.3.1 Operation

Control Mode (software)
Under software control, the data and direction control for the signal aresourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR). The direction of the external I/O pad is controlled by a write to the Porta datadirection register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls thedirection of an external I/O pad.
The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad.External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register(GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only,meaning that it cannot be written from the APB software interface.
Reading External Signals
The data on the GPIO_EXT_PORTA external signal can always be read. The data on the

external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.
An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge
- Both the rising edge and the falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.
The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.
Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost.However, no new interrupts are generated.
For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

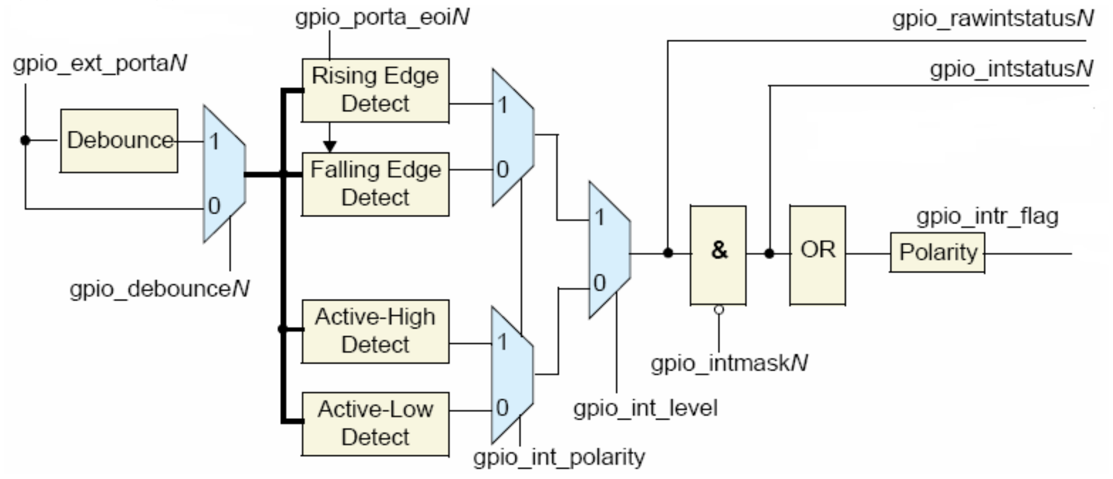


Fig. 21-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.
When input interrupt signals are debounced using a debounce clock (pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one risingedge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization topclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional andunder software control (GPIO_LS_SYNC).

21.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitiveinterrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A inorder to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on alevel-sensitive interrupt.

GPIOs' hierarchy in the chip

GPIO0 is in PD_PMU subsystem, GPIO1/GPIO2/GPIO3 are in PD_BUS subsystem.

21.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 4 GPIOs (GPIO0 ~ GPIO3), and each of them has same register group. Therefore, 4 GPIOs' register groups have 4 different base addresses.

21.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register
GPIO_INT_BOTHEDGE	0x0068	W	0x00000000	Interrupt both edge type

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

21.4.2 Detail Register Description

GPIO_SWPORTA_DR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register

GPIO_SWPORTA_DDR
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 1'b0: Input (default) 1'b1: Output

GPIO_INTEN
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 1'b0: Configure Port A bit as normal GPIO signal (default) 1'b1: Configure Port A bit as interrupt

GPIO_INTMASK
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 1'b0: Interrupt bits are unmasked (default) 1'b1: Mask interrupt

GPIO_INTTYPE_LEVEL
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 1'b0: Level-sensitive (default) 1'b1: Edge-sensitive

GPIO_INT_POLARITY
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 1'b0: Active-low (default) 1'b1: Active-high

GPIO_INT_STATUS
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS
Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 1'b0: No debounce (default) 1'b1: Enable debounce

GPIO_PORTA_EOI
Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 1'b0: No interrupt clear (default) 1'b1: Clear interrupt

GPIO_EXT_PORTA
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A

GPIO_LS_SYNC
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 1'b0: No synchronization to pclk_intr (default) 1'b1: Synchronize to pclk_intr

GPIO_INT_BOTHEDGE
Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	interrupt_both_edge_type Controls the edge type of interrupt that can occur on Port A.Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port A.The values programmed in the registers gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity registers

21.5 Interface Description

Table 21-1 GPIO interface description

Module Pin	Dir	Pad Name	IOMUX Setting
GPIO0 Interface			

Module Pin	Dir	Pad Name	IOMUX Setting
gpio0_porta[7:0]	I/O	GPIO0_A[7:0]	PMUGRF_GPIO0A_IOMUX[15:0]=16’h0
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	PMUGRF_GPIO0B_IOMUX[15:0]=16’h0
gpio0_porta[23:16]	I/O	GPIO0_C[7:0]	PMUGRF_GPIO0C_IOMUX[15:0]=16’h0
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[3:0]	GRF_GPIO1A_IOMUX_L[15:0]=16’h0
		GPIO1_A[7:4]	GRF_GPIO1A_IOMUX_H[15:0]=16’h0
gpio1_porta[15:8]	I/O	GPIO1_B[3:0]	GRF_GPIO1B_IOMUX_L[15:0]=16’h0
		GPIO1_B[7:4]	GRF_GPIO1B_IOMUX_H[15:0]=16’h0
gpio1_porta[23:16]	I/O	GPIO1_C[3:0]	GRF_GPIO1C_IOMUX_L[15:0]=16’h0
		GPIO1_C[7:4]	GRF_GPIO1C_IOMUX_H[15:0]=16’h0
gpio1_porta[31:24]	I/O	GPIO1_D[3:0]	GRF_GPIO1D_IOMUX_L[15:0]=16’h0
		GPIO1_D[7:4]	GRF_GPIO1D_IOMUX_H[15:0]=16’h0
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[3:0]	GRF_GPIO2A_IOMUX_L[15:0]=16’h0
		GPIO2_A[7:4]	GRF_GPIO2A_IOMUX_H[15:0]=16’h
gpio2_porta[15:8]	I/O	GPIO2_B[3:0]	GRF_GPIO2B_IOMUX_L[15:0]=16’h0
		GPIO2_B[7:4]	GRF_GPIO2B_IOMUX_H[15:0]=16’h0
gpio2_porta[23:16]	I/O	GPIO2_C[3:0]	GRF_GPIO2C_IOMUX_L[15:0]=16’h0
		GPIO2_C[7:4]	GRF_GPIO2C_IOMUX_H[15:0]=16’h0
gpio2_porta[31:24]	I/O	GPIO2_D[3:0]	GRF_GPIO2D_IOMUX_L[15:0]=16’h0
		GPIO2_D[7:4]	GRF_GPIO2D_IOMUX_H[15:0]=16’h0
GPIO3 Interface			
gpio3_porta[7:0]	I/O	GPIO3_A[3:0]	GRF_GPIO3A_IOMUX_L[15:0]=16’h0
		GPIO3_A[7:4]	GRF_GPIO3A_IOMUX_H[15:0]=16’h0
gpio3_porta[15:8]	I/O	GPIO3_B[3:0]	GRF_GPIO3B_IOMUX_L[15:0]=16’h0
		GPIO3_B[7:4]	GRF_GPIO3B_IOMUX_H[15:0]=16’h0
gpio3_porta[23:16]	I/O	GPIO3_C[3:0]	GRF_GPIO3C_IOMUX_L[15:0]=16’h0
		GPIO3_C[7:4]	GRF_GPIO3C_IOMUX_H[15:0]=16’h0
gpio3_porta[31:24]	I/O	GPIO3_D[3:0]	GRF_GPIO3D_IOMUX_L[15:0]=16’h0
		GPIO3_D[7:4]	GRF_GPIO3D_IOMUX_H[15:0]=16’h0

21.6 Application Notes

Steps to set GPIO’s direction

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction and Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Default GPIO’s direction is input direction.

Steps to set GPIO’s level

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction.
- Write GPIO_SWPORT_DR[x] as v to set this GPIO’s value.

Steps to get GPIO’s level

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Read from GPIO_EXT_PORT[x] to get GPIO’s value

Steps to set GPIO as interrupt source

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Write GPIO_INTTYPE_LEVEL[x] as v1 and write GPIO_INT_POLARITY[x] as v2 to set interrupt type
- Write GPIO_INTEN[x] as 1 to enable GPIO’s interrupt

Note: Please switch iomux to GPIO mode first!

Chapter 22 I2S/PCM Controller

22.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus. The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.
Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.
Not only I2S but also PCM mode surround audio output and stereo input are supported in I2S/PCM controller.
There are two 2 channel I2S/PCM controllers embedded in the design, I2S1 and I2S2.
Common features for I2S1 and I2S2 are as follows.

- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2 channels audio receiving in PCM mode
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support single LRCK for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity

22.2 Block Diagram

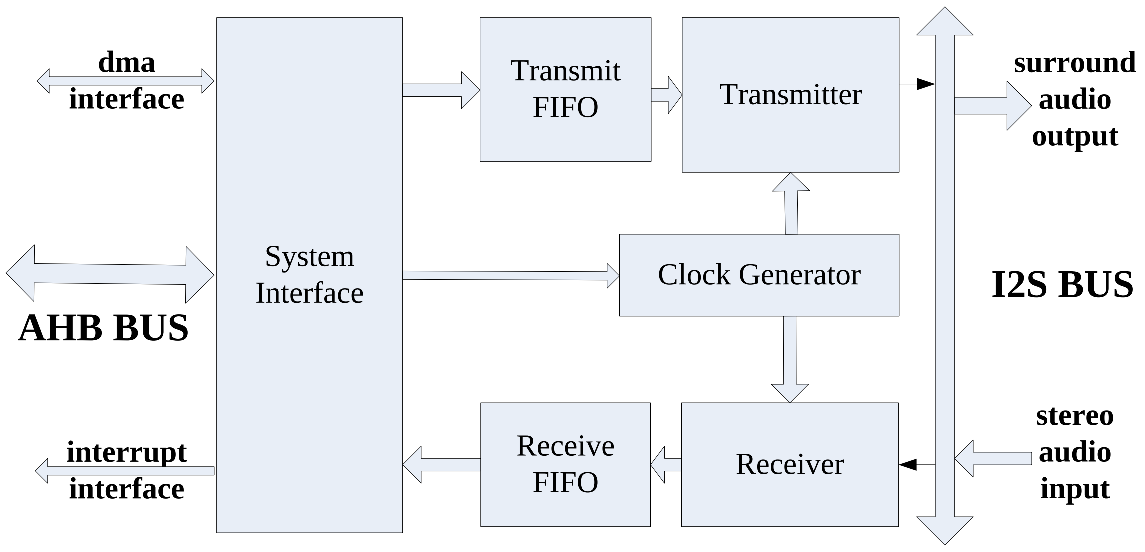


Fig.22-1 I2S/PCM controller (2 channel) Block Diagram

System Interface
The system interface implements the AHB slave operation. It contains not only control registers of transmitter and receiver inside but also interrupt and DMA handshake interface.
Clock Generator
The Clock Generator implements clock generation function. The input source clock to the

module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitter

The Transmitter implements transmission operation. The transmitter can act as either master or slave, with I2S or PCM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

22.3 Function description

In the I2S/PCM controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

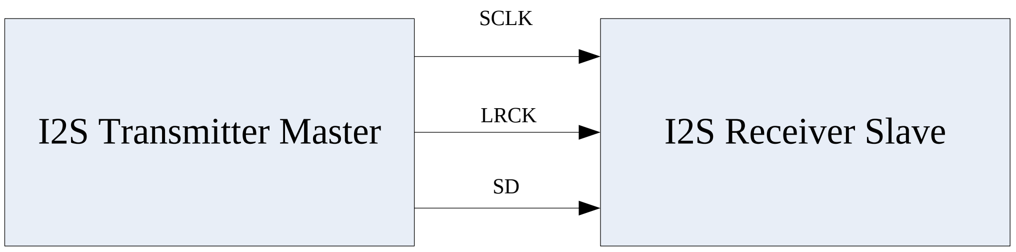


Fig.22-2 I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

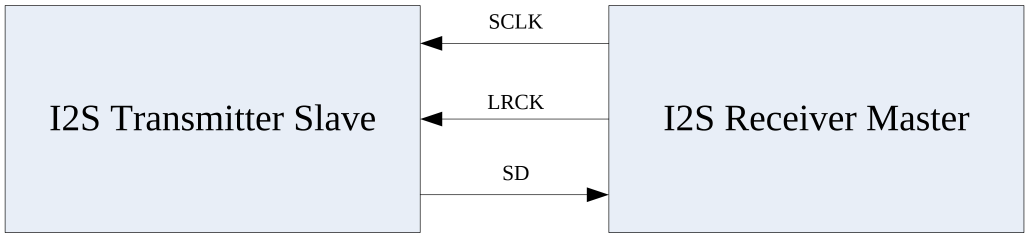


Fig.22-3 I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then the receiver start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

22.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

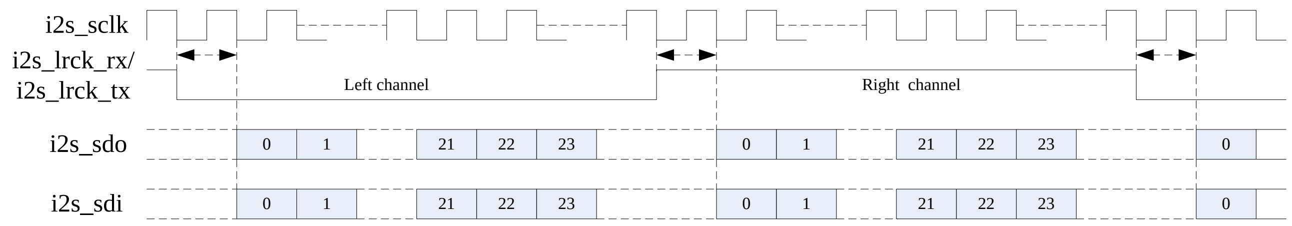


Fig.22-4 I2S normal mode timing format

22.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

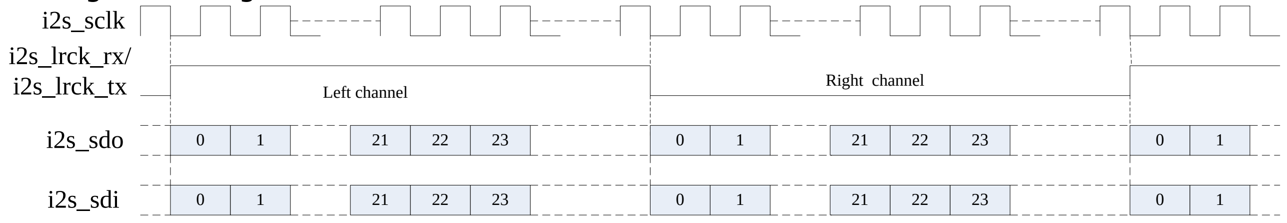


Fig.22-5 I2S left justified mode timing format

22.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

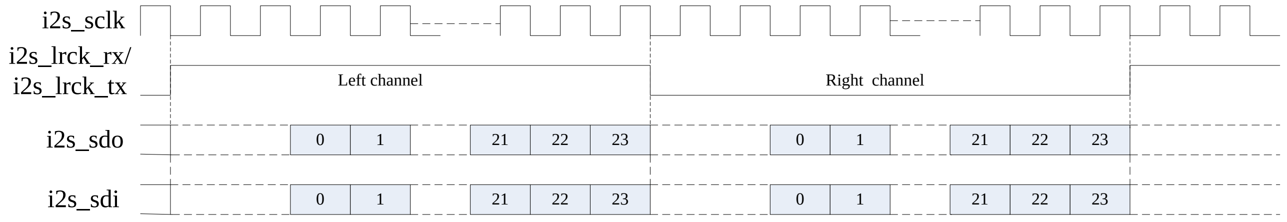


Fig.22-6 I2S right justified modetiming format

22.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

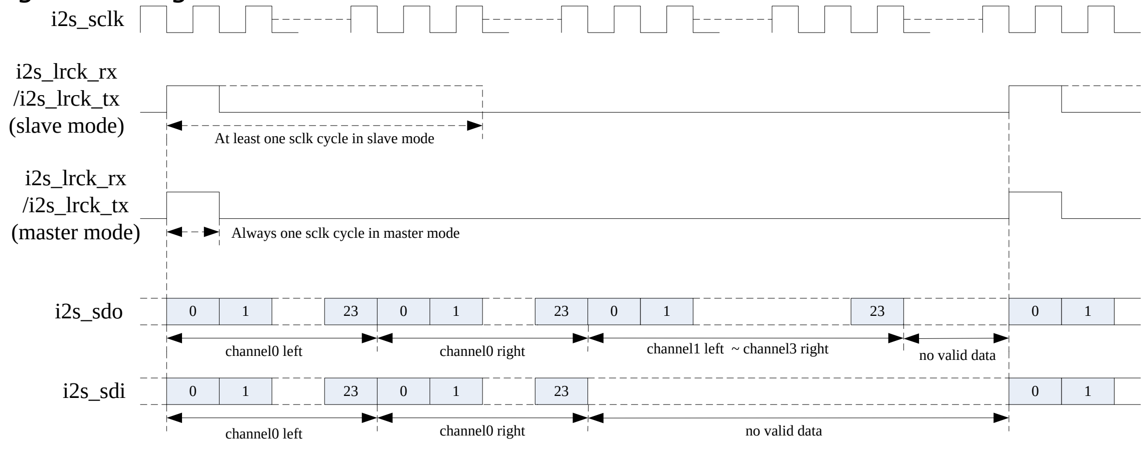


Fig.22-7 PCM early modetiming format

22.3.5 PCM late1 mode

This is the waveform of PCM late1 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high.

The range of SD signal width is from 16 to 32bits.

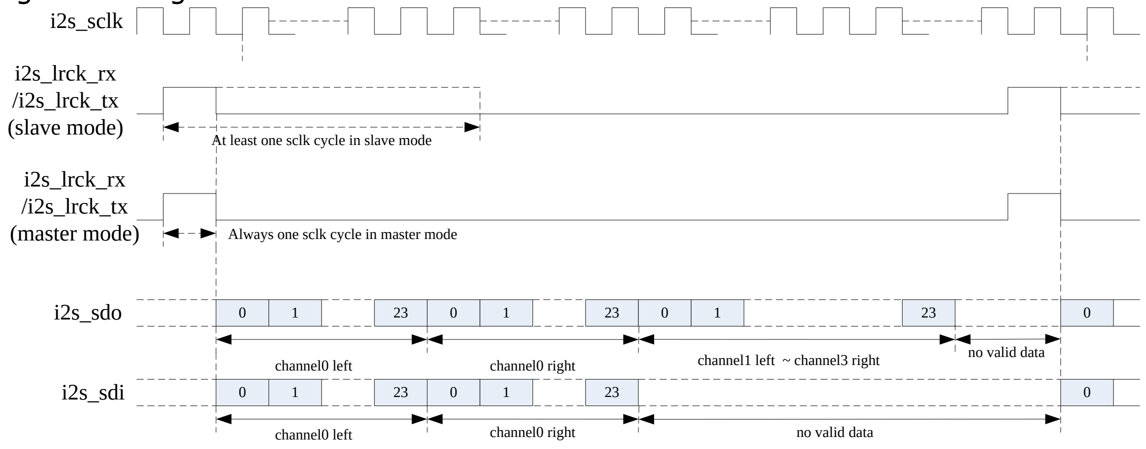

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
i2s_sdi
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

Fig.22-8 PCM late1 modetiming format

22.3.6 PCM late2 mode

This is the waveform of PCM late2 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

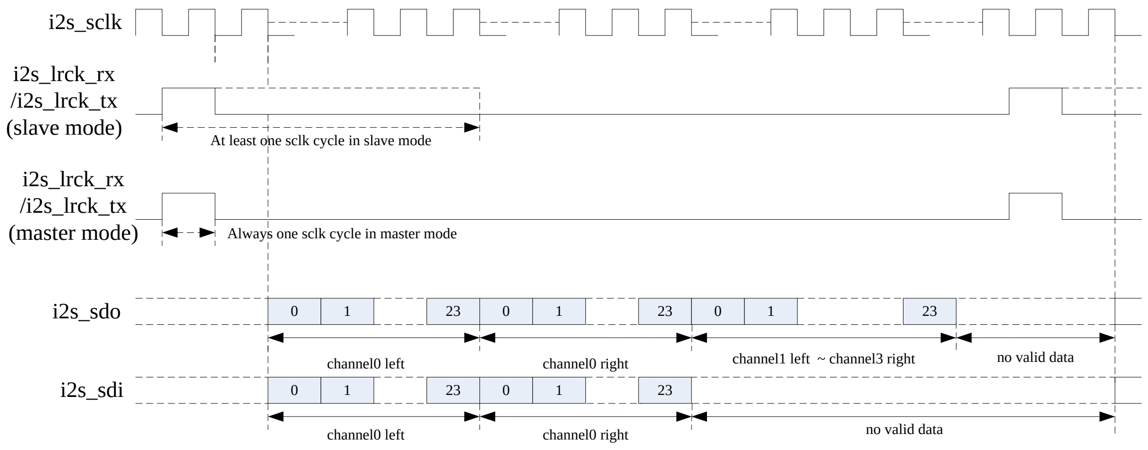

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
i2s_sdi
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

Fig.22-9 PCM late2 modetiming format

22.3.7 PCM late3 mode

This is the waveform of PCM late3 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

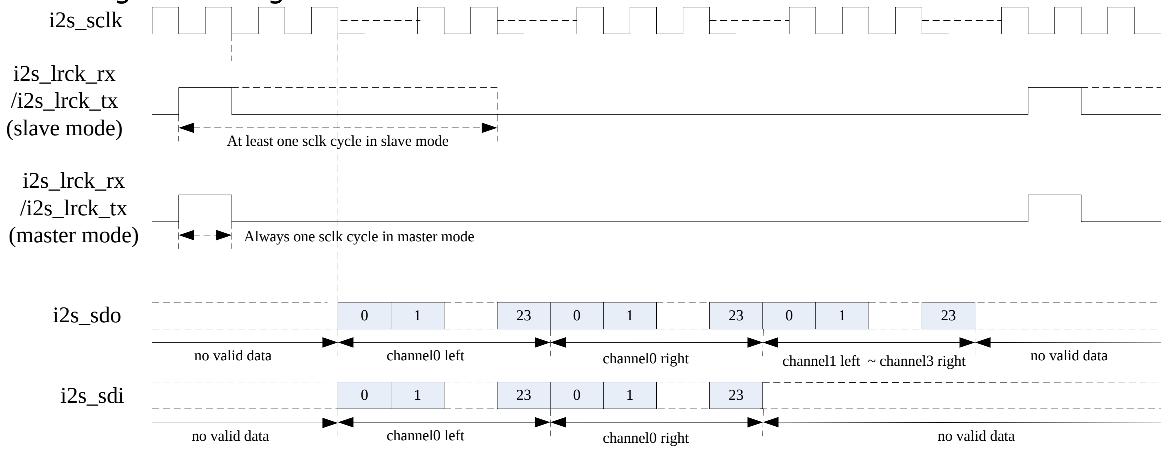

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
i2s_sdi
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right

Fig.22-10 PCM late3 modetiming format

22.4 Register Description

This section describes the control/status registers of the design.

22.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x0000000f	Transmit operation control register.
I2S_RXCR	0x0004	W	0x0000000f	Receive operation control register
I2S_CKR	0x0008	W	0x00071f00	Clock generation register
I2S_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x00000000	Interrupt control register
I2S_INTSR	0x0018	W	0x00000000	Interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	Sclk domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register
I2S_VER	0x0030	W	0x20150001	Version

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

22.4.2 Detail Register Description

I2S_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	TCSR 2'b00:two channel 2'b01~2'b11: reserved
14	RW	0x0	HWT (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1: low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	SJM SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. This bit is invalid if VDW select 16bit data and HWT select 0, Because every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0: right justified 1: left justified
11	RW	0x0	FBM (Can be written only when XFER[0] bit is 0.) 0: MSB 1: LSB
10:9	RW	0x0	IBM (Can be written only when XFER[0] bit is 0.) 0: I2S normal 1: I2S Left justified 2: I2S Right justified 3: reserved
8:7	RW	0x0	PBM (Can be written only when XFER[0] bit is 0.) 0: PCM no delay mode 1: PCM delay 1 mode 2: PCM delay 2 mode 3: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format
4:0	RW	0x0f	VDW (Can be written only when XFER[0] bit is 0.) 0~14: reserved 15: 16bit 16: 17bit 17: 18bit 18: 19bit …… n: (n+1)bit …… 28: 29bit 29: 30bit 30: 31bit 31: 32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:15	RW	0x0	RCSR 2'b00:two channel 2'b01~2'b11: reserved
14	RW	0x0	HWT (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1: low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0: right justified 1: left justified
11	RW	0x0	FBM (Can be written only when XFER[1] bit is 0.) 0: MSB 1: LSB
10:9	RW	0x0	IBM (Can be written only when XFER[1] bit is 0.) 0: I2S normal 1: I2S Left justified 2: I2S Right justified 3: reserved
8:7	RW	0x0	PBM (Can be written only when XFER[1] bit is 0.) 0: PCM no delay mode 1: PCM delay 1 mode 2: PCM delay 2 mode 3: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS (Can be written only when XFER[1] bit is 0.) 0: i2s 1: pcm

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15: 16bit 16: 17bit 17: 18bit 18: 19bit …… n: (n+1)bit …… 28: 29bit 29: 30bit 30: 31bit 31: 32bit

I2S_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM 2'b00/2'b11: tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01: only tx_lrck is used as synchronous signal for TX and RX. 2'b10: only rx_lrck is used as synchronous signal for TX and RX.
27	RW	0x0	MSS (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: master mode(sclk output) 1: slave mode(sclk input)
26	RW	0x0	CKP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	RLP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)

Bit	Attr	Reset Value	Description
24	RW	0x0	TLP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1: oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
23:16	RW	0x00	MDIV (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = (Fmclk / Ftxsclk)-1. That is (mclk frequecy / txsclk frequecy)-1.
15:8	RW	0x1f	RSD (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x00	TSD (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs

I2S_TXFIFOLR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TFL0 Contains the number of valid data entries in the transmit FIFO.

I2S_DMACR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO is equal to or below this field value.

I2S_INTCR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x00	RFT When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE 0: disable 1: enable
16	RW	0x0	RXFIE 0: disable 1: enable

Bit	Attr	Reset Value	Description
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT When the number of transmit FIFO entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE 0: disable 1: enable
0	RW	0x0	TXEIE 0: disable 1: enable

I2S_INTSR
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI 0: inactive 1: active
16	RO	0x0	RXFI 0: inactive 1: active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI 0: inactive 1: active
0	RO	0x0	TXEI 0: inactive 1: active

I2S_XFER
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS 0: stop RX transfer. 1: start RX transfer
0	RW	0x0	TXS 0: stop TX transfer. 1: start TX transfer

I2S_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC This is a self cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC This is a self cleared bit. Write 1 to clear all transmit logic.

I2S_TXDR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR When it is written to, data are moved into the transmit FIFO.

I2S_RXDR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR When the register is read, data in the receive FIFO is accessed.

I2S_RXFIFOLR
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	RFL0 Contains the number of valid data entries in the receive FIFO.

I2S_VER
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x20150001	VER Version of I2S.

22.5 Interface Description

Table 22-1 I2S Interface Description

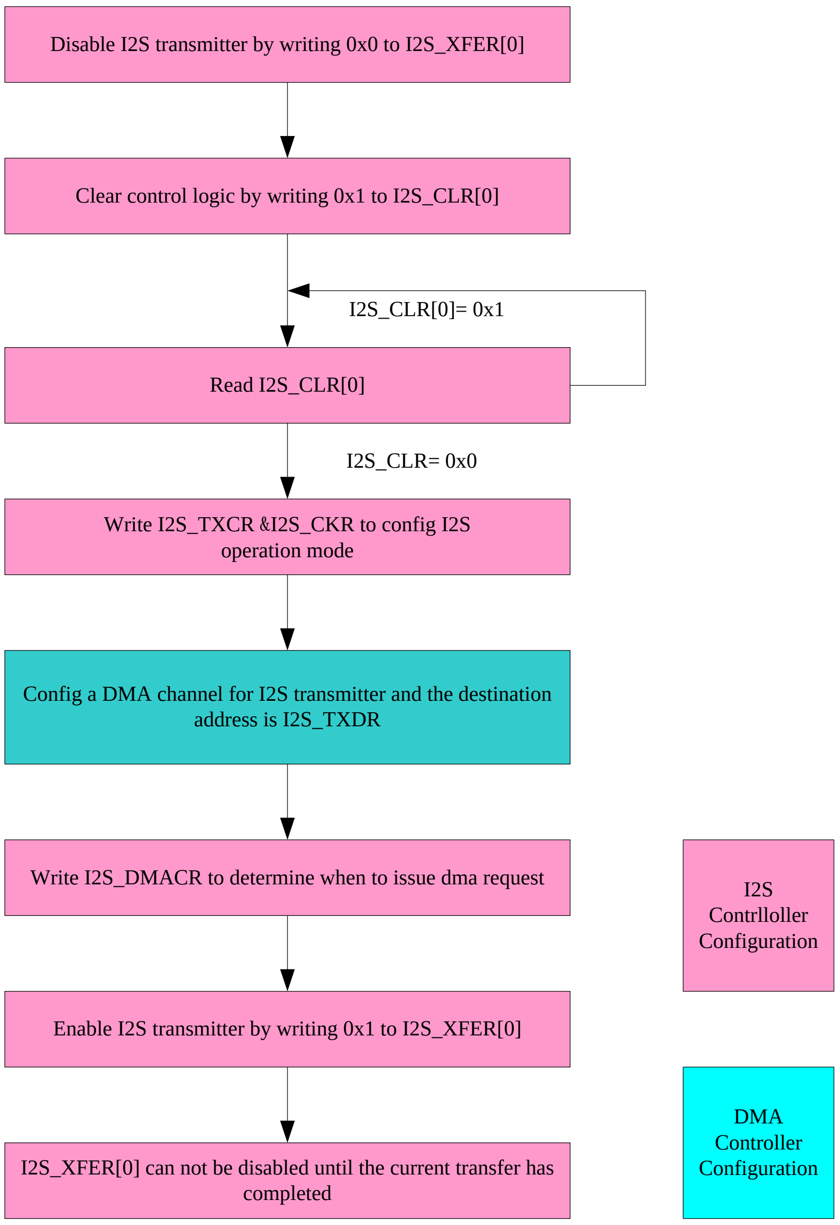
Module Pin	Direction	Pad Name	IOMUX Setting
Interface for i2s1			
i2s1_mclk	I/O	IO_I2S12ch_mclk_GPIO2C3vccio5	GRF_GPIO2C_IOMUX_L[14:12] =3’b001
i2s1_sclk	I/O	IO_I2S12ch_sclk_ GPIO2C2vccio5	GRF_GPIO2C_IOMUX_L[10:8] =3’b001
i2s1_lrck	I/O	IO_I2S12ch_lrck_ GPIO2C1vccio5	GRF_GPIO2C_IOMUX_L[6:4] =3’b001
i2s1_sdo	O	IO_I2S12ch_sdo_ GPIO2C4vccio5	GRF_GPIO2C_IOMUX_H[2:0] =3’b001
i2s1_sdi	I	IO_I2S12ch_sdi_PDMsdi0m1_GPIO2C5vccio5	GRF_GPIO2C_IOMUX_H[6:4] =3’b001
Interface for i2s2			
i2s2_mclk	I/O	IO_LCDChsyncm0_I2S22ch_mclk_CIFd0	GRF_GPIO3A_IOMUX_L[6:4]

Module Pin	Direction	Pad Name	IOMUX Setting
		m1_UART5rx_GPIO3A1vccio4	=3'b010
i2s2_sclk	I/O	IO_LCDCvsyncm0_I2S22ch_sclk_CIFd1 m1_UART5tx_GPIO3A2vccio4	GRF_GPIO3A_IOMUX_L[10:8] =3'b010
i2s2_lrck	I/O	IO_LCDCdenm0_I2S22ch_lrck_CIFd2m1 _UART5cts_GPIO3A3vccio4	GRF_GPIO3A_IOMUX_L[14:12] =3'b010
i2s2_sdi	I	IO_LCDCd1m0_I2S22ch_sdi_CIFd3m1_ UART5rts_GPIO3A5vccio4	GRF_GPIO3A_IOMUX_H[6:4] =3'b010
i2s2_sdo	O	IO_LCDCd3m0_I2S22ch_sdo_CIFd4m1 _GPIO3A7vccio4	GRF_GPIO3A_IOMUX_H[14:12] =3'b010

Notes: I=input, O=output, I/O=input/output, bidirectional

There is a requirement that the sample rate of I2S1 be the same if TX and RX work at the same time. In this situation, i2s1_lrck is driven by i2s1_lrck_tx or i2s1_lrck_rx by configuring GRF_IOFUNC_SEL0[0]. If GRF_IOFUNC_SEL0[0]=0, the i2s1_lrck_rx is connected to i2s1_lrck. Otherwise the i2s1_lrck_tx is connected to i2s1_lrck.
The same situation applies to I2S2. In this situation, i2s2_lrck is driven by i2s2_lrck_tx or i2s2_lrck_rx by configuring GRF_IOFUNC_SEL0[1]. If GRF_IOFUNC_SEL0[1]=0, the i2s2_lrck_rx is connected to i2s2_lrck. Otherwise the i2s2_lrck_tx is connected to i2s2_lrck.

22.6 Application Notes


Disable I2S transmitter by writing 0x0 to I2S_XFER[0]
Clear control logic by writing 0x1 to I2S_CLR[0]
I2S_CLR[0]= 0x1
Read I2S_CLR[0]
I2S_CLR= 0x0
Write I2S_TXCR &I2S_CKR to config I2S operation mode
Config a DMA channel for I2S transmitter and the destination address is I2S_TXDR
Write I2S_DMACR to determine when to issue dma request
Enable I2S transmitter by writing 0x1 to I2S_XFER[0]
I2S_XFER[0] can not be disabled until the current transfer has completed
I2S Contrlloller Configuration
DMA Controller Configuration
Fig.22-11 I2S/PCM controller transmit operation flow chart

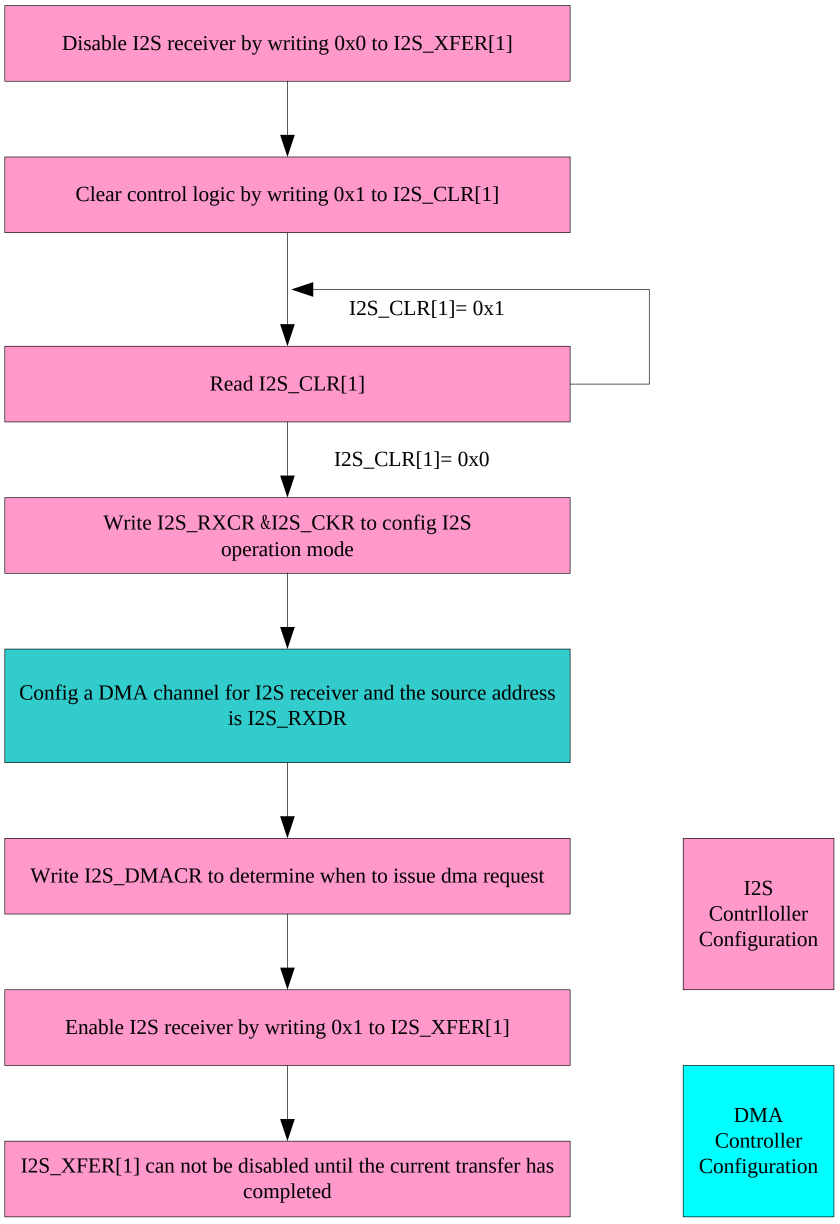

Disable I2S receiver by writing 0x0 to I2S_XFER[1]
Clear control logic by writing 0x1 to I2S_CLR[1]
I2S_CLR[1]= 0x1
Read I2S_CLR[1]
I2S_CLR[1]= 0x0
Write I2S_RXCR &I2S_CKR to config I2S operation mode
Config a DMA channel for I2S receiver and the source address is I2S_RXDR
Write I2S_DMACR to determine when to issue dma request
Enable I2S receiver by writing 0x1 to I2S_XFER[1]
I2S_XFER[1] can not be disabled until the current transfer has completed
I2S Contrllоller Configuration
DMA Controller Configuration

Fig.22-12 I2S/PCM controller receive operation flow chart

Chapter 23 I2S 8-channel

23.1 Overview

The I2S/PCM/TDM controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and is invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

I2S bus is widely used in the devices such as ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

23.1.1 Features

The I2S/PCM/TDM controller supports I2S,PCM and TDM mode stereo audio output and input.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and the other for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshaking interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combined interrupt output
- Support 8-channel audio transmitting in I2S/TDM mode and 2-channelin PCM mode.
- Support 8-channel audio receiving in I2S/TDMmode and 2 channel in PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support TDM normal,1/2 cycle left shift ,1 cycle left shift,2 cycle left shift, right shift mode serial audio data transfer.
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and the other for transmitting audio data. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity
- Support TDM programmable slot bit width: 16~32bits
- Support TDM programmable frame width: 32~512bits
- Support TDM programmable FSYNC width
- Support SDI,SDO IOMUX.

23.2 Block Diagram

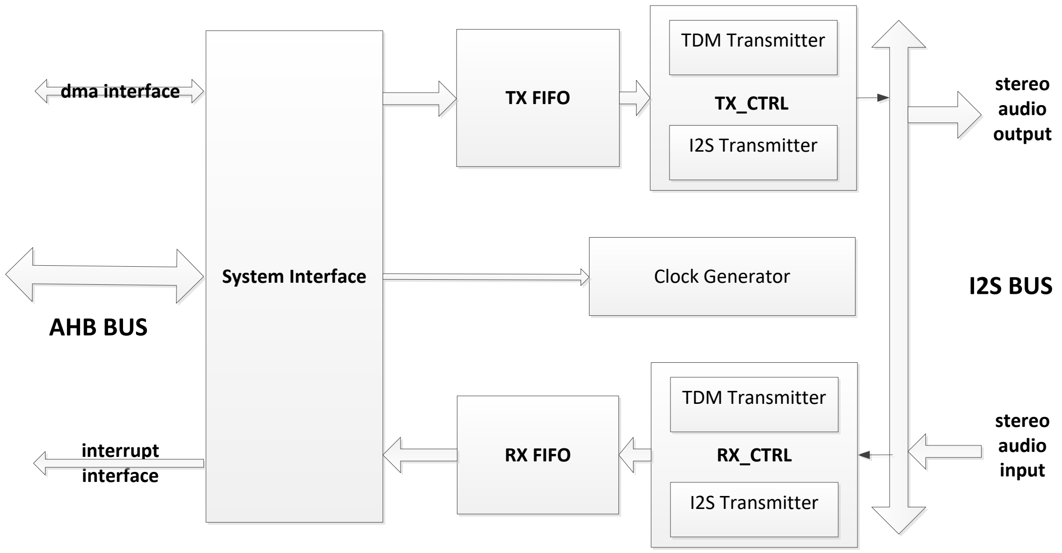


Fig.23-1I2S/PCM/TDM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S, PCM or TDM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either a master or a slave, with I2S, PCM or TDM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

23.3 Function description

In the I2S/PCM/TDM controller, there are four types: transmitter-master &receiver-master;transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

In broadcasting application, the I2S/PCM/TDM controller is used as a transmitter and external or internal audio CODEC is used as a receiver. In recording application, the I2S/PCM/TDM controller is used as a receiver and external or internal audio CODEC is used as a transmitter. Either the I2S/PCM/TDM controller or the audio CODEC can act as a master or a slave, but if one is master, the other must be slave.

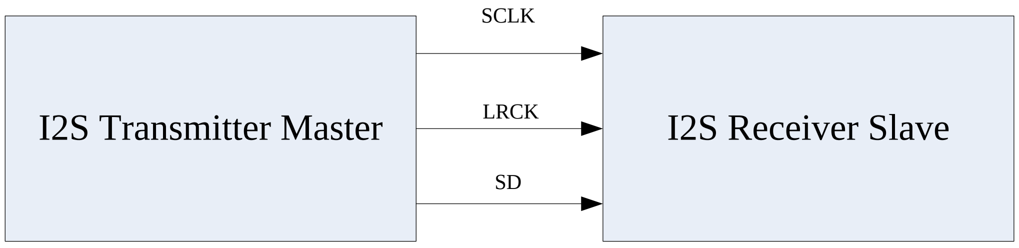


Fig.23-2I2S transmitter-master & receiver-slave condition

When the transmitter acts as a master, it sends all signals to thereceiver (the slave), and CPU controls when to send clock and data to the receiver. When acts as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from the receiver (the master) to the transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when the transmitterto send data.

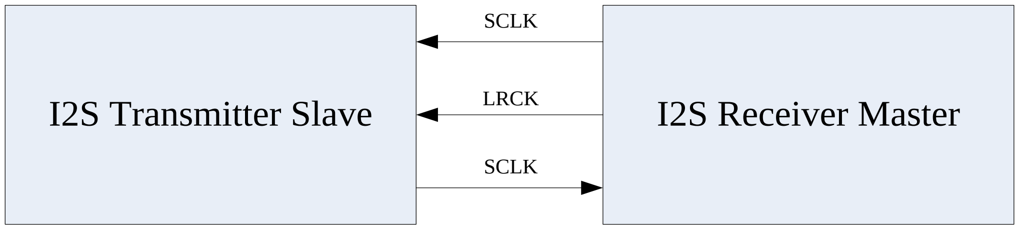


Fig.23-3I2S transmitter-slave & receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (the slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

23.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

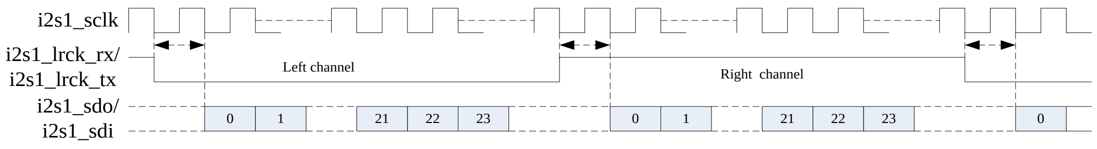


Fig.23-4I2S normal mode timing format

23.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s1_lrck_rx / i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

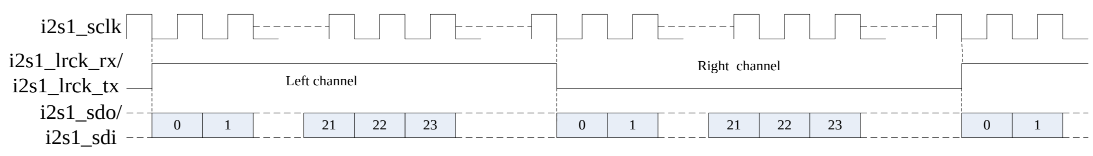


Fig.23-5I2S left justified mode timing format

23.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s1_lrck_rx/ i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode, the last bit of the transferred data is aligned to the transition edge of the LRCK signal while one bit is transferred at one SCLK cycle. The range of SD signal width is from 16 to 32bits.

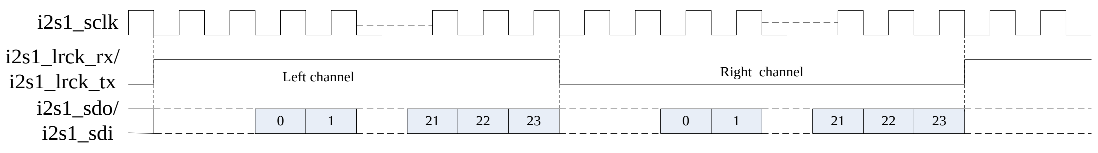


Fig.23-6I2S right justified mode timing format

23.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

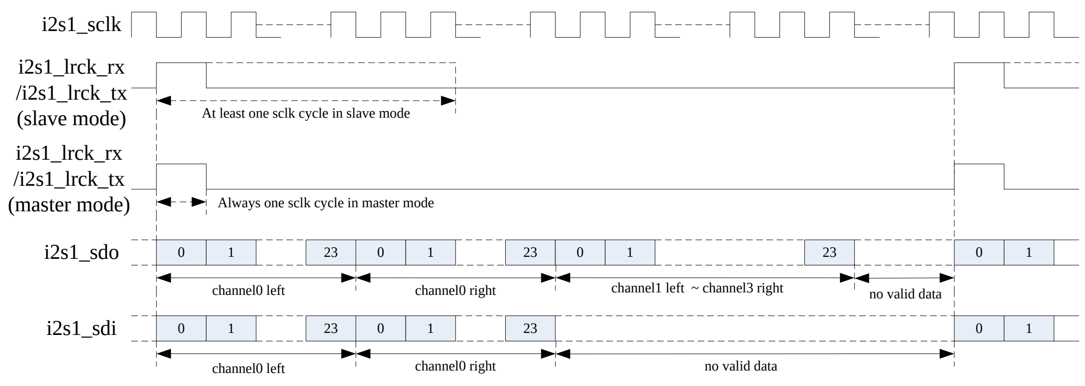


Fig.23-7PCM early mode timing format

23.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

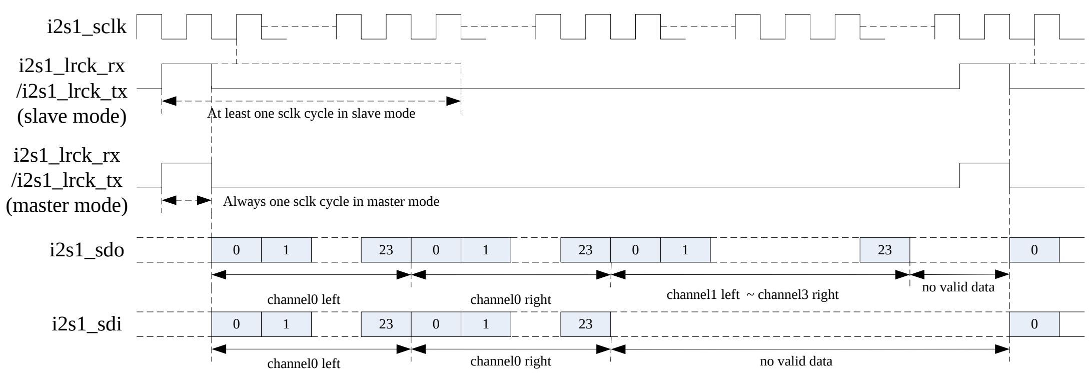


Fig.23-8PCM late1 mode timing format

23.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB)two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

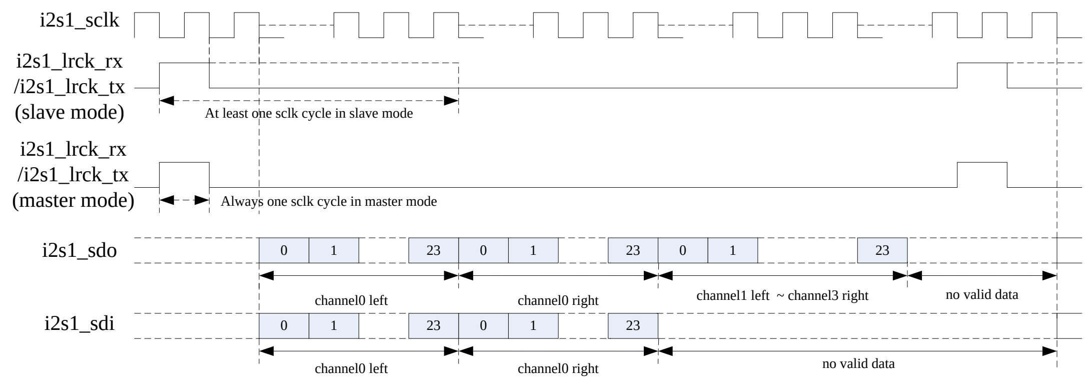


Fig.23-9PCM late2 mode timing format

23.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

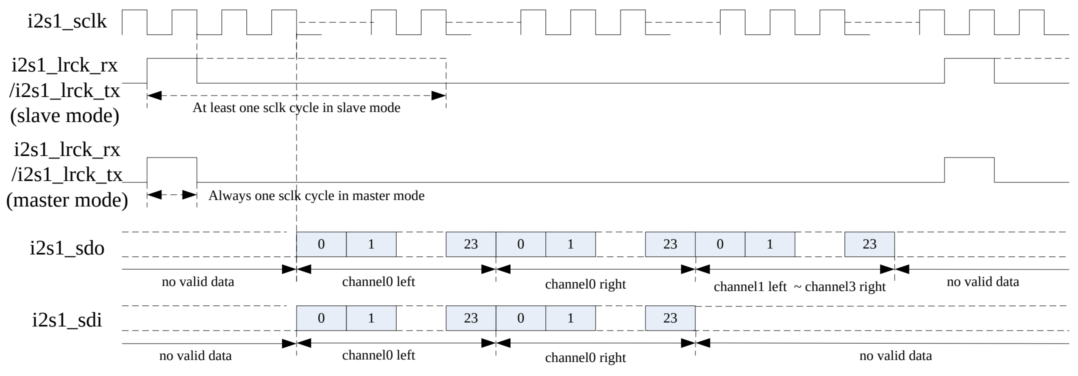
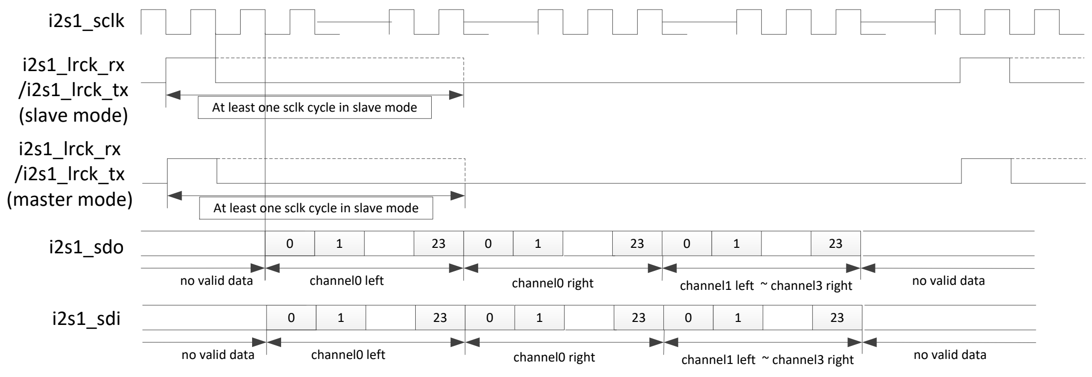


Fig.23-10PCM late3 mode timing format

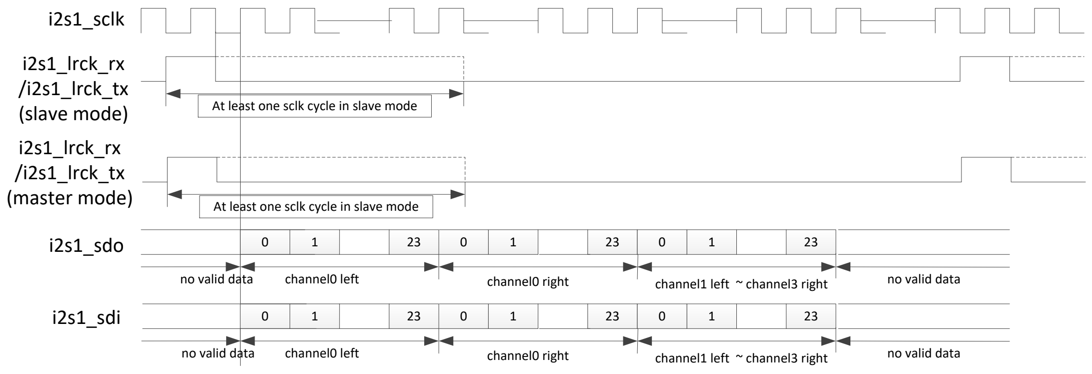
23.3.8 TDM normal mode (PCM format)

This is the waveform of TDM normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the second falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.


i2s1_sclk
i2s1_lrck_rx /i2s1_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s1_lrck_rx /i2s1_lrck_tx (master mode)
At least one sclk cycle in slave mode
i2s1_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s1_sdi
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

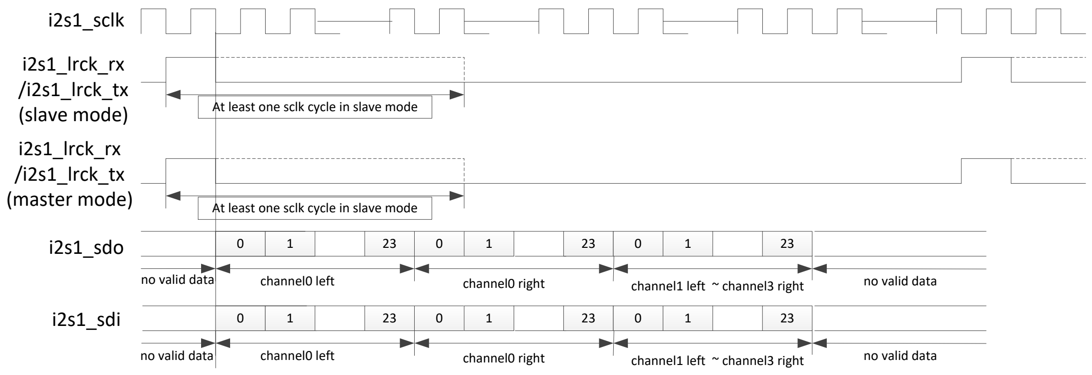
23.3.9 TDM left shift mode0 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the second rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.


i2s1_sclk
i2s1_lrck_rx /i2s1_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s1_lrck_rx /i2s1_lrck_tx (master mode)
At least one sclk cycle in slave mode
i2s1_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s1_sdi
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

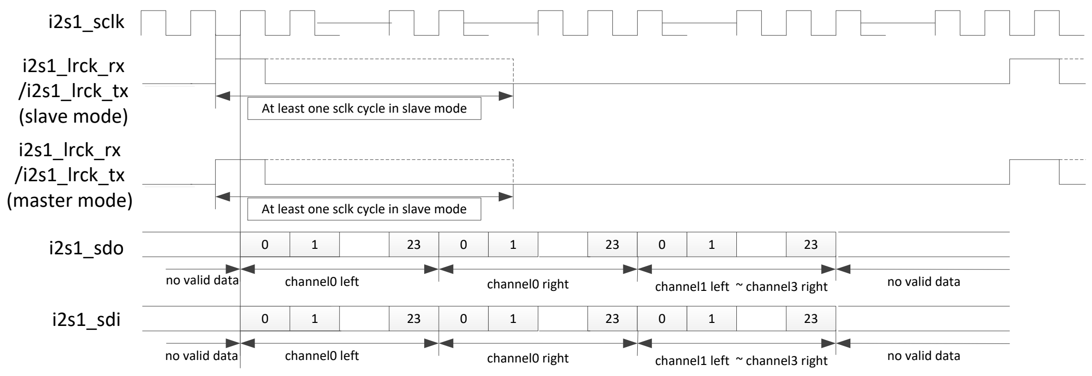
23.3.10 TDM left shift mode1 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the first falling edge of SCLKafter LRCK goes high. The range of SD signal width is from 16 to 32bits.


i2s1_sclk
i2s1_lrck_rx /i2s1_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s1_lrck_rx /i2s1_lrck_tx (master mode)
At least one sclk cycle in slave mode
i2s1_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s1_sdi
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

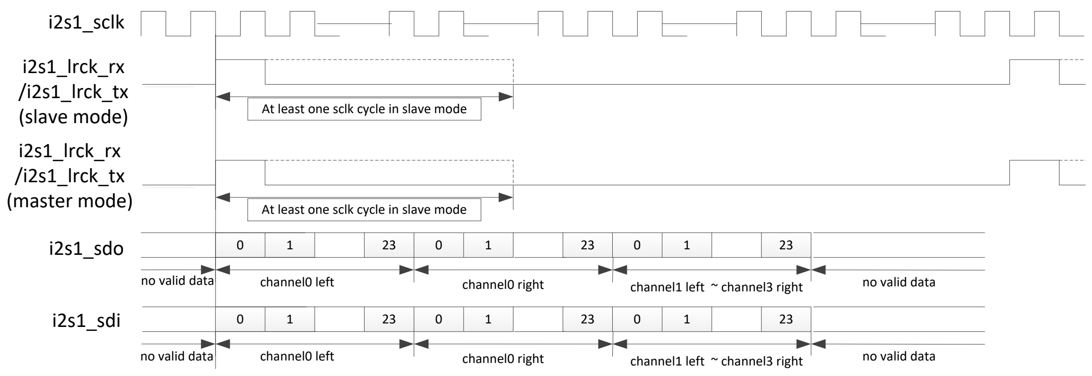
23.3.11 TDM left shift mode2 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the first rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.


i2s1_sclk
i2s1_lrck_rx /i2s1_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s1_lrck_rx /i2s1_lrck_tx (master mode)
At least one sclk cycle in slave mode
i2s1_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s1_sdi
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

23.3.12 TDM left shift mode3 (PCM format)

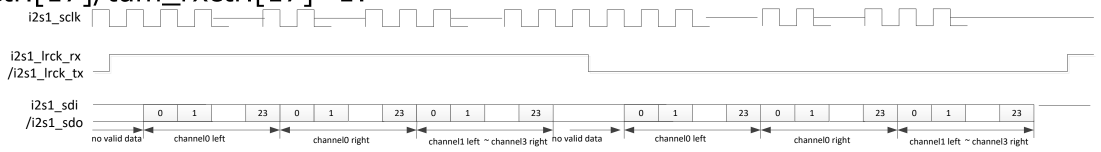
This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.


i2s1_sclk
i2s1_lrck_rx /i2s1_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s1_lrck_rx /i2s1_lrck_tx (master mode)
At least one sclk cycle in slave mode
i2s1_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s1_sdi
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data

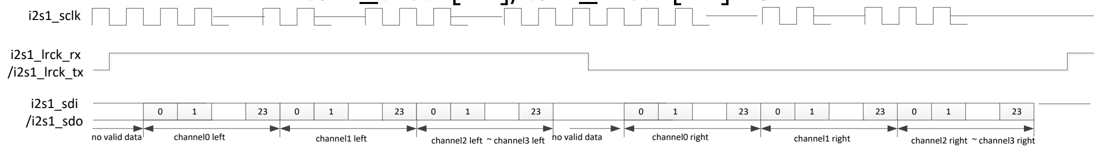
23.3.13 TDM normal mode (I2S format)

This is the waveform of I2S normal mode. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB)on the first falling edge of SCLK after LRCK changes. The range of SD signal width is from 16 to 32bits.

tdm_txctrl[17]/tdm_rxctrl[17]=1:

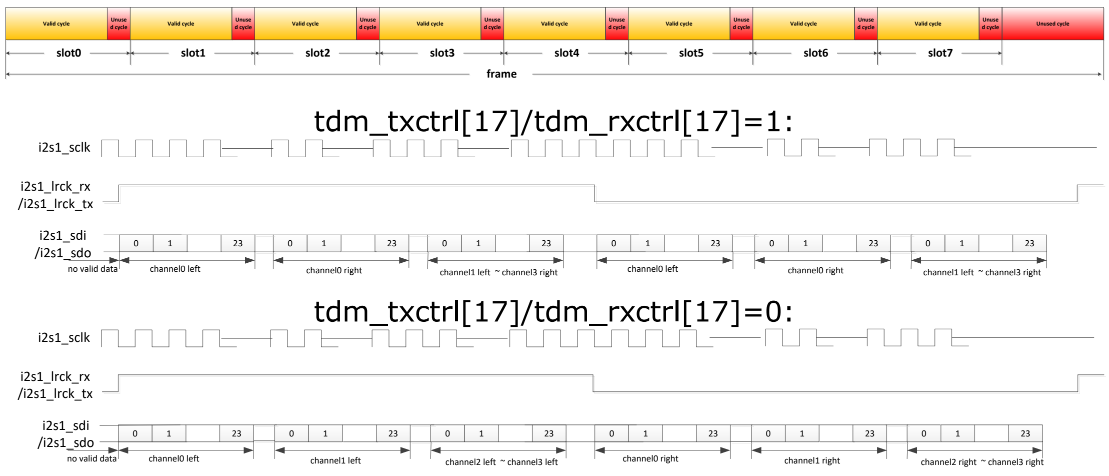


tdm_txctrl[17]/tdm_rxctrl[17]=0:



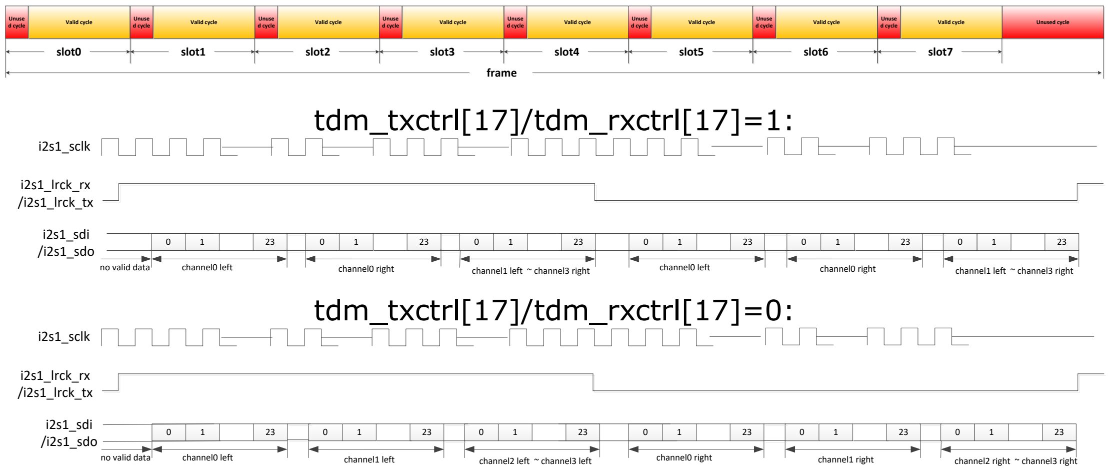
23.3.14 TDM left justified mode (I2S format)

This is the waveform of I2S left justified mode. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.



23.3.15 TDM right justified mode (I2S format)

This is the waveform of I2S right justified mode.For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode.The range of SD signal width is from 16 to 32bits.



23.4 Register description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_8CH_TXCR	0x0000	W	0x7200000f	transmit operation control register.
I2S_8CH_RXCR	0x0004	W	0x01c8000f	receive operation control register
I2S_8CH_CKR	0x0008	W	0x00001f1f	clock generation register
I2S_8CH_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_8CH_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_8CH_INTCR	0x0014	W	0x01f00000	interrupt control register
I2S_8CH_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_8CH_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_8CH_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_8CH_TXDR	0x0024	W	0x00000000	Transimt FIFO Data Register

Name	Offset	Size	Reset Value	Description
I2S_8CH_RXDR	0x0028	W	0x00000000	When the register is read, data in the receive FIFO is accessed.
I2S_8CH_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register
I2S_8CH_TDM_TXCTRL	0x0030	W	0x00003eff	TDM mode transmit operation control register
I2S_8CH_TDM_RXCTRL	0x0034	W	0x00003eff	TDM mode receive operation control register
I2S_8CH_CLKDIV	0x0038	W	0x00000707	clock divider register
I2S_8CH_VERSION	0x003c	W	0x20150001	I2S version register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

23.4.2 Detail Register Description

I2S_8CH_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:29	RW	0x3	tx_path_select3 Tx path select; 2'b00: sdo3 output data from path0; 2'b01: sdo3 output data from path1; 2'b10: sdo3 output data from path2; 2'b11: sdo3 output data from path3; Note: when TDM mode, only path0 enable.
28:27	RW	0x2	tx_path_select2 Tx path select; 2'b00: sdo2 output data from path0; 2'b01: sdo2 output data from path1; 2'b10: sdo2 output data from path2; 2'b11: sdo2 output data from path3; Note: when TDM mode, only path0 enable.
26:25	RW	0x1	tx_path_select1 Tx path select; 2'b00: sdo1 output data from path0; 2'b01: sdo1 output data from path1; 2'b10: sdo1 output data from path2; 2'b11: sdo1 output data from path3; Note: when TDM mode, only path0 enable.
24:23	RW	0x0	tx_path_select0 Tx path select; 2'b00: sdo0 output data from path0; 2'b01: sdo0 output data from path1; 2'b10: sdo0 output data from path2; 2'b11: sdo0 output data from path3; Note: when TDM mode, only path0 enable.

Bit	Attr	Reset Value	Description
22:17	RW	0x00	RCNT (Can be written only when XFER[0] bit is 0.) Only vailid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid. Note: Only function when TX TFS[1]=0;
16:15	RW	0x0	TCSR 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved Note: Only function when TX TFS[1:0] is 0;
8:7	RW	0x0	PBM (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode Note: function when TX TFS[1:0] is 1;

Bit	Attr	Reset Value	Description
6:5	RW	0x0	TFS (Can be written only when XFER[0] bit is 0.) 2'b00: I2S format 2'b01: PCM format 2'b10: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)
4:0	RW	0x0f	VDW (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_8CH_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

24:23	RW	0x3	rx_path_select3
			2'b00: path3 data from sdi0;
			2'b01: path3 data from sdi1;
			2'b10: path3 data from sdi2;
			2'b11: path3 data from sdi3;
			Note: inoperative at TDM mode.

22:21	RW	0x2	rx_path_select2
			Rx path select;
			2'b00: path2 data from sdi0;
			2'b01: path2 data from sdi1;
			2'b10: path2 data from sdi2;
			2'b11: path2 data from sdi3;
			Note: inoperative at TDM mode.

20:19	RW	0x1	rx_path_select1
			Rx path select;
			2'b00: path1 data from sdi0;
			2'b01: path1 data from sdi1;
			2'b10: path1 data from sdi2;
			2'b11: path1 data from sdi3;
			Note: inoperative at TDM mode.

Bit	Attr	Reset Value	Description

18:17	RW	0x0	rx_path_select0
			Rx path select;
			2'b00: path0 data from sdi0;
			2'b01: path0 data from sdi1;
			2'b10: path0 data from sdi2;o
			2'b11: path0 data from sdi3;

16:15	RW	0x0	RCSR 2'b00:two channel 2'b01:four channel 2'b10:six channel 2'b11:eight channel
14	RW	0x0	HWT (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved Note: Only function when RX TFS[1:0] is 0;
8:7	RW	0x0	PBM (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode Note: Only function when RX TFS[1:0] is 1;

Bit	Attr	Reset Value	Description
6:5	RW	0x0	TFS (Can be written only when XFER[1] bit is 0.) 2'b00: I2S format 2'b01: PCM format 2'b10: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)
4:0	RW	0x0f	VDW (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_8CH_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	LRCK_COMMON Lrck as common
27	RW	0x0	MSS (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	RLP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)

Bit	Attr	Reset Value	Description
24	RW	0x0	TLP (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)
23:16	RO	0x0	reserved
15:8	RW	0x1f	RSD (Can be written only when XFER[1] or XFER[0] bit is 0.) 0~30:reserved 31~255:frequency of rx_lrck= (Receive sclk divider[7:1]+1)*2*frequency of sclk Note: function when RX TFS[1:0] is 0 or 1;
7:0	RW	0x1f	TSD (Can be written only when XFER[1] or XFER[0] bit is 0.) 0~30:reserved 31~255:frequency of tx_lrck= (Transmit sclk divider[7:1]+1)*2*frequency of sclk Note: function when TX TFS[1:0] is 0 or 1;

I2S_8CH_TXFIFOLR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RW	0x00	TFL3 Contains the number of valid data entries in the transmit FIFO3.
17:12	RW	0x00	TFL2 Contains the number of valid data entries in the transmit FIFO2.
11:6	RW	0x00	TFL1 Field0000 Description
5:0	RO	0x00	TFL0 Contains the number of valid data entries in the transmit FIFO0.

I2S_8CH_DMACR
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE 0 : Receive DMA disabled 1 : Receive DMA enabled

Bit	Attr	Reset Value	Description
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2S_8CH_INTCR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x1f	RFT When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE 0:disable 1:enable
16	RW	0x0	RXFIE 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT When the number of transmit FIFO (TXFIFO0 if CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC Write 1 to clear TX underrun interrupt.

Bit	Attr	Reset Value	Description
1	RW	0x0	TXUIE 0:disable 1:enable
0	RW	0x0	TXEIE 0:disable 1:enable

I2S_8CH_INTSR
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI 0:inactive 1:active
16	RO	0x0	RXFI 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI 0:inactive 1:active
0	RO	0x0	TXEI 0:inactive 1:active

I2S_8CH_XFER
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS 0:stop TX transfer. 1:start TX transfer

I2S_8CH_CLR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC This is a self cleard bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC This is a self cleard bit. Write 1 to clear all transmit logic.

I2S_8CH_TXDR
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR When it is written to, data are moved into the transmit FIFO.

I2S_8CH_RXDR
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR When the register is read, data in the receive FIFO is accessed.

I2S_8CH_RXFIFOLR
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RW	0x00	RFL3 Contains the number of valid data entries in the Receive FIFO3.
17:12	RW	0x00	RFL2 Contains the number of valid data entries in the Receive FIFO2.
11:6	RW	0x00	RFL1 Contains the number of valid data entries in the Receive FIFO1.
5:0	RW	0x00	RFL0 Contains the number of valid data entries in the Receive FIFO0.

I2S_8CH_TDM_TXCTRL
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:18	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL1 (Can be written only when XFER[0] is 0.) 0: single period of the ASP_CLK. 1: 2 period of the ASP_CLK. n: n+1 period of the ASP_CLK. 6: 7 period of the ASP_CLK. 7: the width is equivalent to a channel block Note: function when TX TFS[1:0] is 2 or 3;
17	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL0 (Can be written only when XFER[0] is 0.) 0: 1/2 frame width. Aspc_ctrl1[8:0] should be set to an even number 1: frame width

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TDM_TX_SHIFT_CTRL (Can be written only when XFER[0] is 0.) 3'b000: PCM format: normal mode, sample data on the third rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: normal mode 3'b001: PCM format: 1/2 cycle shift left, sample data on second falling rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: left justified mode 3'b010: PCM format: 1 cycle shift left, sample data on second rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: right justified mode 3'b011: PCM format: 3/2 cycle shift left, sample data on first falling edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b100: PCM format: 2 cycle shift left, sample data on first rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b101~3'b111 not support Note: function when TX TFS[1:0] is 2 or 3;
13:9	RW	0x1f	TDM_TX_SLOT_BIT_WIDTH (Can be written only when XFER[0] is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit ...... n:(n+1)bit ...... 31:32bit Note: function when TX TFS[1:0] is 2 or 3;

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_TX_FRAME_WIDTH (Can be written only when XFER[0] is 0.) 0~30:reserved 31:32bit 32:33bit 33:34bit 34:35bit …… n:(n+1)bit …… 511:512bit Note: functional when TX TFS[1:0] is 2 or 3;

I2S_8CH_TDM_RXCTRL
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:18	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL1 (Can be written only when XFER[0] is 0.) 0: single period of the ASP_CLK. 1: 2 period of the ASP_CLK. n: n+1 period of the ASP_CLK. 6: 7 period of the ASP_CLK. 7: the width is equivalent to a channel block Note: function when RX TFS[1:0] is 2 or 3;
17	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL0 (Can be written only when XFER[0] is 0.) 0: 1/2 frame width. Aspc_ctrl1[8:0] should be set to an even number 1: frame width

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TDM_RX_SHIFT_CTRL (Can be written only when XFER[0] is 0.) 3'b000: PCM format: normal mode, sample data on the third rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: normal mode 3'b001: PCM format: 1/2 cycle shift left, sample data on second falling rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: left justified mode 3'b010: PCM format: 1 cycle shift left, sample data on second rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: right justified mode 3'b011: PCM format: 3/2 cycle shift left, sample data on first falling edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b100: PCM format: 2 cycle shift left, sample data on first rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b101~3'b111 not support Note: function when RX TFS[1:0] is 2 or 3;
13:9	RW	0x1f	TDM_RX_SLOT_BIT_WIDTH (Can be written only when XFER[0] is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 31:32bit Note: function when RX TFS[1:0] is 2 or 3;

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_RX_FRAME_WIDTH (Can be written only when XFER[0] is 0.) 0~30:reserved 31:32bit 32:33bit 33:34bit 34:35bit …… n:(n+1)bit …… 511:512bit Note: functional when RX TFS[1:0] is 2 or 3;

I2S_8CH_CLKDIV
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

15:8	RW	0x07	RX_MDIV
			(Can be written only XFER[0] bit is 0.)
			Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequecy / txsclk frequecy-1)
			0 :Fmclk=Ftxsclk;
			1 :Fmclk=2*Ftxsclk;
			2,3 :Fmclk=4*Ftxsclk;
			4,5 :Fmclk=6*Ftxsclk;
			……
			2n,2n+1:Fmclk=(2n+2)*Ftxsclk;
			……
			60,61:Fmclk=62*Ftxsclk;
			62,63:Fmclk=64*Ftxsclk;
			……
			252,253:Fmclk=254*Ftxsclk;
			254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description

7:0	RW	0x07	TX_MDIV
			(Can be written only when XFER[1] bit is 0.)
			Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequecy / txsclk frequecy-1)
			0 :Fmclk=Ftxsclk;
			1 :Fmclk=2*Ftxsclk;
			2,3 :Fmclk=4*Ftxsclk;
			4,5 :Fmclk=6*Ftxsclk;
			……
			2n,2n+1:Fmclk=(2n+2)*Ftxsclk;
			……
			60,61:Fmclk=62*Ftxsclk;
			62,63:Fmclk=64*Ftxsclk;
			……
			252,253:Fmclk=254*Ftxsclk;
			254,255:Fmclk=256*Ftxsclk;

I2S_8CH_VERSION

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x20150001	I2S_VERSION i2s_version

23.5 Interface Description

Table 23-1 I2S Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_8ch_mclk	I/O	IO_LCDCd13_I2S08ch_mclk_GPIO3C1vccio4	GRF_GPIO3C_IOMUX_L[6:4]=3’b010
i2s0_8ch_sclk_rx	I/O	IO_LCDCd8m0_I2S08ch_sclkrx_CIFd7m1_SPI1mosi_GPIO3B4vccio4	GRF_GPIO3B_IOMUX_H[2:0]=3’b010
i2s0_8ch_sclk_tx	I/O	IO_LCDCd15_I2S08ch_sclktx_PWM5_GPIO3C3vccio4	GRF_GPIO3C_IOMUX_H[14:12]=3’b010
i2s0_8ch_lrck_rx	I/O	IO_LCDCd9m0_I2S08ch_lrckrx_GPIO3B5vccio4	GRF_GPIO3B_IOMUX_H[6:4]=3’b010
i2s0_8ch_lrck_tx	I/O	IO_LCDCd14_I2S08ch_lrcktx_PWM4_GPIO3C2vccio4	GRF_GPIO3C_IOMUX_L[10:8]=3’b010
i2s_8ch_sdo0	O	IO_LCDCd16_I2S08ch_sdo0_PWM6_GPIO3C4vccio4	GRF_GPIO3C_IOMUX_H[2:0]=3’b010
i2s_8ch_sdo1	O	IO_LCDCd12_I2S08ch_sdo1_GPIO3C0vccio4	GRF_GPIO3C_IOMUX_L[2:0]=3’b010
i2s_8ch_sdo2	O	IO_LCDCd11m0_I2S08ch_sdo2_CIFd9m1_SPI1clk_GPIO3B7vccio4	GRF_GPIO3B_IOMUX_H[14:12]=3’b010
i2s_8ch_sdo3	O	IO_LCDCd10m0_I2S08ch_sdo3_CIFd8m1_SPI1miso_GPIO3B6vccio4	GRF_GPIO3B_IOMUX_H[10:8]=3’b010
i2s2_8ch_sdi0	I	IO_LCDCd17_I2S08ch_sdi0_PWM7_GPIO3C5vccio4	GRF_GPIO3C_IOMUX_H[6:4]=3’b010
i2s2_8ch_sdi1	I	IO_LCDCd7_I2S08ch_sdi1_GPIO3B3vccio4	GRF_GPIO3B_IOMUX_L[14:12]=3’b010
i2s2_8ch_sdi2	I	IO_LCDCd5m0_I2S08ch_sdi2_CIFd6m1_SPI1csn_GPIO3B1vccio4	GRF_GPIO3B_IOMUX_L[6:4]=3’b010

Module Pin	Directi on	Pad Name	IOMUX Setting
i2s2_8ch_sdi3	I	IO_LCDCd4m0_I2S08ch_sdi3_CIFd5 m1_GPIO3B0vccio4	GRF_GPIO3B_IOMUX_L[2:0] =3'b010

23.6 Application Notes

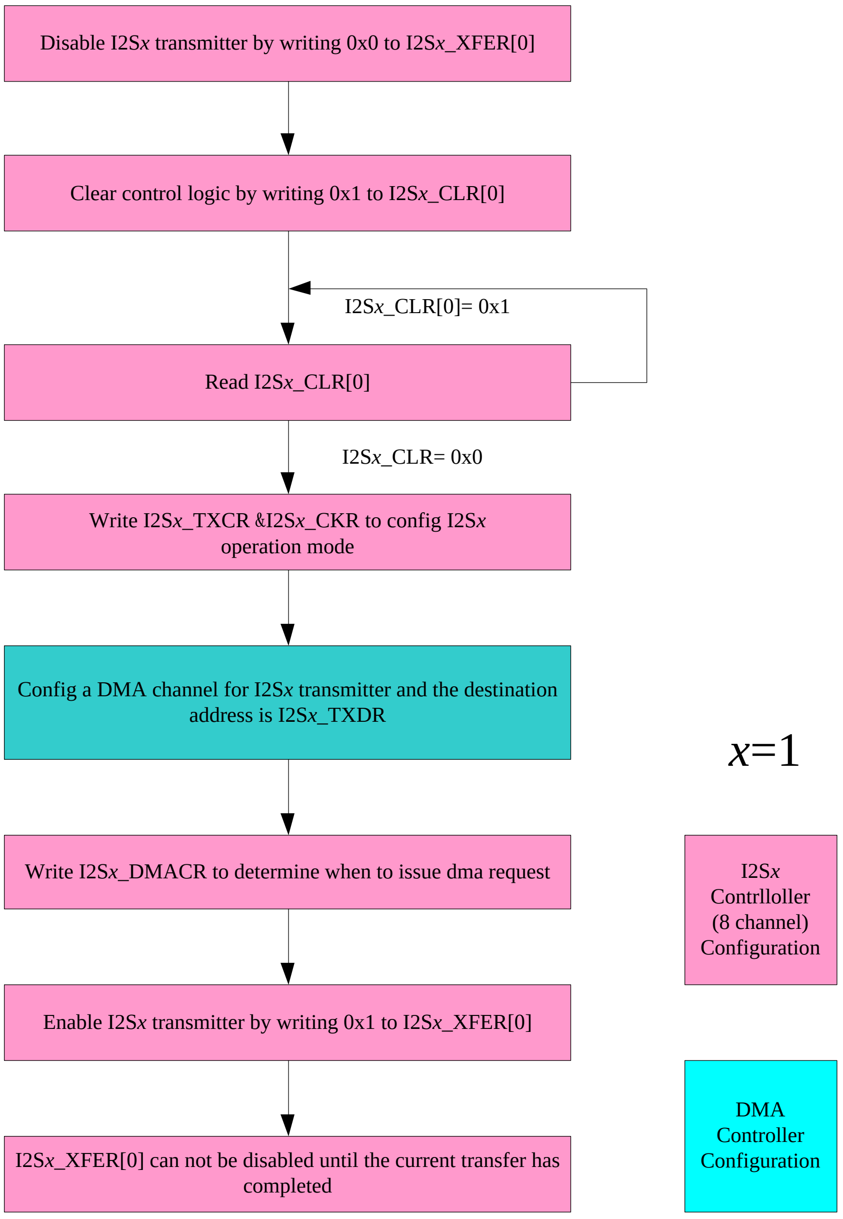

Disable I2Sx transmitter by writing 0x0 to I2Sx_XFER[0]
Clear control logic by writing 0x1 to I2Sx_CLR[0]
I2Sx_CLR[0]= 0x1
Read I2Sx_CLR[0]
I2Sx_CLR= 0x0
Write I2Sx_TXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx transmitter and the destination address is I2Sx_TXDR
x=1
Write I2Sx_DMACR to determine when to issue dma request
I2Sx Contrlloller (8 channel) Configuration
Enable I2Sx transmitter by writing 0x1 to I2Sx_XFER[0]
DMA Controller Configuration
I2Sx_XFER[0] can not be disabled until the current transfer has completed

Fig.23-11I2S/PCM/TDM controller transmit operation flow chart

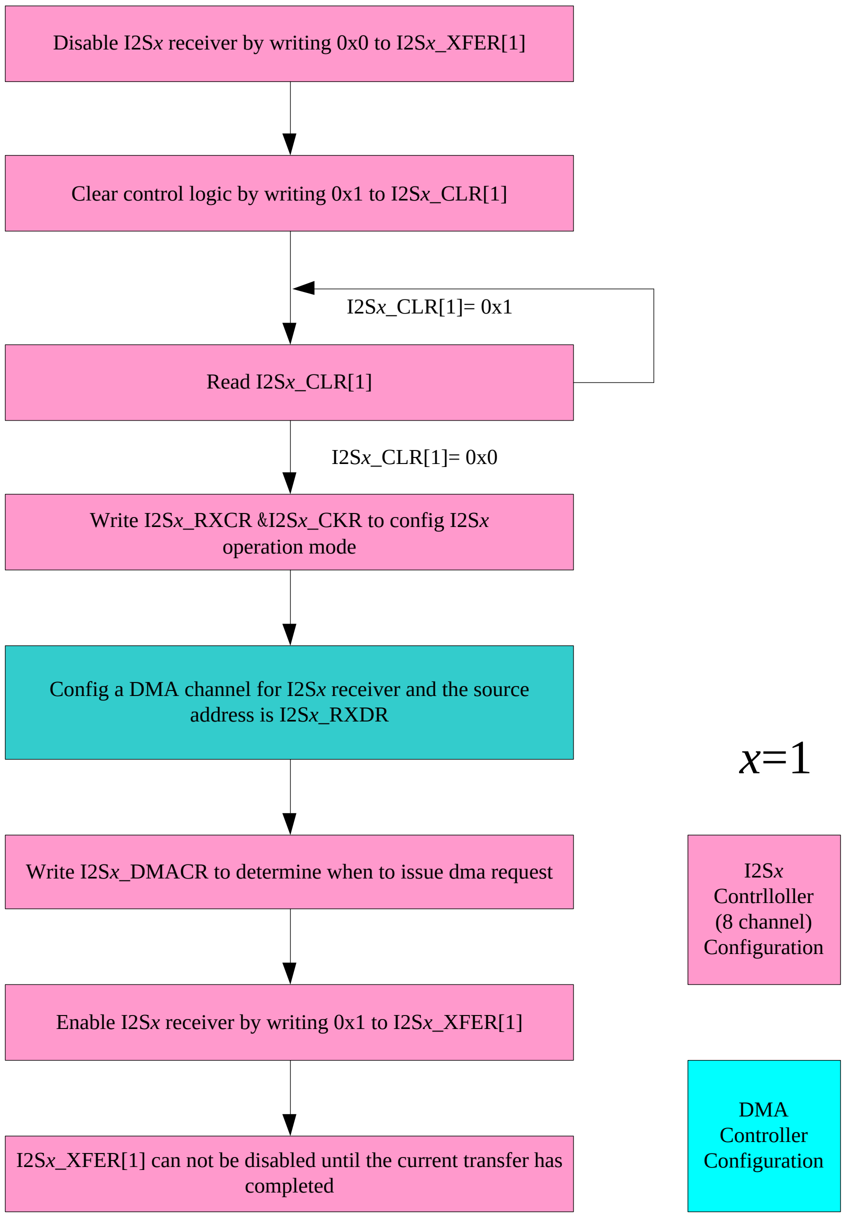

Disable I2Sx receiver by writing 0x0 to I2Sx_XFER[1]
Clear control logic by writing 0x1 to I2Sx_CLR[1]
I2Sx_CLR[1]= 0x1
Read I2Sx_CLR[1]
I2Sx_CLR[1]= 0x0
Write I2Sx_RXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx receiver and the source address is I2Sx_RXDR
x=1
Write I2Sx_DMACR to determine when to issue dma request
I2Sx Contrlloller (8 channel) Configuration
Enable I2Sx receiver by writing 0x1 to I2Sx_XFER[1]
DMA Controller Configuration
I2Sx_XFER[1] can not be disabled until the current transfer has completed

Fig.23-12I2S/PCM/TDM controller receive operation flow chart

Note: User should clear TX/RX logical by CLR[0]/CLR[1] and wait clear operation done before configure the other registers.

Chapter 24 I2C Interface

24.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.
I2C Controller supports the following features:
- Support 4 independent I2C: I2C0, I2C1, I2C2, I2C3
- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- Fiter out glitch on SCL and SDA

24.2 Block Diagram

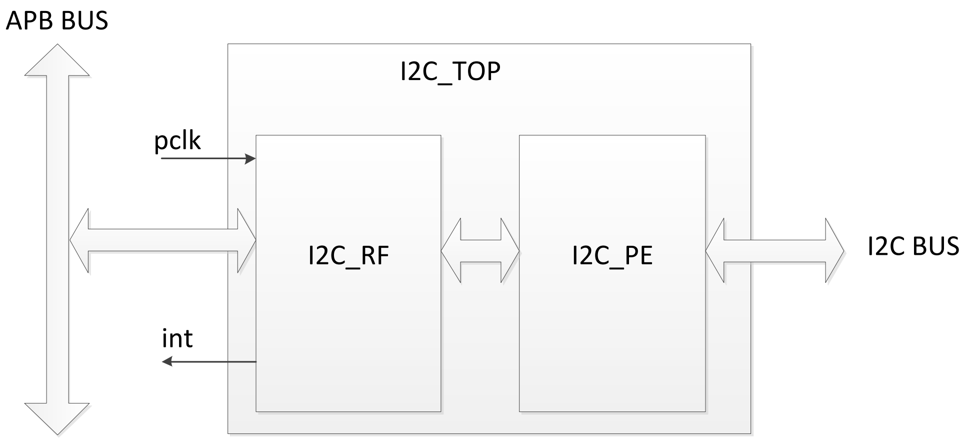


Fig.24-1I2C architecture

24.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

24.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

24.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

24.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.
The I2C controller supports only Masterfunction. Itsupports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can

be up to 400Kbit/sec.
The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

24.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the working clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

24.3.2 Master Mode Programming

- SCL Clock

When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:
SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)
SCL = PCLK/ SCLK Divisor

- Data Receiver Register Access

When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.
When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register

Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.
When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command

Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command

Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode

There are four i2c operation modes.

- When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
- When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
- When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
- When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command

- When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.
- In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].
- In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition

There are 7 interrupt bits in I2C_ISR register related to master mode.

- Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed

transmitting a byte.
- Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
- MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
- MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
- Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
- Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
- NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

- Last byte acknowledge control
- If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
- If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

- How to handle NAK handshake received
- If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
- If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

- I2C controller data transfer waveform
- Bit transferring
  - Data Validity

The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

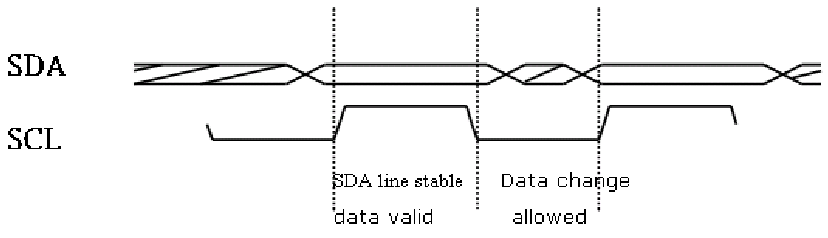


Fig.24-2I2C DATA Validity

  - START and STOP conditions

START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

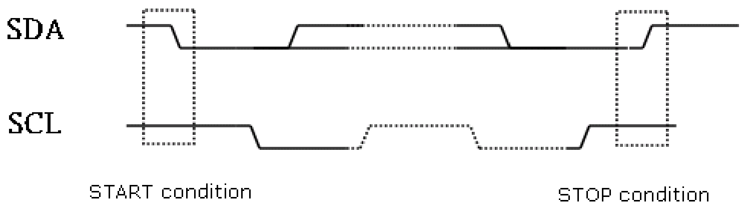


Fig.24-3I2C Start and stop conditions

  - Data transfer
    - Acknowledge

After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means “ACK”, on the contrary, it’s “NOT ACK”.

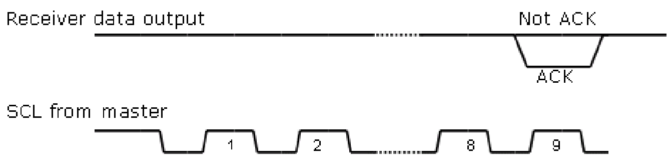


Fig.24-4I2C Acknowledge

- Byte transfer

The master own I2C bus might initiate multi byte to transfer to a slave.The transfer starts from a “START” command and ends in a “STOP”command. After every byte transfer, the receiver must reply an ACK to transmitter.

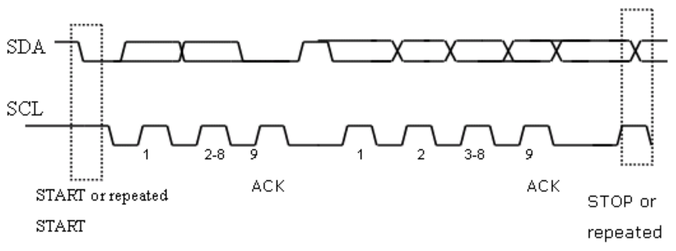


Fig.24-5I2C byte transfer

24.4 Register Description

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00030300	control register
RKI2C_CLKDIV	0x0004	W	0x00060006	clock divider register，I2C CLK = PCLK / (16*CLKDIV)
RKI2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master rx mode
RKI2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master rx mode
RKI2C_MTXCNT	0x0010	W	0x00000000	master transmit count.specify the total bytes to be transmit (0~32)
RKI2C_MRXCNT	0x0014	W	0x00000000	master rx count.specify the total bytes to be recieved(0~32)
RKI2C_IEN	0x0018	W	0x00000000	interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	interrupt pending register
RKI2C_FCNT	0x0020	W	0x00000000	finished count: the count of data which has been transmitted or receivedfor debug purpose
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	slave hold debounce configure register
RKI2C_TXDATA0	0x0100	W	0x00000000	I2C tx data register 0
RKI2C_TXDATA1	0x0104	W	0x00000000	I2C tx data register 1
RKI2C_TXDATA2	0x0108	W	0x00000000	I2C tx data register 2
RKI2C_TXDATA3	0x010c	W	0x00000000	I2C tx data register 3

Name	Offset	Size	Reset Value	Description
RKI2C_TXDATA4	0x0110	W	0x00000000	I2C tx data register 4
RKI2C_TXDATA5	0x0114	W	0x00000000	I2C tx data register 5
RKI2C_TXDATA6	0x0118	W	0x00000000	I2C tx data register 6
RKI2C_TXDATA7	0x011c	W	0x00000000	I2C tx data register 7
RKI2C_RXDATA0	0x0200	W	0x00000000	I2C rx data register 0
RKI2C_RXDATA1	0x0204	W	0x00000000	I2C rx data register 1
RKI2C_RXDATA2	0x0208	W	0x00000000	I2C rx data register 2
RKI2C_RXDATA3	0x020c	W	0x00000000	I2C rx data register 3
RKI2C_RXDATA4	0x0210	W	0x00000000	I2C rx data register 4
RKI2C_RXDATA5	0x0214	W	0x00000000	I2C rx data register 5
RKI2C_RXDATA6	0x0218	W	0x00000000	I2C rx data register 6
RKI2C_RXDATA7	0x021c	W	0x00000000	I2C rx data register 7
RKI2C_ST	0x0220	W	0x00000003	status debug register
RKI2C_DBGCTRL	0x0224	W	0x00000f00	Debug config register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

24.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0003	version rki2c version information
15:14	RW	0x0	stop_setup stop setup config: TSU;sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
13:12	RW	0x0	start_setup start setup config: TSU;sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD;sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
11	RO	0x0	reserved
10:8	RW	0x0	data_upd_st SDA update point config: Used to config sda change state when scl is low, used to adjust setup/hold time 4'bn:Thold = (n + 1) * Tclk_i2c Note: 0 <= n <= 5
7	RO	0x0	reserved
6	RW	0x0	act2nak operation when NAK handshake is received: 1'b0: ignored 1'b1: stop transaction

Bit	Attr	Reset Value	Description
5	RW	0x0	ack last byte acknowledge control in master receive mode: 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop stop enable, when this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start start enable, when this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode i2c mode select: 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable: 1'b0:not enable 1'b1:enable

RKI2C_CLKDIV
Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count: T(SCL_HIGH) = Tclk_i2c * (CLKDIVH + 1) * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count: T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8

RKI2C_MRXADDR
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid: 1'b0:invalid 1'b1:valid

Bit	Attr	Reset Value	Description
25	RW	0x0	addmvld address middle byte valid: 1'b0:invalid 1'b1:valid
24	RW	0x0	addlvld address low byte valid: 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	saddr master address register. the lowest bit indicate write or read 24 bits address register

RKI2C_MRXRADDR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid: 1'b0:invalid 1'b1:valid
25	RW	0x0	sraddmvld address middle byte valid: 1'b0:invalid 1'b1:valid
24	RW	0x0	sraddlvld address low byte valid: 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	sraddr slave register address accessed. 24 bits register address

RKI2C_MTXCNT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt master transmit count. 6 bits counter

RKI2C_MRXCNT
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master rx count. 6 bits counter

RKI2C_IEN
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclen slave hold scl interrupt enable: 1'b0:disable 1'b1:enable
6	RW	0x0	nakrcvien NAK handshake received interrupt enable: 1'b0:disable 1'b1:enable
5	RW	0x0	stopien stop operation finished interrupt enable: 1'b0:disable 1'b1:enable
4	RW	0x0	startien start operation finished interrupt enable: 1'b0:disable 1'b1:enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable: 1'b0:disable 1'b1:enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable: 1'b0:disable 1'b1:enable
1	RW	0x0	brfien byte rx finished interrupt enable: 1'b0:disable 1'b1:enable
0	RW	0x0	btfien byte tx finished interrupt enable: 1'b0:disable 1'b1:enable

RKI2C_IPD

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclipd slave hold scl interrupt pending bit: 1'b0:no interrupt available 1'b1:slave hold scl interrupt appear, write 1 to clear
6	W1C	0x0	nakrcvipd NAK handshake received interrupt pending bit: 1'b0:no interrupt available 1'b1:NAK handshake received interrupt appear, write 1 to clear
5	W1C	0x0	stopipd stop operation finished interrupt pending bit: 1'b0:no interrupt available 1'b1:stop operation finished interrupt appear, write 1 to clear
4	W1C	0x0	startipd start operation finished interrupt pending bit: 1'b0:no interrupt available 1'b1:start operation finished interrupt appear, write 1 to clear
3	W1C	0x0	mbrfipd MRXCNT data received finished interrupt pending bit: 1'b0:no interrupt available 1'b1:MRXCNT data received finished interrupt appear, write 1 to clear
2	W1C	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit: 1'b0:no interrupt available 1'b1:MTXCNT data transfer finished interrupt appear, write 1 to clear
1	W1C	0x0	brfipd byte rx finished interrupt pending bit: 1'b0:no interrupt available 1'b1:byte rx finished interrupt appear, write 1 to clear
0	W1C	0x0	btfipd byte tx finished interrupt pending bit: 1'b0:no interrupt available 1'b1:byte tx finished interrupt appear, write 1 to clear

RKI2C_FCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	fcnt the count of data which has been transmitted or received for debug purpose

RKI2C_SCL_OE_DB
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	scl_oe_db slave hold scl debounce. cycles for debounce (unit: Tclk_i2c)

RKI2C_TXDATA0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 data0 to be transmitted. 32 bits data

RKI2C_TXDATA1
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 data1 to be transmitted. 32 bits data

RKI2C_TXDATA2
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 data2 to be transmitted. 32 bits data

RKI2C_TXDATA3
Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 data3 to be transmitted. 32 bits data

RKI2C_TXDATA4
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 data4 to be transmitted. 32 bits data

RKI2C_TXDATA5
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 data5 to be transmitted. 32 bits data

RKI2C_TXDATA6
Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 data6 to be transmitted. 32 bits data

RKI2C_TXDATA7
Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 data7 to be transmitted. 32 bits data

RKI2C_RXDATA0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0 data0 received. 32 bits data

RKI2C_RXDATA1
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1 data1 received. 32 bits data

RKI2C_RXDATA2
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2 data2 received. 32 bits data

RKI2C_RXDATA3
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3 data3 received. 32 bits data

RKI2C_RXDATA4
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4 data4 received. 32 bits data

RKI2C_RXDATA5
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5 data5 received. 32 bits data

RKI2C_RXDATA6
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6 data6 received. 32 bits data

RKI2C_RXDATA7
Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7 data7 received. 32 bits data

RKI2C_ST
Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	scl_st scl status: 1'b0: scl status low 1'b0: scl status high

Bit	Attr	Reset Value	Description
0	RO	0x0	sda_st sda status: 1'b0: sda status low 1'b0: sda status high

RKI2C_DBGCTRL
Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	h0_check_scl 0: Check if scl been pull down by slave at the whole SCL_HIGH. 1: Check if scl been pull down by slave only at the h0 of SCL_HIGH(SCL_HIGH including h0~h7).
13	RW	0x0	nak_release_scl 0: Hold scl as low when recieved nack 1: Release scl as high when recieved nack
12	RW	0x0	flt_en SCL edage glitch filter enable 0: disable 1: enable
11:8	RW	0x0	slv_hold_scl_th Slave hold scl threshold = slv_hold_scl_th * Tclk_i2c
7:4	RW	0x0	flt_r Filter scl rising edge glitches of width less than flt_r * Tclk_i2c
3:0	RW	0x0	flt_f Filter scl falling edge glitches of width less than flt_f * Tclk_i2c

24.5 Interface Description

Table 24-1I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	IO_I2C0sda_GPIO0B1pmuio2	GRF_GPIO0B_IOMUX_L[6:4]=3’b001
i2c0_scl	I/O	IO_I2C0scl_GPIO0B0pmuio2	GRF_GPIO0B_IOMUX_L[2:0]=3’b001
I2C1 Interface			
i2c1_sda	I/O	IO_I2C1sda_UART3rtsm0_GPIO0C3pmuio2	GRF_GPIO0C_IOMUX_L[14:12]=2’b001
i2c1_scl	I/O	IO_I2C1scl_UART3ctsm0_PMUdebug5_GPIO0C2pmuio2	GRF_GPIO0C_IOMUX_L[10:8]=2’b001
I2C2 Interface			
i2c2_sda	I/O	IO_CIFd11m0_I2C2sda_GPIO2C0vccio3	GRF_GPIO2C_IOMUX_L[2:0]=3’b010
i2c2_scl	I/O	IO_CIFd10m0_I2C2scl_GPIO2B7vccio3	GRF_GPIO2B_IOMUX_H[14:12]=3’b010
I2C3 Interface			

i2c3_sda	I/O	IO_FLASHcle_UART3ctsm1_SPI0mosi_I2C3sda_GPIO1B4vccio0	GRF_GPIO1B_IOMUX_H[2:0]=3'b100
i2c3_scl	I/O	IO_FLASHwrn_UART3rtsm1_SPI0miso_I2C3scl_GPIO1B5vccio0	GRF_GPIO1BH_IOMUX_H[6:4]=3'b100

24.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode. Users are strongly advised to follow

- Transmit only mode (I2C_CON[1:0]=2'b00)

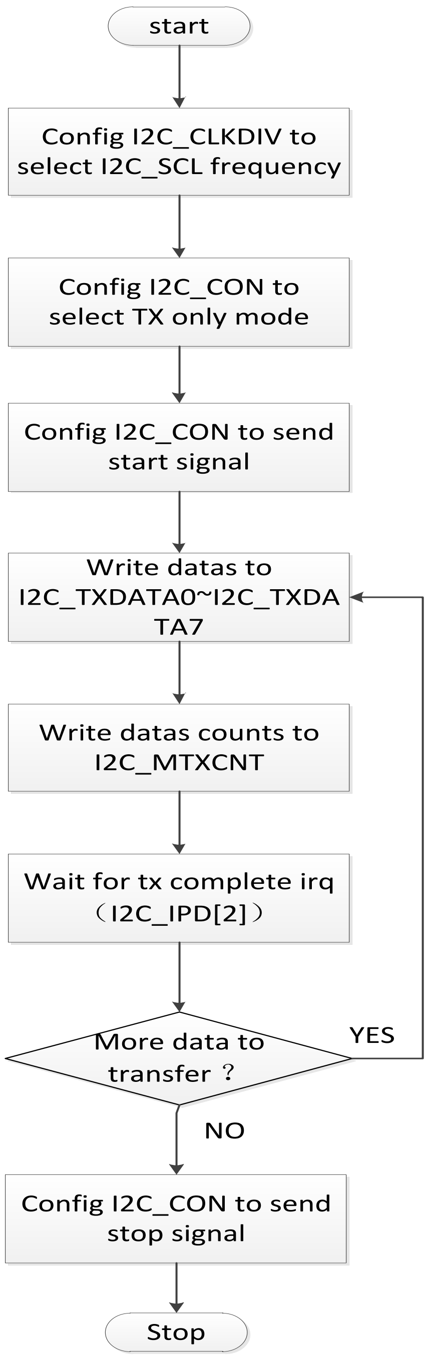

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select TX only mode
Config I2C_CON to send start signal
Write datas to I2C_TXDATA0~I2C_TXDATA7
Write datas counts to I2C_MTXCNT
Wait for tx complete irq （I2C_IPD[2]）
More data to transfer ？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig.24-6I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2’b10)

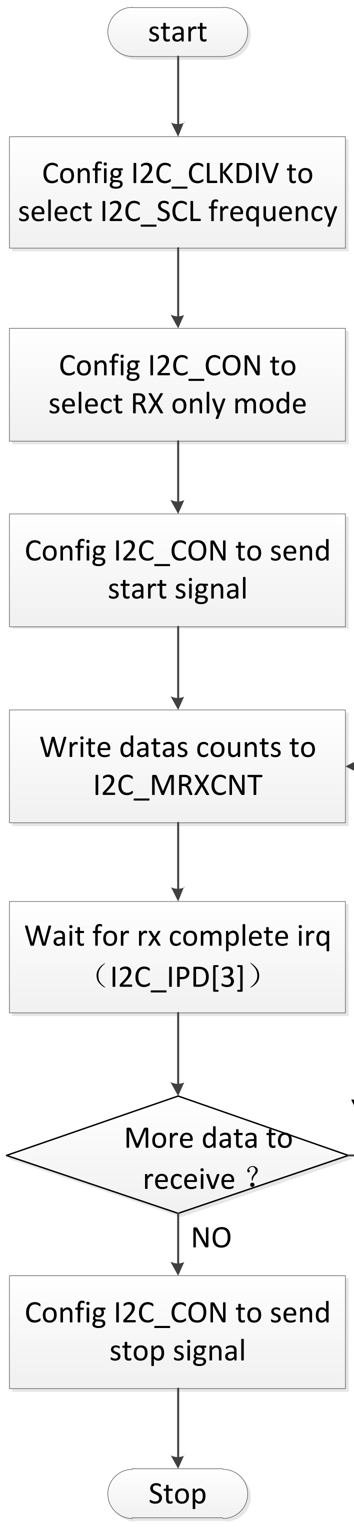

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select RX only mode
Config I2C_CON to send start signal
Write datas counts to I2C_MRXCNT
Wait for rx complete irq（I2C_IPD[3]）
More data to receive？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig.24-7I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

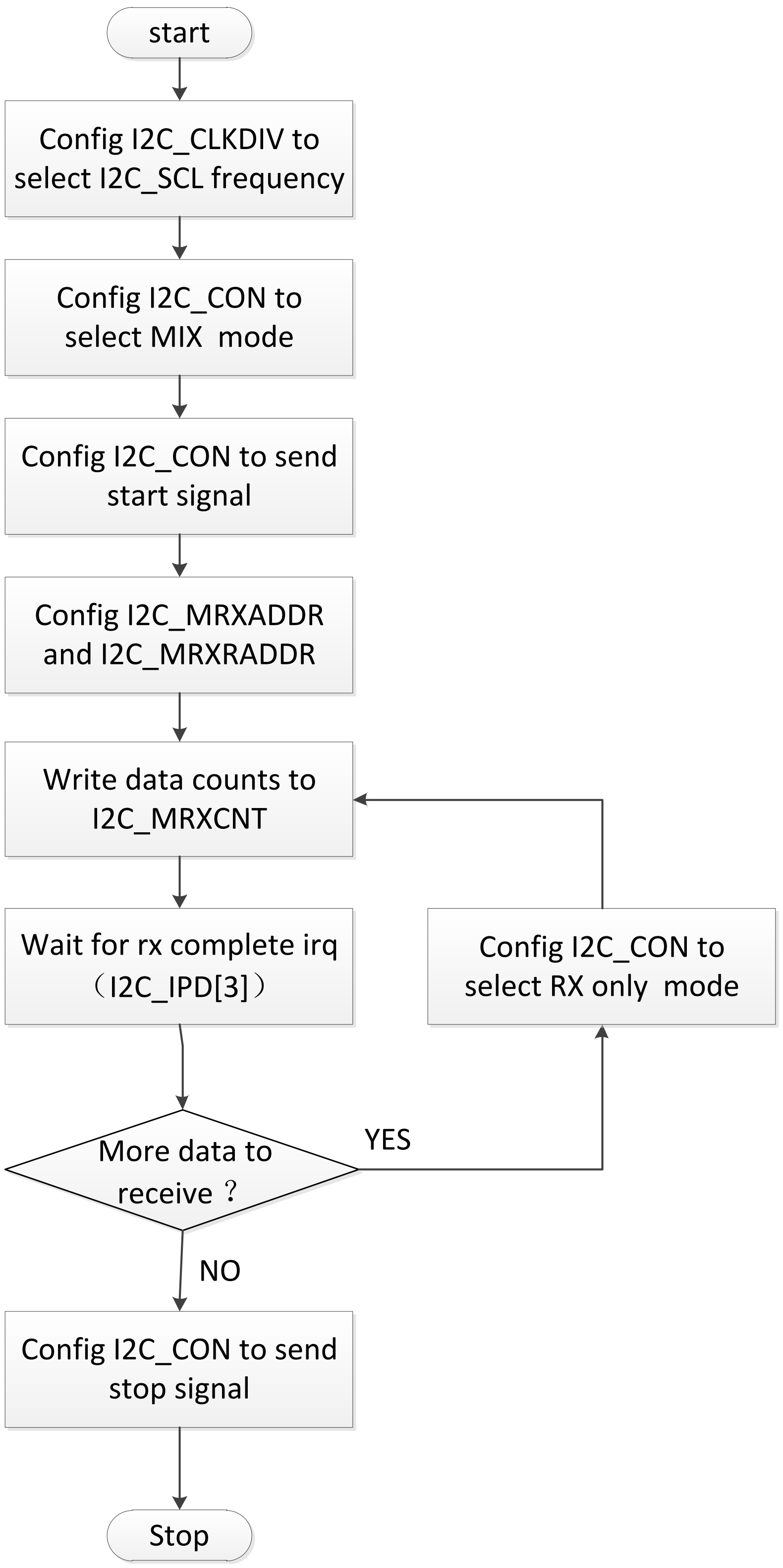

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select MIX mode
Config I2C_CON to send start signal
Config I2C_MRXADDR and I2C_MRXRADDR
Write data counts to I2C_MRXCNT
Wait for rx complete irq（I2C_IPD[3]）
Config I2C_CON to select RX only mode
More data to receive？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig.24-8I2C Flow chat for mix mode

Chapter 25 Audio Serial Port Controller (ASPC)

25.1 Overview

The Audio Serial Port Controller (ASPC) is a PDM interface controller and decoder that supportmono PDM format. It integrates a clock generator driving the PDM microphone and embeds filters which decimatethe incoming bitstream to obtain most common audio rates.
ASPC supports the following features:

- Support one internal 32-bit wide and 128-location deep FIFOsfor receiving audio data
- Support receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of receive FIFO full interrupt
- Support combined interrupt output
- Support AHB bus slave interface
- Support DMA handshaking interface and configurable DMA water level
- Support PDM master receive mode
- Support 4 paths. Each path is composed of two digital microphone channels, the ASPC can be used with four stereo or eight mono microphones. Each path is enabled or disabled independently
- Support 16 ~24 bit sample resolution
- Support sample rate:
  8khz,16khz,32kHz,64kHz,128khz,11.025khz,22.05khz,44.1khz,88.2khz,176.4khz,12khz,24khz,48khz,96khz,192khz
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support programmable data sampling sensibility (rising or falling edge)

25.2 Block Diagram

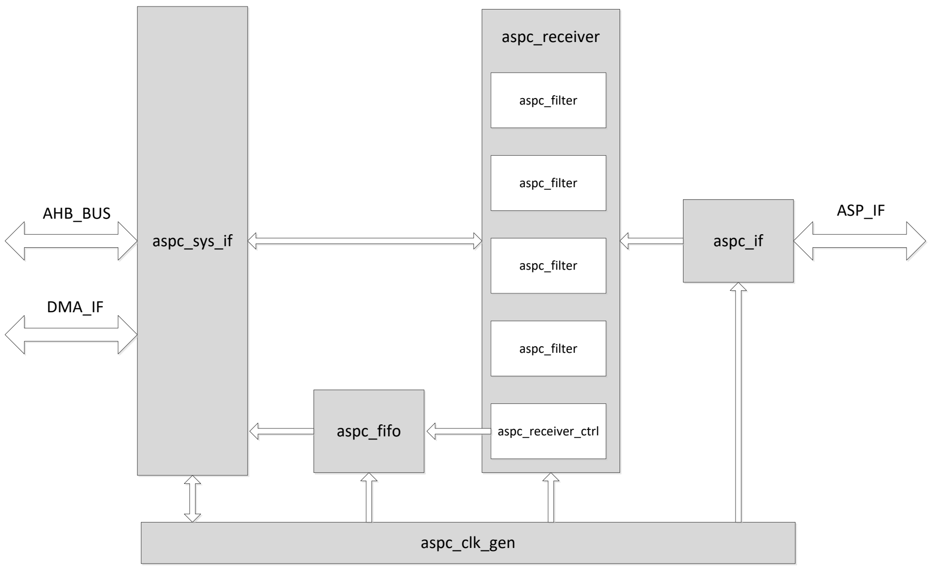


Fig. 25-1 ASPC Block Diagram

System Interface
The system interface implements the APB slave operation. It contains not only control registers of receiver inside but also interrupt and DMA handshaking interface.
Clock Generator
The Clock Generator implements clock generation function. The input source clock to the module is MCLK, and by the divider of the module, the clock generator generates CLK_PDM toreceiver.
Receiver
The receiver can act as a decimation filter of PDM. And export PCM format data.
Receive FIFO
The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 128.
ASP interface
The ASP interface implements PDM bit streams receive operation.

25.3 Function Description

25.3.1 AHB Interface

There is an AHB slave interface in ASPC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in 1.4.1.

25.3.2 PDM Interface

The PDM interface is a 5-wire interface. The ASPC module can support up to four external stereo and eight digital microphones.
Fig.1-2 and Fig.1-3 show two cases of use of the ASPC, but all configurations are possible with stereo and mono digital microphones.

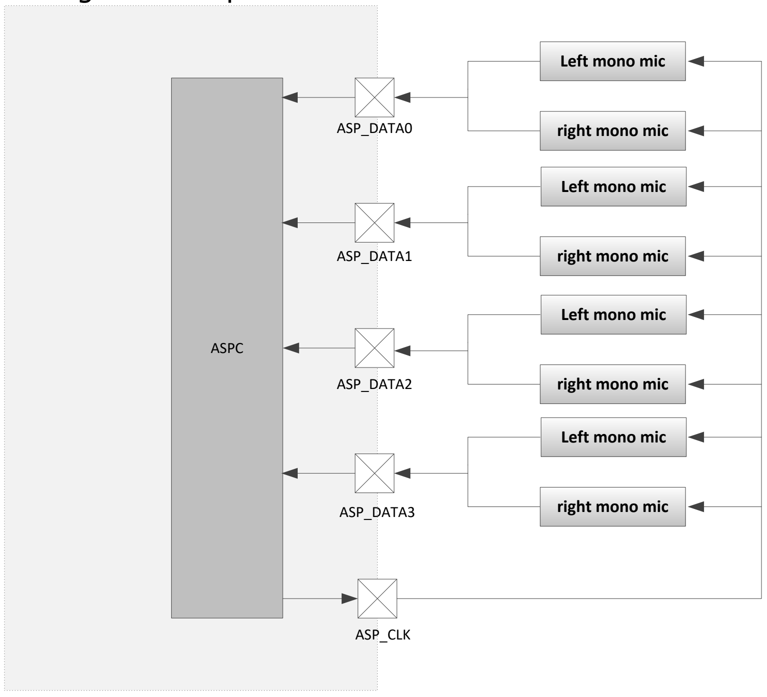


Fig. 25-2 ASPC with Eight Mono MIC

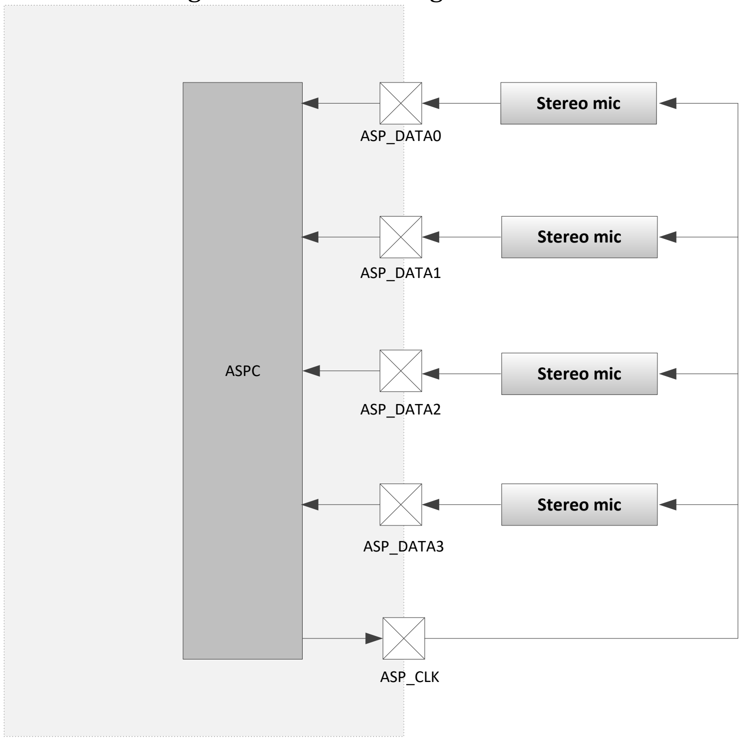


Fig. 25-3 ASPC with Four Stereo MIC

The PDM interface consists of a serial-data shift clock output (ASP_CLK) and a serial data input (ASP_DATA). The clock is fanned out to both digital mics, and both digital mics' data (left channel and right channel) outputs share a single signal line. To share a single line, the digital mics tristate their output during one phase of the clock(high or low part of cycle, depending on how they are configured via their L/R input).

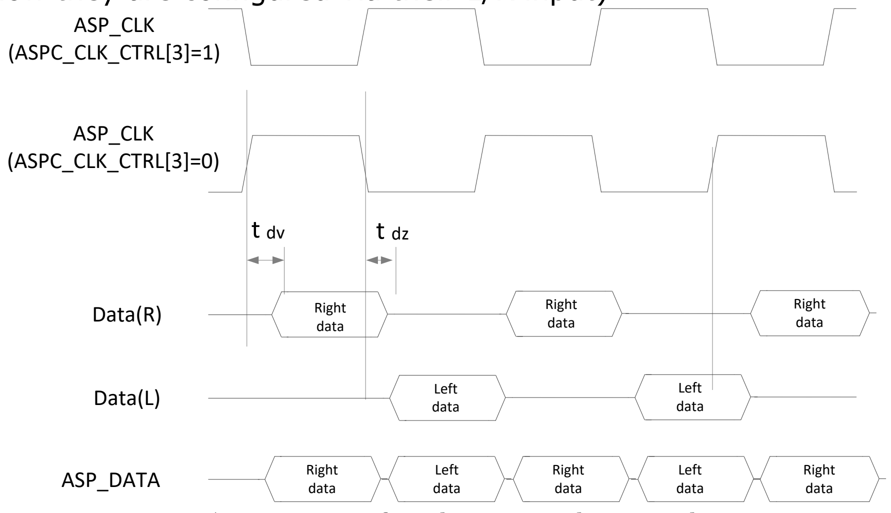


Fig. 25-4 ASPC interface diagram with external MIC

25.3.3 Digital Filter

The external PDMIC generates a PDM stream of bits and transfers it in one period or one half-period of the clock provided by the ASPC. The aim of the ASPC is to process data from the PDM interface, decimate and filter the data, and store the processed data in the FIFO. The four paths are identical. Each path is composed of a left and a right channel. The PDM interface delivers eight parallel data of 1bit. Each bit goes to a filter. The aim of the filter is to limit the noise and export PCM format audio data.

25.3.4 Clock Configuration

MCLK is the source clock signal. ASP_CLK is the output clocks generated in the ASPC and is fed to the external microphones. They are also the internal clock of the external microphones. User must take care about the value of ASP_CLK when selecting the source clock (MCLK).

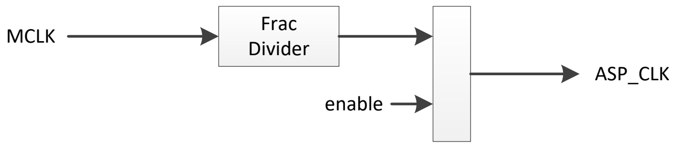


Fig. 25-5 ASPC Clock Structure

Table 25-1 Relation between ASP_CLK and sample rate

ASP_CLK	Sample rate
3.072Mhz	12khz,24khz,48khz,96khz,192khz
2.8224Mhz	11.025khz,22.05khz,44.1khz,88.2khz,176.4khz
2.048Mhz	8khz,16khz,32kHz,64kHz,128khz

User must configure the frac_div_con depended on the frequency of Mclk. If Mclk/acp_clk is more than 40, ASPC_CLK_CTRL[6] should set to 1;

25.4 Register Description

25.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
ASPC_SYSCONFIG	0x0000	W	0x00000000	ASPC system config register
ASPC_CTRL0	0x0004	W	0x78000017	ASPC control register 0
ASPC_CTRL1	0x0008	W	0x0bb8ea60	ASPC control register 1
ASPC_CLK_CTRL	0x000c	W	0x00000000	ASPC clock control register

Name	Offset	Size	Reset Value	Description
ASPC_HPF_CTRL	0x0010	W	0x00000000	ASPC high pass filter control register
ASPC_FIFO_CTRL	0x0014	W	0x00000000	ASPC fifo control register
ASPC_DMA_CTRL	0x0018	W	0x0000001f	ASPC dma control register
ASPC_INT_EN	0x001c	W	0x00000000	ASPC interrupt enable register
ASPC_INT_CLR	0x0020	W	0x00000000	ASPC interrupt clear register
ASPC_INT_ST	0x0024	W	0x00000000	ASPC interrupt status register
ASPC_RXFIFO_DATA_REG	0x0030	W	0x00000000	ASPC receive fifo data register
ASPC_DATA0R_REG	0x0034	W	0x00000000	ASPC path0 right channel data register
ASPC_DATA0L_REG	0x0038	W	0x00000000	ASPC path0 left channel data register
ASPC_DATA1R_REG	0x003c	W	0x00000000	ASPC path1 right channel data register
ASPC_DATA1L_REG	0x0040	W	0x00000000	ASPC path1 left channel data register
ASPC_DATA2R_REG	0x0044	W	0x00000000	ASPC path2 right channel data register
ASPC_DATA2L_REG	0x0048	W	0x00000000	ASPC path2 left channel data register
ASPC_DATA3R_REG	0x004c	W	0x00000000	ASPC path3 right channel data register
ASPC_DATA3L_REG	0x0050	W	0x00000000	ASPC path3 left channel data register
ASPC_DATA_VALID	0x0054	W	0x00000000	path data valid register
ASPC_VERSION	0x0058	W	0x59313030	ASPC version register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

25.4.2 Detail Register Description

ASPC_SYSCONFIG

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	rx_start RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	rx_clr ASPC RX logic clear; This is a self cleard bit. High active. Write 0x1: clear RX logic Write 0x0: no action Read 0x1: clear ongoing Read 0x0: clear done

ASPC_CTRL0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RW	0x0	sjm_sel Store justified mode: (Can be written only when SYSCONFIG[2] is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 1.Because if HWT is 0, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
30	RW	0x1	path3_en Path 3 enable; 1'b1: enable 1'b0: disable
29	RW	0x1	path2_en Path 2 enable; 1'b1: enable 1'b0: disable
28	RW	0x1	path1_en Path 1 enable; 1'b1: enable 1'b0: disable
27	RW	0x1	path0_en Path 0 enable; 1'b1: enable 1'b0: disable
26	RW	0x0	hwt_en HWT Halfword word transform Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
25:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x17	data_vld_width (Can be written only when SYSCONFIG[2] is 0.) Valid Data width 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 23:24bit

ASPC_CTRL1
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0bb8	frac_div_numerator fraction divider numerator; (Can be written only when SYSCONFIG[2] is 0.)
15:0	RW	0xea60	frac_div_denomonator fraction divider denominator; (Can be written only when SYSCONFIG[2] is 0.)

ASPC_CLK_CTRL
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	frac_div_ratio_sel fraction clk divider ratio select: (Can be written only when SYSCONFIG[2] is 0.) 0: ratio is more than 40; 1: ratio is less than 35;
5	RW	0x0	pdm_clk_en Pdm clk enable.working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 0:pdm clk disable 1:pdm clk enable
4	RO	0x0	reserved
3	RW	0x0	clk_polar ASP_CLK polarity selection (Can be written only when SYSCONFIG[2] is 0.) 0: no inverted 1: inverted

Bit	Attr	Reset Value	Description
2:0	RW	0x0	pdm_ds_ratio DS_RATIO,working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 3'b000: sample rate 192k/176.5k/128k 3'b001: sample rate 96kk/88.2k/64k 3'b010: sample rate 48kk/44.1k/32k 3'b011: sample rate 24kk/22.05k/16k 3'b100: sample rate 12kk/11.025k/8k

ASPC_HPF_CTRL
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	hpfle HPFLE high pass filter enable for left channel 1'b0: high pass filter for right channel is disabled. 1'b1: high pass filter for right channel is enabled.
2	RW	0x0	hpfre HPFRE high pass filter enable for right channel 1'b0: high pass filter for right channel is disabled. 1'b1: high pass filter for right channel is enabled.
1:0	RW	0x0	hpf_cf HPF_CF high pass filter configure register high pass filter configure register 2'b00: 3.79Hz 2'b01: 60Hz 2'b10: 243Hz 2'b11: 493Hz

ASPC_FIFO_CTRL
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	rft Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO threshold interrupt is triggered.

Bit	Attr	Reset Value	Description
7:0	RO	0x00	rfl RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

ASPC_DMA_CTRL
Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	rde Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
7	RO	0x0	reserved
6:0	RW	0x1f	rdl Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.

ASPC_INT_EN
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	rxoie RX overflow interrupt enable 0:disable 1:enable
0	RW	0x0	rxtie RX threshold interrupt enable 0:disable 1:enable

ASPC_INT_CLR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	rxoic RX overflow interrupt clear, high active, auto clear.
0	RO	0x0	reserved

ASPC_INT_ST
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	rxoi RX overflow interrupt 0:inactive 1:active
0	RO	0x0	rxfi RX full interrupt 0:inactive 1:active

ASPC_RXFIFO_DATA_REG
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdr Receive FIFO shadow Register When the register is read, data in the receive FIFO is accessed.

ASPC_DATA0R_REG
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0r Data of the path 0 right channel

ASPC_DATA0L_REG
Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0l Data of the path 0 left channel

ASPC_DATA1R_REG
Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	data1r Data of the path 1 right channel

ASPC_DATA1L_REG
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data1l Data of the path 1 left channel

ASPC_DATA2R_REG

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2r Data of the path 2 right channel

ASPC_DATA2L_REG
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2l Data of the path 2 left channel

ASPC_DATA3R_REG
Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3r Data of the path 3 right channel

ASPC_DATA3L_REG
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3l Data of the path 3 left channel

ASPC_DATA_VALID
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

3	RC	0x0	path0_vld
			0: DATA0R_REG, DATA0L_REG value is invalid;
			1: DATA0R_REG, DATA0L_REG value is valid;

2	RC	0x0	path1_vld
			0: DATA1R_REG, DATA1L_REG value is invalid;
			1: DAT1R_REG, DATA1L_REG value is valid;

1	RC	0x0	path2_vld
			0: DATA2R_REG, DATA2L_REG value is invalid;
			1: DATA2R_REG, DATA2L_REG value is valid;

0	RC	0x0	path3_vld
			0: DATA3R_REG, DATA3L_REG value is invalid;
			1: DATA3R_REG, DATA3L_REG value is valid;

ASPC_VERSION
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x59313030	version ASPC version

25.5 Interface Description

Table 25-2ASPC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
O_asp_clk	O	IO_PDMclk0m1_GPIO2C6vccio5/ IO_LCDCd18_PDMclk0m0_CIFd10m1_GPIO3C6vccio4/ IO_LCDCd19_PDMclk1_CIFd11m1_GPIO3C7vcci4	PDMclk0m1: GRF_GPIO2C_IOMUX_H[10:8]=1 PDMclkm1: GRF_GPIO3C_IOMUX_H[10:8]=2 PDMclk1: GRF_GPIO3C_IOMUX_H [14:12]=2
I_asp_data0	I	IO_I2S12ch_sdi_PDMsdi0m1_GPIO2C5vccio5/ IO_LCDd23_PDMsdi0m0_CIFclkinm1_ISPfl_trig_GPIO3D3vccio4	PDMsdi0m0: GRF_GPIO3D_IOMUX_L [14:12]=2 PDMsdi0m1: GRF_GPIO2C_IOMUX_H [6:4]=2
I_asp_data1	I	IO_LCDCd20_PDMsdi1_CIFclkoutm1_GPIO3D0vccio4	GRF_GPIO3D_IOMUX_L [2:0]=2
I_asp_data2	I	IO_LCDCd21_PDMsdi2_CIFvsyncm1_ISPprelight_trig_GPIO3D1vccio4	GRF_GPIO3D_IOMUX_L [6:4]=2
I_asp_data3	I	IO_LCDCd22_PDMsdi3_CIFhrefm1_ISPflash_trig_GPIO3D2vccio4	GRF_GPIO3D_IOMUX_L [10:8]=2

Notes: I=input, O=output, I/O=input/output, bidirectional

Furthermore, different IOs are selected and connected to different flash interface, which is shown as follows.

25.6 Application Notes

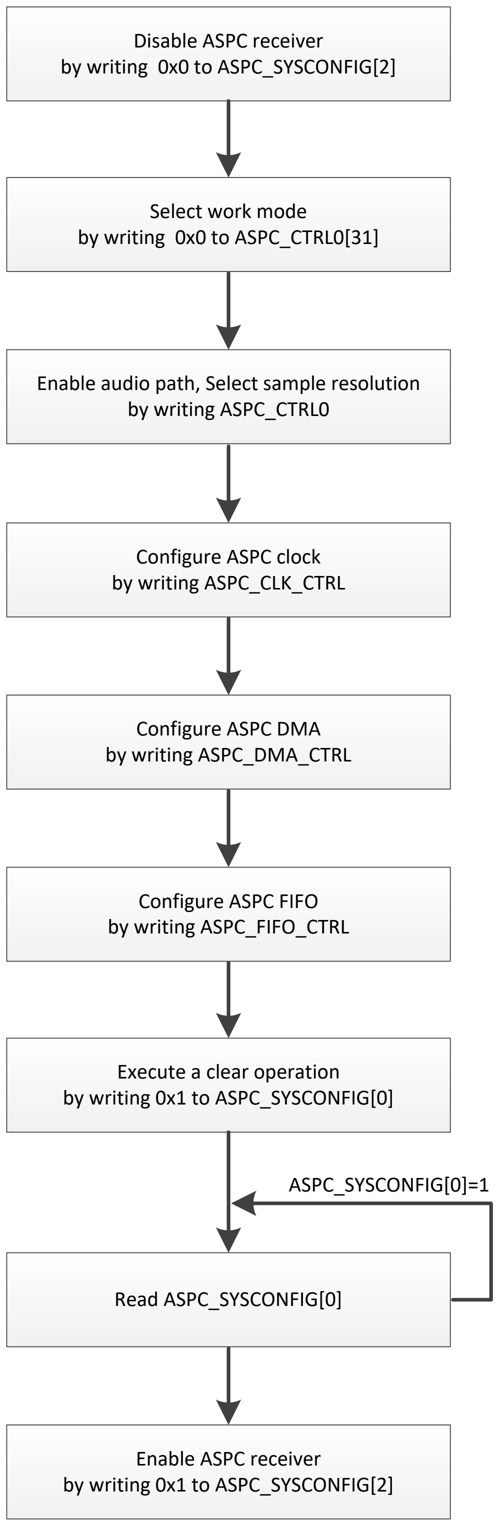

Disable ASPC receiver
by writing 0x0 to ASPC_SYSCONFIG[2]
Select work mode
by writing 0x0 to ASPC_CTRL0[31]
Enable audio path, Select sample resolution
by writing ASPC_CTRL0
Configure ASPC clock
by writing ASPC_CLK_CTRL
Configure ASPC DMA
by writing ASPC_DMA_CTRL
Configure ASPC FIFO
by writing ASPC_FIFO_CTRL
Execute a clear operation
by writing 0x1 to ASPC_SYSCONFIG[0]
ASPC_SYSCONFIG[0]=1
Read ASPC_SYSCONFIG[0]
Enable ASPC receiver
by writing 0x1 to ASPC_SYSCONFIG[2]

Fig. 25-6ASPC operation flow

Chapter 26 OTP

26.1 Overview

The One Time Programmable Controller (OTPC) is used for communication with the OTP subsystem to achieve the controlling command and receive the returning data. The configuration and command information are written from a master(CPU) over the APB bus to the OTPC and converted to standard form and transmitted to the OTP. The data from OTP can be stored in the registers of OTPC for the master (CPU) to read back.
OTP Controller supports the following features:

- Support APB interface
- Support OTP SBPI master interface
- Support OTP user master interface
- Support two programmable working clock for SBPI and user interface
- Support one interrupt output
- Support two busy signals
- SBPI master:
  - Support configurable device ID
  - Support maximum 32 consecutive valid command
  - Support CS automatic de-assert
  - Support CS manual de-assert
  - Support maximum 32B consecutive reading and storage
  - Support reading MISO and FLAG status by the APB bus
- User interface master:
  - Support software configurable DCTRL
  - Support single reading

26.2 Block Diagram

This section provides a description about the functions and behavior under various conditions. The OTP Controller comprises with:

- AMBA APB interface
- SBPI interface
- USER interface
- REG FILE
- SBPI FSM
- USER FSM

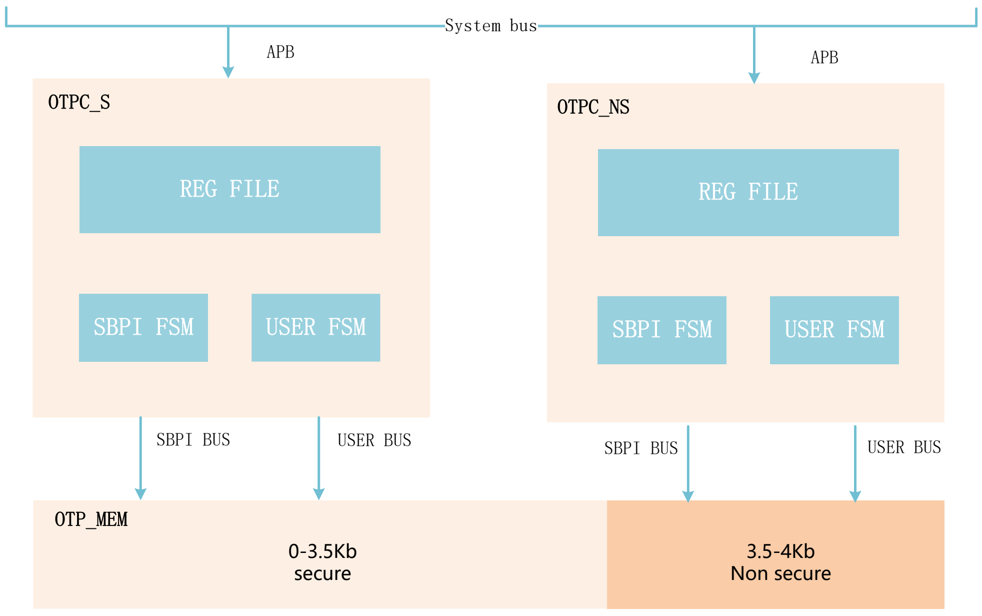


Fig. 26-1OTP Architecture

APB INTERFACE
The host processor accesses data, control, and status information on the OTPCthrough the APB interface including secure and non-secure.
Register file
Be responsible for the main OTPC functionality including control, status and interrupt generation.
SBPI FSM & USER FSM
The two FSMs are used for converting command to standard form and receiving data from SBPI and USER interface.
SBPI BUS & USER BUS
SBPI bus and USER bus are used for the transmission of command and the reception of data.
OTP MEMORY
There are two pieces of memory and each of them is 4Kb. The all 4Kb of MEM_0 is secure. The 0-3.5Kb of MEM_1 is secure while the remaining 0.5Kb is non-secure.

26.3 Function Description

SBPI Interface Protocol
The Sidense Serial-Parallel Interface, SBPI, defines a half-duplex serial and byte-parallel protocol in which instruction and data are transferred between SBPI agents. Attentively, OTPC only support byte-parallel mode and data are transferred between OTPC and SHF_AP. The SHF_AP is a synthesizable RTL block that interfaces with the Sidense SHF OTP memory and Integrated Power Supply (IPS) blocks.
The following description presents the SBPI communication and control protocol for byte-parallel mode. For byte-parallel mode, SP is held LOW.
A typical byte-parallel data transfer frame shown in Figure 1-2 consists of Start-of-Frame (SOF), a Frame Body, and an End-of-Frame (EOF). The IDLE time between frames is used to select the agent (ID).

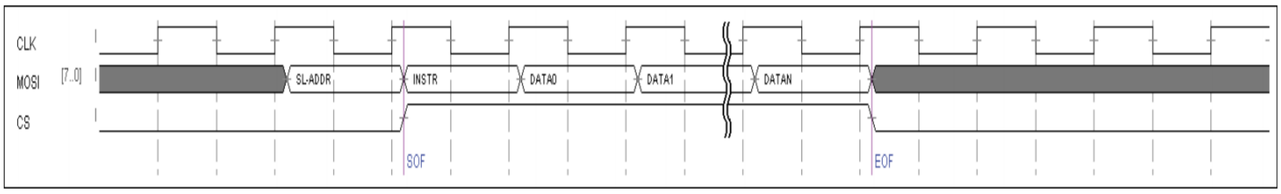


Fig. 26-2 OTP SBPI protocol

- SOF: A frame is started in a selected agent by the first cycle where CKE is HIGH, CS is HIGH and CLK rises; CS is made HIGH in a clock cycle following agent ID selection.
- EOF: A frame ends when CS is made LOW.
- BODY: The frame body consists of one or more clock cycles depending on instruction type. There are up to 3 phases for instruction sent as part of the frame:
  - Argument phase: 1 or more cycles to transfer needed arguments for the instruction to the target slave register
  - Action phase: number of cycles required by the execution engine to provide the required sequence of operations
  - Result phase: number of cycles required to transfer data from the slave’s registers to the master

USER Interface Protocol
When DCTRL is asserted LOW, the OTP access is under SBPI control while the data outputs Q and QP remain active and contain the results of the most recent operation. If no new read or data processing is performed using the DAP, the Q outputs contain the last read data.
The user interface is enabled by asserting DCTRL to HIGH. However, this does not disable the SBPI interface. Once DCTRL is asserted HIGH, the SHF address bus and the access strobe signals CK are under user control and the outputs Q and QP contain data read from OTP.
The user read cycle is controlled by the CK access strobe pulse width. A read cycle to the addressedword is initiated on the rising edge of the CK read strobe signal, when the OTP’s WE input is LOW(sourced internally from the DAP of the SHF_AP) and the OTP select input, SEL is

HIGH. The addressA[8:0] and select SEL inputs are latched on the rising edge of CK access strobe signal. The dataoutputs Q[7:0] and QP[7:0], become valid following the subsequent falling edge of the CK accessstrobe signal, or after the tACC if the SHF internal timer is enabled. If an output data bit does notchange state during a read operation, there is no intermediate transition at the output during the accesstime.

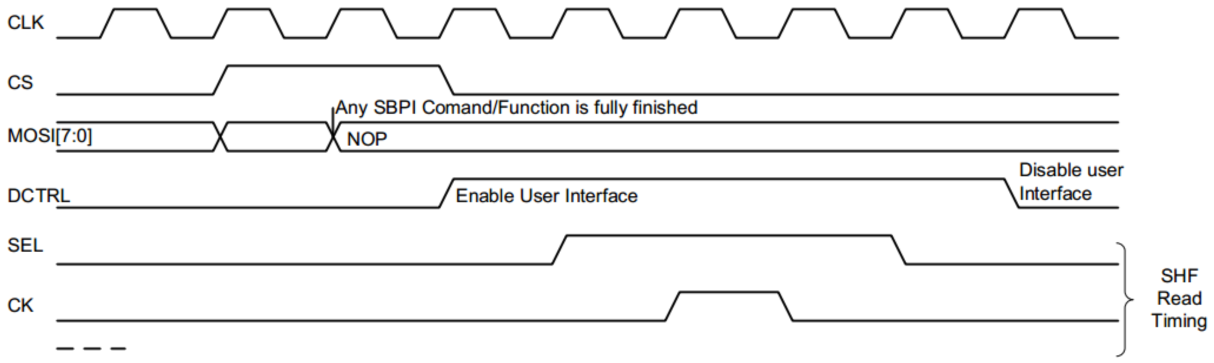


Fig. 26-3 OTP USER protocol

OTP BOOT Function

The BOOT function verifies power supply by performing ROM read operations, loads special registers to drive QSR signals (optional), and sets the default read mode into DAP. The user host controller may need to reload the DAP or PMC registers for the subsequent BIST, PROGRAM or READ operations. We can consult with SHF and IPS data sheets for correct mode input signal settings for read and program operations.

After both VCC and VDD have reached operating range levels, the RSTn should be released and theBOOT function can be invoked by issuing a START instruction to the PMC from the host function viathe SPBI interface, since the PMC and DAP are preset for BOOT operation.

The BOOT function’s purpose is to reliably set the internal Q_SR[n+7:0], Q_RR[n-1:0], and Q_RRP[7:0]registers with user defined content read from pre-defined OTP locations once supplies have reached asuitable level. The internal Q_SR[n+7:0] register bits are all preset to 1’b1 when RSTn is asserted LOW.The internal Q_RR[n-1:0] and Q_RRP[7:0] register bits are all preset to 1’b0 when RSTn is assertedLOW. The BOOT operation sets these internal registers with user content by first performing multiple“read and verify” operations of test ROM locations, until all ROM locations are read correctly. TheBOOT operation then loads the internal Q_SR[n+7:0], Q_RR[n-1:0] and Q_RRP[7:0] registers with dataread from the pre-defined SHF OTP locations.

The SHF_AP output signal FLAG can be used by the host controller to determine when the BOOToperation is completed in order to issue a STOP instruction to the PMC to terminate the BOOT function.At this point, the OTP can be accessed through the SBPI interface or, by asserting DCTRL to HIGH, theOTP can be accessed through the user interface to perform OTP read operations.

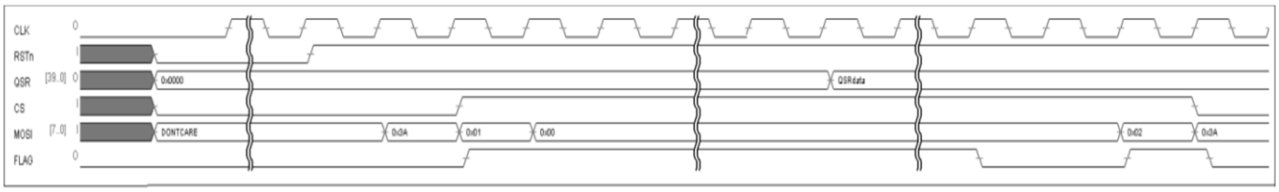


Fig. 26-4 OTP BOOT command

- 0x3A on MOSI = PMC SBPI address
- 0x00 on MOSI = PMC NOP instruction
- 0x01 on MOSI = PMC START instruction
- 0x02 on MOSI = PMC STOP instruction

OTP BIST Function

The Array Clean BIST function provides an array clean checking and bit repair capabilities. It isactivated similarly to the PROG function. The user host controller may need to provide controlparameters using register configurations before initiating the BIST function, select BIST in the PMCregister PMC_CTRL_STATUS and then issue the START instruction. The

PMC_CTRL_STATUSregister and the FLAG output signal will indicate routine completion and status. By default, the BIST canbe configured to run through the entire address space and to attempt to repair any bad bits in the array.Alternatively the BIST can be run through a selected portion of the address space, with or without bitrepair.

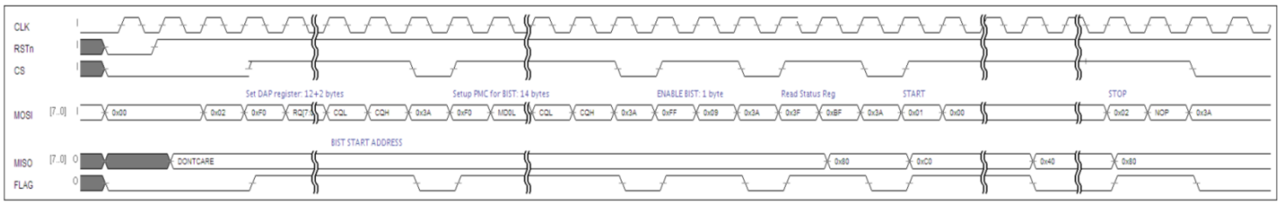


Fig. 26-5 OTP BIST command

OTP PROGRAM Function

In order to initiate programming the OTP, the host (user) must reload DAP and PMC registers withproper program and verify configurations, write the initial address and a data word into DAP, and selectthe program function in PMC. The programming starts when the host issues a START instruction to thePMC. The PMC takes over control of the SBPI bus to the DAP and asserts the FLAG HIGH, indicatingthat the programming operation is active. When the PMC reaches the end of the programmingoperation, the FLAG signal will be asserted LOW. The host can also monitor the PMC'sPMC_CTRL_STATUS register via the MISO bus while the programming operation is underway. ThePMC_CTRL_STATUS register indicates the end of programming operation, (same as the FLAG) andalso indicates if the program was successful using error codes. The host must issue the STOPinstruction to the PMC which sets the FLAG bit back to HIGH and terminates the programming function.It then re-gains control over the whole SBPI bus, following which the next data word can be written intothe DAP register. The OTP address can be incremented automatically using a PMC setting, unless thehost overrides it. The ECC parity bits are added automatically to the data provided by the user, andprogrammed into the OTP, unless the ECC is disabled.

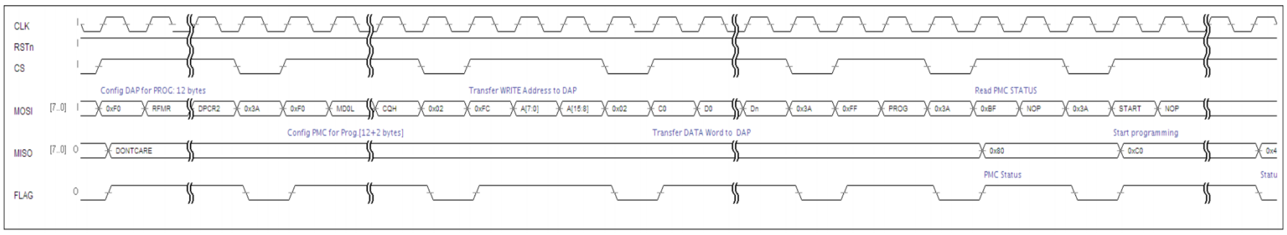

CLK
RSTn
CS
MOSI
MISO
FLAG
Config DAP for PROG: 12 bytes
Config PMC for Prog.[12+2 bytes]
Transfer WRITE Address to DAP
Transfer DATA Word to DAP
Read PMC STATUS
Start programming
PMC Status
DONTCARE

Fig. 26-6 OTP PROG command

26.4 Register Description

26.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
OTPC_SBPI_CTRL	0x0020	W	0x00000000	OTPC SBPI control register
OTPC_SBPI_CMD_VALID_PRELOAD	0x0024	W	0x00000000	OTPC SBPI command preload register
OTPC_SBPI_CS_VALID_PRELOAD	0x0028	W	0x00000000	OTPC SBPI CS valid parameter preload register
OTPC_SBPI_STATUS	0x002c	W	0x00000000	OTPC SBPI status register
OTPC_USER_CTRL	0x0100	W	0x00000000	OTPC USER control register
OTPC_USER_ADDR	0x0104	W	0x00000000	OTPC USER reading address register
OTPC_USER_ENABLE	0x0108	W	0x00000000	OTPC USER enable register
OTPC_USER_STATUS	0x0110	W	0x00000000	OTPC USER status register
OTPC_USER_QP	0x0120	W	0x00000000	OTPC USER QP storage register

Name	Offset	Size	Reset Value	Description
OTPC_USER_Q	0x0124	W	0x00000000	OTPC USER Q storage register
OTPC_USER_QSR	0x0128	W	0x00000000	OTPC USER QSR storage register
OTPC_USER_QRR	0x012c	W	0x00000000	OTPC USER QRR storage register
OTPC_INT_CON	0x0300	W	0x00000000	OTPC interrupt register
OTPC_INT_STATUS	0x0304	W	0x00000000	OTPC interrupt status register
OTPC_SBPI_CMD_BASE	0x1000	W	0x00000000	SBPI_CMD will be programmable from offset 0x1000 to 0x2000, which is 4kBAnd there are 1024 registers totally, which are correspond to a certain command.The address of these registers are:0x10000x1004......0x1ffc
OTPC_SBPI_READ_DATA_BASE	0x2000	W	0x00000000	There are 1024 registers which are all 32bit. They are mapped to OTPC_SBPI_CMD registers, if the corresponding command is a read command, the read data will be captured in the matched OTPC_SBPI_READ_DATA registers. The address of these registers are:0x20000x2004

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

26.4.2 Detail Register Description

OTPC_SBPI_CTRL

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15:8	RW	0x00	sbpi_device_id device id value, user to choose a device
7:4	RO	0x0	reserved
3	R/W SC	0x0	sbpi_cs_deassert write 1 to this bit will deassert cs this bit will be selfclear, do not write 0 to this bit
2	RW	0x0	sbpi_cs_auto 1'b0: cs deassert only under software control 1'b1: cs deassert under software control and will be automatically deassert when a cs_counter reach 0
1	RW	0x0	sbpi_sp sp control of sbpi bus 1'b0: parallel mode 1'b1: serial mode(controller not support)
0	R/W SC	0x0	sbpi_enable write 1 to this register enable sbpi FSM enable It will be selfclear to 0, software do not write 0 to this bit

OTPC_SBPI_CMD_VALID_PRELOAD
Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15:0	RW	0x0000	otpc_sbpi_cmd_valid_preload a value define number of sbpi valid command

OTPC_SBPI_CS_VALID_PRELOAD
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15:0	RW	0x0000	otpc_sbpi_cs_valid_preload a value define number of cs valid cycles, when sbpi_cs_auto is set to 1 of OTPC_SBPI_CTRL

OTPC_SBPI_STATUS
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RO	0x0	SP SP value of SBPI bus
21	RO	0x0	CS CS value of SBPI bus
20:13	RO	0x00	MOSI MOSI value of SBPI bus
12:5	RO	0x00	MISO MISO value of SBPI bus
4	RO	0x0	FLAG FLAG state of SBPI interface
3:1	RO	0x0	sbpi_current_state FSM states of SBPI
0	RO	0x0	sbpi_busy sbpi_busy status

OTPC_USER_CTRL
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	user_pd 1'b0: PD of user interface will be set to 0 1'b1: PD of user interface will be set to 1
0	RW	0x0	user_dctrl 1'b0: DCTRL of user interface will be set to 0 1'b1: DCTRL of user interface will be set to 1

OTPC_USER_ADDR
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15:0	RW	0x0000	otpc_user_addr 16bit A of User interface

OTPC_USER_ENABLE
Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15:1	RO	0x0	reserved
0	RW	0x0	otpc_user_enable write 1 to enable USER FSM will be selfclear, do not write 0 to this bit

OTPC_USER_STATUS
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RO	0x0	DCTRL DCTRL of USER interface
22:7	RO	0x0000	A A of USER interface
6	RO	0x0	PD PD of USER interface
5	RO	0x0	reserved
4	RO	0x0	SEL SEL of USER interface
3:1	RO	0x0	user_current_state state of USER FSM
0	RO	0x0	user_busy user_busy indication

OTPC_USER_QP
Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	QP QP value of USER interface

OTPC_USER_Q
Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	Q Q value of USER interface

OTPC_USER_QSR
Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	QSR QSR value of USER interface

OTPC_USER_QRR
Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	QRR QRR value of USER interface

OTPC_INT_CON

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bit write mask for lower bits
15	RW	0x0	otpc_global_int_enable 1'b0 : disable all interrupt 1'b1 : enable all interrupt
14:3	RO	0x0	reserved
2	RW	0x0	user_done_int_enable 1'b0 : disable user done interrupt 1'b1 : enable user done interrupt
1	RW	0x0	sbpi_done_int_enable 1'b0 : disable sbpi done interrupt 1'b1 : enable sbpi done interrupt
0	RW	0x0	sbpi_flag_detect_int_enable 1'b0 : disable sbpi flag detect interrupt 1'b1 : enable sbpi flag detect interrupt

OTPC_INT_STATUS

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	user_done_int_status indicate a user done interrupt status
1	RW	0x0	sbpi_done_int_status indicate a sbpi done status
0	R/W SC	0x0	sbpi_flag_detect_int_status indicate detecting a flag negedge

OTPC_SBPI_CMD_BASE

Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	command_value contain value of the command

OTPC_SBPI_READ_DATA_BASE

Address: Operational Base + offset (0x2000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	sbpi_read_data read_data from sbpi bus

26.5 Application Notes

26.5.1 GRF Register Summary

GRF Register	Register Description
SGRF_SOC_CON2[12]	OTP CKE enable selection 1b'1:enable 1'b0:disable
SGRF_SOC_CON2[13]	OTP secure or non-secure selection 1'b1:secure 1'b0:non-secure