Rockchip
RV1106
Technical Reference Manual

Revision History

Date	Revision	Description
2022-03-24	0.3	Update Chapter CRU, GRF, MCU Subsystem and TSADC
2022-02-18	0.2	Update Chapter 4 General Register Files
2022-01-27	0.1	Initial Release

Table of Content

Figure Index

Table Index

Warranty Disclaimer

Rockchip Electronics Co., Ltd makes no warranty, representation or guarantee (expressed, implied, statutory, or otherwise) by or with respect to anything in this document, and shall not be liable for any implied warranties of non-infringement, merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Information furnished is believed to be accurate and reliable. However, Rockchip Electronics Co.,Ltd assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use.

Rockchip Electronics Co., Ltd's products are not designed, intended, or authorized for using as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Rockchip Electronics Co., Ltd's product could create a situation where personal injury or death may occur, should buyer purchase or use Rockchip Electronics Co., Ltd's products for any such unintended or unauthorized application, buyers shall indemnify and hold Rockchip Electronics Co., Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Rockchip Electronics Co., Ltd was negligent regarding the design or manufacture of the part.

Copyright and Patent Right

Information in this document is provided solely to enable system and software implementers to use Rockchip Electronics Co.,Ltd 's products. There are no expressed and patent or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Rockchip Electronics Co., Ltd does not convey any license under its copyright and patent rights nor the rights of others.
All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.

Trademarks

Rockchip and Rockchip™ logo and the name of Rockchip Electronics Co., Ltd's products are trademarks of Rockchip Electronics Co., Ltd. and are exclusively owned by Rockchip Electronics Co., Ltd. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Confidentiality

The information contained herein (including any attachments) is confidential. The recipient hereby acknowledges the confidentiality of this document, and except for the specific purpose, this document shall not be disclosed to any third party.

Reverse engineering or disassembly is prohibited.

ROCKCHIP ELECTRONICS CO.,LTD. RESERVES THE RIGHT TO MAKE CHANGES IN ITS PRODUCTS OR PRODUCT SPECIFICATIONS WITH THE INTENT TO IMPROVE FUNCTION OR DESIGN AT ANY TIME AND WITHOUT NOTICE AND IS NOT REQUIRED TO UNDATE THIS DOCUMENTATION TO REFLECT SUCH CHANGES.

Copyright © 2022 Rockchip Electronics Co., Ltd.

Chapter 1 System Overview

1.1 Address Mapping

RV1106 boot from internal BootRom, which supports remap function by software programming. Remap is controlled by PMU_SGRF_SOC_CON1[11:10]. When remap is set to 2'b01, the BootRom is un-accessible and PMU_SRAM is mapped to address 0xFFFF0000. When remap is set to 2'b10, the BootRom is un-accessible and SYSTEM_SRAM is mapped to address 0xFFFF0000.

Table 1-1 Address Mapping

Module	Start Address	Size	Module	Start Address	Size
PERI_GRF	FF000000	64KB	ACODEC PHY	FF480000	64KB
VENC_GRF	FF010000	32KB	PWM2	FF490000	64KB
NPU_GRF	FF018000	32KB	UART0	FF4A0000	64KB
PMU_GRF	FF020000	64KB	UART1	FF4B0000	64KB
DDR_GRF	FF030000	64KB	UART2	FF4C0000	64KB
CORE_GRF	FF040000	64KB	UART3	FF4D0000	64KB
VI_GRF	FF050000	64KB	UART4	FF4E0000	64KB
VO_GRF	FF060000	32KB	UART5	FF4F0000	64KB
RESERVED	FF068000	32KB	SPI0	FF500000	64KB
PERI_SGRF	FF070000	8KB	SPI1	FF510000	64KB
VI_SGRF	FF072000	8KB	DECOM	FF520000	64KB
NPU_SGRF	FF074000	8KB	GPIO1	FF530000	32KB
CPU_SGRF	FF076000	8KB	GPIO1_IOC	FF538000	32KB
RKVENC_SGRF	FF078000	8KB	GPIO2	FF540000	32KB
VO_SGRF	FF07A000	8KB	GPIO2_IOC	FF548000	32KB
RESERVED	FF07C000	16KB	GPIO3	FF550000	32KB
PMU_SGRF	FF080000	64KB	GPIO3_IOC	FF558000	32KB
RESERVED	FF090000	448KB	GPIO4	FF560000	32KB
INTERCONNECT	FF100000	448KB	GPIO4_IOC	FF568000	32KB
RESERVED	FF170000	320KB	RESERVED	FF570000	64KB
RTC	FF1C0000	64KB	TIMER_6CH	FF580000	64KB
RESERVED	FF1D0000	128KB	STIMER_2CH	FF590000	64KB
GIC400	FF1F0000	64KB	WDTNS	FF5A0000	64KB
DEBUG	FF200000	256KB	WDTS	FF5B0000	64KB
CORE_PVTM	FF240000	64KB	MAILBOX	FF5C0000	64KB
RESERVED	FF250000	640KB	INTMUX	FF5D0000	64KB
TIMER_HP	FF2F0000	64KB	RESERVED	FF5E0000	384KB
PMU	FF300000	64KB	MCU CACHE	FF640000	64KB
I2C0	FF310000	64KB	MCU CACHE RAM TEST	FF650000	64KB
I2C1	FF320000	64KB	NPU	FF660000	64KB
RESERVED	FF330000	64KB	PMU_SRAM	FF670000	64KB
DSM	FF340000	64KB	NPU_CBUF	FF680000	256KB
PWM0	FF350000	64KB	SYSTEM SRAM	FF6C0000	256KB
PWM1	FF360000	64KB	SYSTEM_SRAM_EXT	FF700000	1MB
PMU_WDT	FF370000	32KB	DDR CONTROLLER	FF800000	64KB
PMU_MAILBOX	FF378000	32KB	DDR_MONITOR	FF810000	64KB

Module	Start Address	Size	Module	Start Address	Size
GPIO0	FF380000	32KB	DDR_PHY	FF820000	64KB
GPIO0_IOC	FF388000	32KB	DDR_DFICTRL	FF830000	32KB
PMU_PVTM	FF390000	64KB	DDR_HWLP	FF838000	32KB
PMU_CRU	FF3A0000	64KB	RESERVED	FF840000	768KB
TOP_CRU	FF3B0000	8KB	DDR SECURE	FF900000	64KB
PERI_CRU	FF3B2000	8KB	SHRM SECURE	FF910000	64KB
VI_CRU	FF3B4000	8KB	RESERVED	FF920000	384KB
NPU_CRU	FF3B6000	8KB	RGA	FF980000	64KB
CORE_CRU	FF3B8000	8KB	VOP_LITE	FF990000	64KB
RKVENC_CRU	FF3BA000	8KB	SDIO	FF9A0000	64KB
VO_CRU	FF3BC000	8KB	RESERVED	FF9B0000	320KB
DDR_CRU	FF3BE000	4KB	ISP	FFA00000	64KB
DDR_SUBCRU	FF3BF000	4KB	VICAP	FFA10000	64KB
SARADC_CON	FF3C0000	32KB	CSI2HOST0	FFA20000	64KB
TSADC_CON	FF3C8000	32KB	CSI2HOST1	FFA30000	64KB
OTP_NS_CTRL	FF3D0000	32KB	RESERVED	FFA40000	64KB
OTP_S_CTRL	FF3D8000	32KB	RKVENC	FFA50000	64KB
OTG_APB	FF3E0000	32KB	VEPU_PP	FFA60000	64KB
CSI_PHY	FF3E8000	32KB	DVBM	FFA70000	64KB
RESERVED	FF3F0000	32KB	MAC	FFA80000	64KB
MAC_PHY	FF3F8000	32KB	EMMC	FFA90000	64KB
RESERVED	FF400000	128KB	SDMMC	FFAA0000	64KB
DMAC	FF420000	64KB	RESERVED	FFAB0000	64KB
SDMAC	FF430000	64KB	FSPI	FFAC0000	64KB
CRYPTO	FF440000	32KB	RKIVE	FFAD0000	64KB
TRNG_NS	FF448000	16KB	I2S0_8CH	FFAE0000	32KB
TRNG_SEC	FF44C000	16KB	RESERVED	FFAE8000	32KB
I2C2	FF450000	64KB	RESERVED	FFAF0000	64KB
I2C3	FF460000	64KB	USB2	FFB00000	1MB
I2C4	FF470000	64KB	BOOTROM	FFFF0000	64KB

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

Table 1-2 Address Remapping

remap[1:0]=2'b00		remap[1:0]=2'b01		remap[1:0]=2'b10	
		not accessible	BootRom(20KB)	not accessible	BootRom(20KB)
0xFFFF0000	BootRom(20KB)	0xFFFF0000	PMU_SRAM(8KB)	0xFFFF0000	SYSTEM_SRAM (64KB)
0xFF670000	PMU_SRAM(8KB)	0xFF670000	PMU_SRAM(8KB)	0xFF670000	PMU_SRAM(8KB)
0xFF6C0000	SYSTEM_SRAM (256KB)	0xFF6C0000	SYSTEM_SRAM (256KB)	0xFF6C0000	SYSTEM_SRAM (256KB)

1.2 System Boot

RV1106 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 OTG or UART interface. 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 BootRom in advance.
The following features are supported.

- Support system boot from the following device:
  - Serial NOR Flash, 1bit or 4bits data width (device layout in FSPI IO)
  - Serial NAND Flash, 1bit data width (device layout in FSPI IO)
  - eMMC Interface, 4bits or 8bits data width
  - SDMMC Card, 4bits data width
- Support system code download by USB OTG or UART

1.3 System Interrupt Connection

RV1106 provides a general interrupt controller (GIC) for CPU, which has 128 SPI (shared peripheral interrupts) interrupt sources. The triggered type for each SPI interrupt is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table.

Table 1-3 RV1106 Interrupt Connection List

Number	Source	Polarity	Number	Source	Polarity
0-31 PPI		Low level	97	dma_int2	High level
32	cache_irq	High level	98	dma_int3	High level
33	|mailbox_irq_ap	High level	99	dma_int4	High level
34	|mailbox_irq_bb	High level	100	dma_int5	High level
35	core_pvtm_int	High level	101	dma_int6	High level
36	pmu_pvtm_int	High level	102	dma_int7	High level
37	gpio0_int	High level	103	dma_int8	High level
38	gpio0_ext_int	High level	104	dma_int9	High level
39	gpio1_int	High level	105	dma_int10	High level
40	gpio1_ext_int	High level	106	dma_int11	High level
41	gpio2_int	High level	107	dma_int12	High level
42	gpio2_ext_int	High level	108	dma_int13	High level
43	gpio3_int	High level	109	dma_int14	High level
44	gpio3_ext_int	High level	110	dma_int15	High level
45	gpio4_int	High level	111	dma_abort_int	High level
46	gpio4_ext_int	High level	112	i2s0_8ch_int	High level
47	crypto_int	High level	113	msch_alarm_irq	High level
48	trng_s_int	High level	114	ddrmon_int	High level
49	trng_ns_int	High level	115	upctl_alert_err_int	High level
50	i2c0_int	High level	116	upctl_awpoison_int	High level
51	i2c1_int	High level	117	upctl_arpoison_int	High level
52	i2c2_int	High level	118	rknpor_powergood	High level
53	i2c3_int	High level	119	rga_intr	High level
54	i2c4_int	High level	120	vop_intr_post_lb	High level
55	spi0_int	High level	121	vop_intr	High level
56	spi1_int	High level	122	otpc_ns_int	High level
57	uart0_int	High level	123	otpc_s_int	High level

Number	Source	Polarity	Number	Source	Polarity
58	uart1_int	High level	124	otp_mask_int	High level
59	uart2_int	High level	125	irq_gmac_sbd_intr_o	High level
60	uart3_int	High level	126	irq_gmac_sbd_perch_tx_intr_o	High level
61	uart4_int	High level	127	irq_gmac_sbd_perch_rx_intr_o	High level
62	uart5_int	High level	128	irq_gmac_pmt_intr_o	High level
63	pwm0_int	High level	129	tsadc_int	High level
64	pwm0_int_pwr	High level	130	rtc_int	High level
65	pwm1_int	High level	131	csi2host0_intr1	High level
66	pwm1_int_pwr	High level	132	csi2host0_intr2	High level
67	pwm2_int	High level	133	csi2host1_intr1	High level
68	pwm2_int_pwr	High level	134	csi2host1_intr2	High level
69	hptimer_int	High level	135	mipi_irq	High level
70	timer0_int	High level	136	mi_irq	High level
71	timer1_int	High level	137	isp_irq	High level
72	timer2_int	High level	138	vicap_int	High level
73	timer3_int	High level	139	o_enc_int	High level
74	timer4_int	High level	140	vepu_pp_int	High level
75	timer5_int	High level	141	rknn_int	High level
76	stimer0_int	High level	142	NOT USED	High level
77	stimer1_int	High level	143	sw_ive_irq	High level
78	wdt_ns_int	High level	144	decom_int	High level
79	wdt_s_int	High level	145	pmuwdt_int	High level
80	emmc_int	High level	146	|pmu_mailbox_irq_ap	High level
81	sdio_int	High level	147	|pmu_mailbox_irq_bb	High level
82	sdmmc_dectn_in_flt	High level	148	dvbm_int	High level
83	sdmmc_detectn_irq	High level	149	irq_macphy	High level
84	sdmmc_int	High level	150	irq_mgp_macphy	High level
85	sfc_int	High level	151	irq_excmgp_macphy	High level
86	usbotg_int	High level	152	NOT USED	High level
87	usbotg_host_sys_err	High level	153	NOT USED	High level
88	usbotg_pme_generation	High level	154	NOT USED	High level
89	usbotg_host_legacy_smi_interrupt	High level	155	NOT USED	High level
90	otg_bvalid_irq	High level	156	NOT USED	High level
91	otg_id_irq	High level	157	NOT USED	High level
92	otg_linestate_irq	High level	158	naxierrirq	High level
93	otg_disconnect_irq	High level	159	npmuirq	High level
94	saradc_int	High level	160		High level
95	dma_int0	High level	97		High level
96	dma_int1	High level	98		High level

1.4 System DMA Hardware Request Connection

RV1106 provides three DMA controller (DMAC) inside the system, the following tables are the DMA hardware request list.

Table 1-4 RV1106 DMAC Hardware Request Connection List

Req Number	Source	Polarity	Req Number	Source	Polarity
0	SPI0_RX	High level	16	UART5_RX	High level
1	SPI0_TX	High level	17	UART5_TX	High level
2	SPI1_RX	High level	18	PWM0	High level
3	SPI1_TX	High level	19	PWM1	High level
4	NOT USED	High level	20	PWM2	High level
5	NOT USED	High level	21	I2S0_8CH_RX	High level
6	UART0_RX	High level	22	I2S0_8CH_TX	High level
7	UART0_TX	High level	23	NOT USED	High level
8	UART1_RX	High level	24	NOT USED	High level
9	UART1_TX	High level	25	NOT USED	High level
10	UART2_RX	High level	26	NOT USED	High level
11	UART2_TX	High level	27	NOT USED	High level
12	UART3_RX	High level	28	NOT USED	High level
13	UART3_TX	High level	29	NOT USED	High level
14	UART4_RX	High level	30	NOT USED	High level
15	UART4_TX	High level	31	NOT USED	High level

Chapter 2 Clock & Reset Unit (CRU)

2.1 Overview

CRU is an APB slave module that is designed for generating all kinds of internal and system clocks and resets in the 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, software reset or temperature sensor.
CRU is located at several addresses.

Table 2-1 CRU Base Address

	Description	Base Address
PMUCRU_BASE	Used for PMUCRU.	0XFF3A0000
CRU_BASE	Used for TOPCRU.	0XFF3B0000
PERICRU_BASE	Used for PERICRU.	0XFF3B2000
VICRU_BASE	Used for VICRU.	0XFF3B4000
NPUCRU_BASE	Used for NPUCRU.	0XFF3B6000
CORECRU_BASE	Used for CORECRU.	0XFF3B8000
VEPUCRU_BASE	Used for VEPUCRU.	0XFF3BA000
VOCRU_BASE	Used for VOCRU.	0XFF3BC000
DDRCRU_BASE	Used for DDRCRU.	0XFF3BE000
SUBDDR_BASE	Used for SUBDDRCRU.	0XFF3BF000

The CRU supports the following features:

- Compliance with AMBA APB interface
- Embedded with 2 fractional PLLs (GPLL and DPLL) and 2 integer PLLs (APLL and CPLL)
- Flexible selection of clock source
- Use clock matrix scheme
- Support dividing clock separately
- Support gating clock separately
- Support software reset each module separately

2.2 Block Diagram

The CRU comprises with:

- PLL
- Register configuration unit
- Clock generate unit
- Reset generate unit

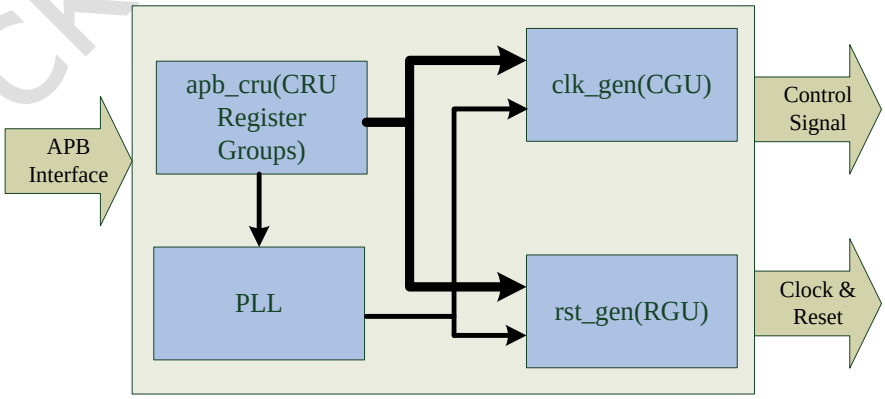


Fig. 2-1 CRU Block Diagram

2.3 Function Description

2.3.1 System Clock Solution

There are 2 fractional PLLs in RV1106: DPLL and GPLL.There are also 2 integer PLLs: APLL,

and CPLL. Each PLL can only receive 24MHz oscillator as input reference clock and can be set to three work modes: normal mode, slow mode and deep slow mode.Before power on or changing PLL setting, we must program PLL into slow mode or deep slow mode.
To maximize the flexibility, some clocks can select divider source from multiple PLLs.To provide some specific frequency, another solution is integrated: fractional divider. Divfree50 divider and divfree NP5 divider are also provided for some modules. All clocks can be gated by software.
The basic units for clock generation are:

- Gating
- MUX (multiplexer)
- Divfree(Glitch free divider)
  - clk_out_freq= clk_in_freq/(divcon + 1)
  - When divcon is even, the clock duty cycle of clk_out is 50%
  - When divcon is odd, the clock duty cycle of clk_out is not 50%
- Fracdiv(Fractional divider)
  - clk_out_freq= clk_in_freq*numerator/denominator, both numerator and denominator are 16 bits
- Divfree50(Glitch free divider for duty cycle 50%)
  - clk_out_freq= clk_in_freq/(divcon +1)
  - When divcon is even or odd, the clock duty cycle of clk_out is 50%
- DivFreeNP5(Glitch free divider for null point 5)
  - clk_out_freq= clk_in_freq/(div_con+1.5)
  - The clock duty cycle of clk_out is not 50%

The settings of all basic units are controlled by CRU registers.

2.3.2 System Reset Solution

Almost all modules have these reset source as the following figure shows. The 'xxx' in the figure is the module name.

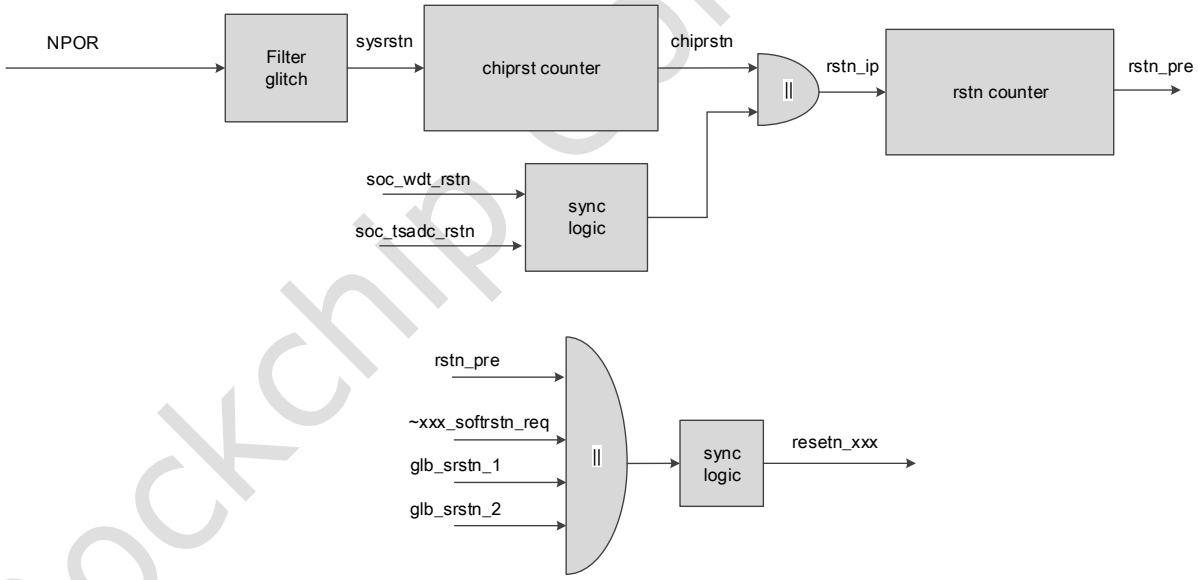


Fig. 2-2 Reset Architecture Diagram

Reset source of each reset signal includes:

- NPOR: External power on reset
- soc_wdt_rstn: Reset from WDT module
- soc_tsadc_rstn: Reset from TSADC module
- softrstn_req: Software reset request by programming CRU_SOFTRST_CON
- glb_srstn_1: First global software reset by programming CRU_GLB_SRST_FST as 0xfdb9
- glb_srstn_2: Second global software reset by programming CRU_GLB_SRST_SND as 0xeca8

2.3.3 PLL Introduction

PLL is phase locked loop (PLL) with a wide-output-frequency-range for frequency synthesis.

Table 2-2 RV1106 PLL Introduction

	FRACPLL(INT Mode)		INTPLL		FRACPLL(FRAC Mode)	
Parameter	Min	Max	Min	Max	Min	Max
Fref(MHz)	24		24		24	
Refdiv	1	63	1	63	1	63
Fbdiv	16	3800	16	190	20	380
Postdiv	1	49	1	49	1	49
PFD(MHz)	1	Fvco/16	10	Fvco/16	1	Fvco/20
Fvco(MHz)	950	3800	475	1900	950	3800
Fout(MHz)	19	3800	9	1900	19	3800
lock time		500		1500		500
Fvco	Fvco=Fref*Fbdiv /Refdiv		Fvco=Fref*Fbdiv /Refdiv		Fvco=Fref*(Fbdiv+Ffrac /2^24)/Refdiv	
Fout	Fout=Fvco/(postdiv1*postdiv2)		Fout=Fvco/(postdiv1*postdiv2)		Fout=Fvco/(postdiv1*postdiv2)	

2.3.3.1 FRACPLL Introduction

The fractional PLLs inside RV1106 output clock's frequency up to 3.8GHz. The PLL is a general purpose, high-performance PLL-based clock generator. Also, 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.

The PLL supports the following features:

- Input frequency range:1MHz to 1200MHz for integer mode and 10MHz to 1200MHz for fractional mode
- PFD minimum reference frequency range: 1MHz for integer mode and 10MHz for fractional mode
- Output frequency range: 19MHz to 3.8GHz
- VCO output frequency range: 950MHz to 3.8GHz
- 24bit 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) ensures excellent supply rejection in noisy SoC applications
- Lock detect signal indicates when frequency lock has been achieved

FRACpll block diagram is shown below.

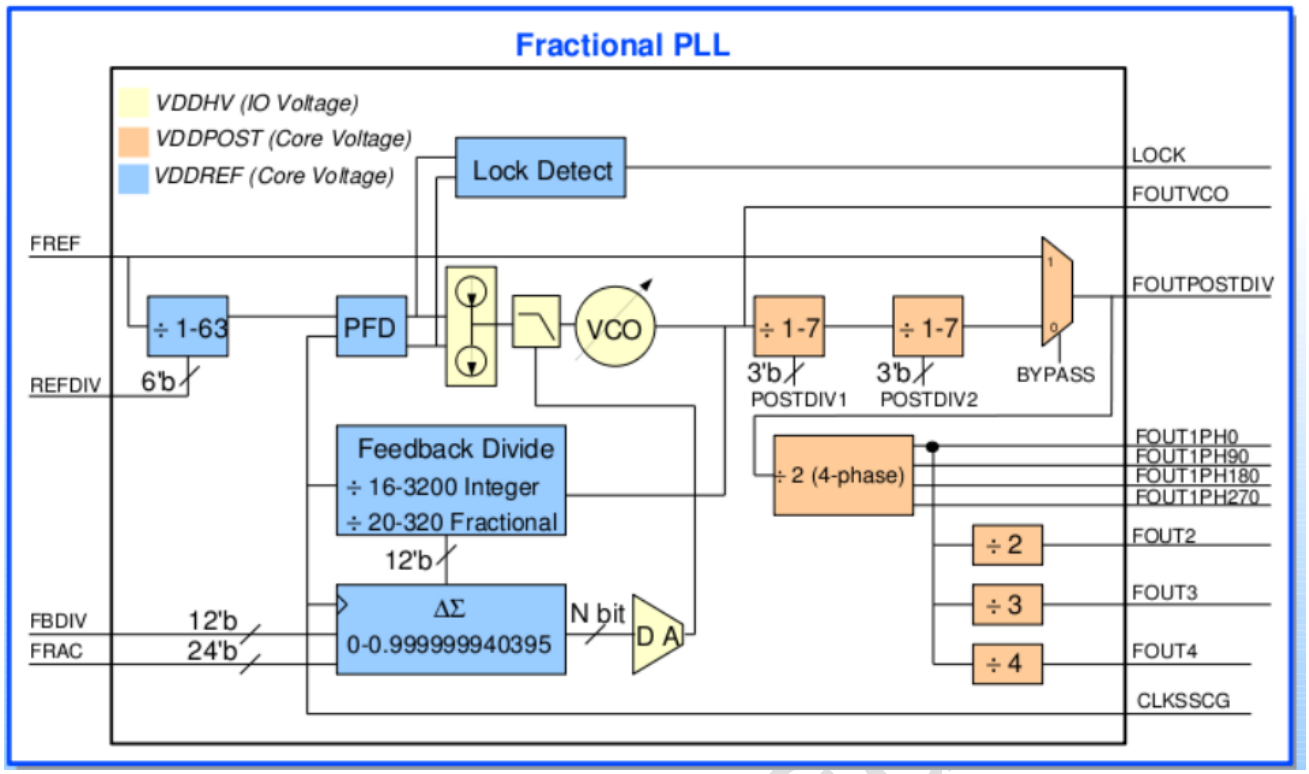


Fig. 2-3 FRACPLL Block Diagram

2.3.3.2 INTPLL Introduction

The integer PLLs inside RV1106 output clock's frequency up to 1.9 GHz. The PLL is a multi-function, general purpose frequency synthesizer optimized for low power digital clocking. Wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for a variety of different application. With excellent supply noise immunity, the PLL is ideal for use in noisy SoC environments.

The PLL supports the following features:

- Input frequency range: 10MHz to 800MHz
- PFD minimum reference frequency range: 10MHz
- Output frequency range: 9MHz to 1.9GHz
- VCO output frequency range: 475 MHz to 1.9GHz
- 4:1 VCO frequency range allows PLL to be optimized for minimum jitter or minimum power
- Low period jitter provides maximum timing margin in high frequency designs
- Lock detect signal indicates when frequency lock has been achieved

INTPLL block diagram is shown below.

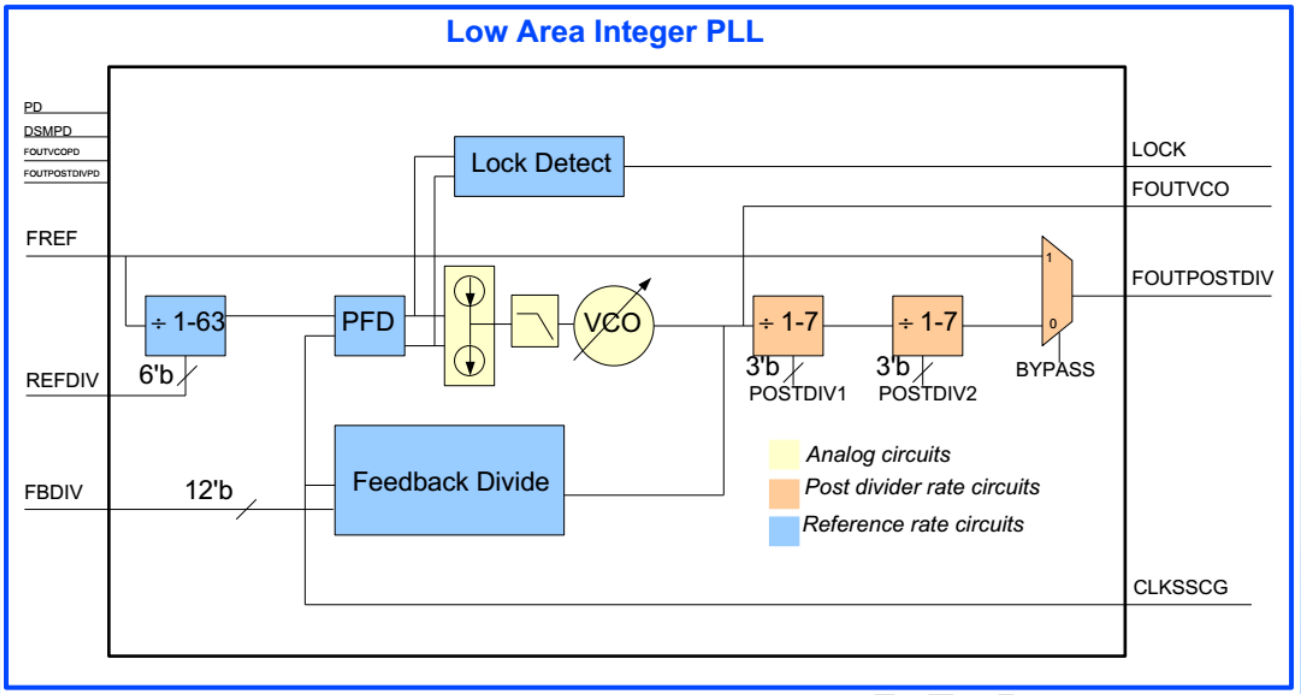


Fig. 2-4 INTPLL Block Diagram

2.4 CRU Register Description

2.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x0000105B	APLL configuration register0
CRU_APLL_CON1	0x0004	W	0x00001042	APLL configuration register1
CRU_APLL_CON2	0x0008	W	0x00000000	APLL configuration register2
CRU_APLL_CON3	0x000C	W	0x00000007	APLL configuration register3
CRU_APLL_CON4	0x0010	W	0x00007F00	APLL configuration register4
CRU_CPLL_CON0	0x0020	W	0x00001053	CPLL configuration register0
CRU_CPLL_CON1	0x0024	W	0x00001042	CPLL configuration register1
CRU_CPLL_CON2	0x0028	W	0x00000000	CPLL configuration register2
CRU_CPLL_CON3	0x002C	W	0x00000007	CPLL configuration register3
CRU_CPLL_CON4	0x0030	W	0x00007F00	CPLL configuration register4
CRU_DPLL_CON0	0x0040	W	0x0000312C	DPLL configuration register0
CRU_DPLL_CON1	0x0044	W	0x00001083	DPLL configuration register1
CRU_DPLL_CON2	0x0048	W	0x00000000	DPLL configuration register2
CRU_DPLL_CON3	0x004C	W	0x00000007	DPLL configuration register3
CRU_DPLL_CON4	0x0050	W	0x00007F00	DPLL configuration register4
CRU_GPLL_CON0	0x0060	W	0x00001063	GPLL configuration register0
CRU_GPLL_CON1	0x0064	W	0x00001042	GPLL configuration register1
CRU_GPLL_CON2	0x0068	W	0x00000000	GPLL configuration register2
CRU_GPLL_CON3	0x006C	W	0x00000007	GPLL configuration register3
CRU_GPLL_CON4	0x0070	W	0x00007F00	GPLL configuration register4
CRU_SSGTBL0_3	0x0140	W	0x00000000	External wave table register0
CRU_SSGTBL4_7	0x0144	W	0x00000000	External wave table register1
CRU_SSGTBL8_11	0x0148	W	0x00000000	External wave table register2

Name	Offset	Size	Reset Value	Description
CRU_SSGTBL12_15	0x014C	W	0x00000000	External wave table register3
CRU_SSGTBL16_19	0x0150	W	0x00000000	External wave table register4
CRU_SSGTBL20_23	0x0154	W	0x00000000	External wave table register5
CRU_SSGTBL24_27	0x0158	W	0x00000000	External wave table register6
CRU_SSGTBL28_31	0x015C	W	0x00000000	External wave table register7
CRU_SSGTBL32_35	0x0160	W	0x00000000	External wave table register8
CRU_SSGTBL36_39	0x0164	W	0x00000000	External wave table register9
CRU_SSGTBL40_43	0x0168	W	0x00000000	External wave table register10
CRU_SSGTBL44_47	0x016C	W	0x00000000	External wave table register11
CRU_SSGTBL48_51	0x0170	W	0x00000000	External wave table register12
CRU_SSGTBL52_55	0x0174	W	0x00000000	External wave table register13
CRU_SSGTBL56_59	0x0178	W	0x00000000	External wave table register14
CRU_SSGTBL60_63	0x017C	W	0x00000000	External wave table register15
CRU_SSGTBL64_67	0x0180	W	0x00000000	External wave table register16
CRU_SSGTBL68_71	0x0184	W	0x00000000	External wave table register17
CRU_SSGTBL72_75	0x0188	W	0x00000000	External wave table register18
CRU_SSGTBL76_79	0x018C	W	0x00000000	External wave table register19
CRU_SSGTBL80_83	0x0190	W	0x00000000	External wave table register20
CRU_SSGTBL84_87	0x0194	W	0x00000000	External wave table register21
CRU_SSGTBL88_91	0x0198	W	0x00000000	External wave table register22
CRU_SSGTBL92_95	0x019C	W	0x00000000	External wave table register23
CRU_SSGTBL96_99	0x01A0	W	0x00000000	External wave table register24
CRU_SSGTBL100_103	0x01A4	W	0x00000000	External wave table register25
CRU_SSGTBL104_107	0x01A8	W	0x00000000	External wave table register26
CRU_SSGTBL108_111	0x01AC	W	0x00000000	External wave table register27
CRU_SSGTBL112_115	0x01B0	W	0x00000000	External wave table register28
CRU_SSGTBL116_119	0x01B4	W	0x00000000	External wave table register29
CRU_SSGTBL120_123	0x01B8	W	0x00000000	External wave table register30
CRU_SSGTBL124_127	0x01BC	W	0x00000000	External wave table register31
CRU_MODE_CON00	0x0280	W	0x00000000	Internal clock select and division register 0
CRU_CLKSEL_CON00	0x0300	W	0x00000A73	Internal clock select and division register 0
CRU_CLKSEL_CON01	0x0304	W	0x00000147	Internal clock select and division register 1
CRU_CLKSEL_CON02	0x0308	W	0x000000E3	Internal clock select and division register 2
CRU_CLKSEL_CON03	0x030C	W	0x00000082	Internal clock select and division register 3
CRU_CLKSEL_CON04	0x0310	W	0x00000841	Internal clock select and division register 4
CRU_CLKSEL_CON05	0x0314	W	0x00000001	Internal clock select and division register 5

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON06	0x0318	W	0x001403DE	Internal clock select and division register 6
CRU_CLKSEL_CON07	0x031C	W	0x00000006	Internal clock select and division register 7
CRU_CLKSEL_CON08	0x0320	W	0x001403DE	Internal clock select and division register 8
CRU_CLKSEL_CON09	0x0324	W	0x00000006	Internal clock select and division register 9
CRU_CLKSEL_CON10	0x0328	W	0x001403DE	Internal clock select and division register 10
CRU_CLKSEL_CON11	0x032C	W	0x00000006	Internal clock select and division register 11
CRU_CLKSEL_CON12	0x0330	W	0x001403DE	Internal clock select and division register 12
CRU_CLKSEL_CON13	0x0334	W	0x00000006	Internal clock select and division register 13
CRU_CLKSEL_CON14	0x0338	W	0x001403DE	Internal clock select and division register 14
CRU_CLKSEL_CON15	0x033C	W	0x00000006	Internal clock select and division register 15
CRU_CLKSEL_CON16	0x0340	W	0x001403DE	Internal clock select and division register 16
CRU_CLKSEL_CON17	0x0344	W	0x00000006	Internal clock select and division register 17
CRU_CLKSEL_CON18	0x0348	W	0x03355460	Internal clock select and division register 18
CRU_CLKSEL_CON19	0x034C	W	0x00000007	Internal clock select and division register 19
CRU_CLKSEL_CON20	0x0350	W	0x03355460	Internal clock select and division register 20
CRU_CLKSEL_CON21	0x0354	W	0x0000000B	Internal clock select and division register 21
CRU_CLKSEL_CON23	0x035C	W	0x00000038	Internal clock select and division register 23
CRU_CLKSEL_CON24	0x0360	W	0x00000580	Internal clock select and division register 24
CRU_CLKSEL_CON25	0x0364	W	0x00000004	Internal clock select and division register 25
CRU_CLKSEL_CON26	0x0368	W	0x001403DE	Internal clock select and division register 26
CRU_CLKSEL_CON27	0x036C	W	0x00000006	Internal clock select and division register 27
CRU_CLKSEL_CON28	0x0370	W	0x001403DE	Internal clock select and division register 28

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON29	0x0374	W	0x00000006	Internal clock select and division register 29
CRU_CLKSEL_CON30	0x0378	W	0x001403DE	Internal clock select and division register 30
CRU_CLKSEL_CON31	0x037C	W	0x00000006	Internal clock select and division register 31
CRU_CLKSEL_CON32	0x0380	W	0x001403DE	Internal clock select and division register 32
CRU_CLKSEL_CON33	0x0384	W	0x00000002	Internal clock select and division register 33
CRU_GATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
CRU_GATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
CRU_GATE_CON02	0x0808	W	0x0000E000	Internal clock gate and division register 2
CRU_GATE_CON03	0x080C	W	0x00000001	Internal clock gate and division register 3
CRU_SOFTRST_CON02	0x0A08	W	0x00000000	Internal clock reset register 2
CRU_GLB_CNT_TH	0x0C00	W	0x00640064	GLB_CNT_TH
CRU_GLB_RST_ST	0x0C04	W	0x00000000	GLB_RST_ST
CRU_GLB_SRST_FST	0x0C08	W	0x00000000	GLB_SRST_FST
CRU_GLB_SRST_SND	0x0C0C	W	0x00000000	GLB_SRST_SND
CRU_GLB_RST_CON	0x0C10	W	0x00000000	GLB_RST_CON

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.4.2 Detail Registers Description

CRU_APLL_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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) shoule obey the rule : postdiv1 > = postdiv2

Bit	Attr	Reset Value	Description
11:0	RW	0x05b	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_APLL_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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	RW	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	0x02	refdiv Reference Clock Divide Value, (1-63) APLL is int pll. refdiv must use 1 or 2

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
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	0x000000	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	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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
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

Bit	Attr	Reset Value	Description
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_APLL_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x00	reserved
0	RW	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 (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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) shoule obey the rule : postdiv1 > = postdiv2
11:0	RW	0x053	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_CPLL_CON1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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	RW	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	0x02	refdiv Reference Clock Divide Value, (1-63) APLL is int pll. refdiv must use 1 or 2

CRU_CPLL_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

Bit	Attr	Reset Value	Description
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	0x000000	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_CPLL_CON3

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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
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_CPLL_CON4

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs (0-255)
7:1	RO	0x00	reserved
0	RW	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 (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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) shoule obey the rule : postdiv1 > = postdiv2
11:0	RW	0x12c	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_DPLL_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

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	RW	0x0	pll_lock PLL lock status 1'b0: Unlock 1'b1: Lock
9	RO	0x0	reserved
8:6	RW	0x2	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 (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

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	0x000000	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_DPLL_CON3

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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
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_DPLL_CON4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs, (0-255)
7:1	RO	0x00	reserved
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: No select ext_wave 1'b1: Select ext_wave

CRU_GPLL_CON0

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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) shoule obey the rule : postdiv1 > = postdiv2
11:0	RW	0x063	fbdiv Feedback Divide Value, valid divider settings are: [16, 2500] in integer mode [20, 500] in fractional mode Tips: No plus one operation

CRU_GPLL_CON1

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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	RW	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	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_GPLL_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

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_GPLL_CON3

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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
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_GPLL_CON4

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x7f	ssmod_ext_maxaddr External wave table data inputs, (0-255)
7:1	RO	0x00	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ssmod_sel_ext_wave select external wave 1'b0: No select ext_wave 1'b1: Select ext_wave

CRU_SSGTBL0_3

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0144)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0148)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x014C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0150)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0154)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0158)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x015C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0164)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0168)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x016C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0170)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0174)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0178)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x017C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x018C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0190)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0194)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x0198)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x019C)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01A0)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01A4)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01A8)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01AC)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01B0)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01B4)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01B8)

Bit	Attr	Reset Value	Description
31:0	RW	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 (0x01BC)

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

CRU_MODE_CON00

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	clk_gpll_mode clk_gpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_gpll 2'b10: clk_deepslow
3:2	RW	0x0	clk_cpll_mode clk_cpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_cpll 2'b10: clk_deepslow
1:0	RW	0x0	clk_apll_mode clk_apll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_apll 2'b10: clk_deepslow

CRU_CLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x1	clk_matrix_100m_src_sel clk_matrix_100m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
10:6	RW	0x09	clk_matrix_100m_src_div Divide clk_matrix_100m_src by (div_con + 1).
5	RW	0x1	clk_matrix_50m_src_sel clk_matrix_50m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x13	clk_matrix_50m_src_div Divide clk_matrix_50m_src by (div_con + 1).

CRU_CLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	clk_matrix_200m_src_sel clk_matrix_200m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
10:6	RW	0x05	clk_matrix_200m_src_div Divide clk_matrix_200m_src by (div_con + 1).
5	RW	0x0	clk_matrix_150m_src_sel clk_matrix_150m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x07	clk_matrix_150m_src_div Divide clk_matrix_150m_src by (div_con + 1).

CRU_CLKSEL_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_matrix_300m_src_sel clk_matrix_300m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
10:6	RW	0x03	clk_matrix_300m_src_div Divide clk_matrix_300m_src by (div_con + 1).
5	RW	0x1	clk_matrix_250m_src_sel clk_matrix_250m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x03	clk_matrix_250m_src_div Divide clk_matrix_250m_src by (div_con + 1).

CRU_CLKSEL_CON03

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	clk_matrix_400m_src_sel clk_matrix_400m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
10:6	RW	0x02	clk_matrix_400m_src_div DT50 division register. Divide clk_matrix_400m_src by (div_con + 1).
5	RW	0x0	clk_matrix_339m_src_sel clk_matrix_339m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x02	clk_matrix_339m_src_div NP5 division register. Divide clk_matrix_339m_src by ((2 * div_con + 3) / 2).

CRU_CLKSEL_CON04

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x1	clk_matrix_500m_src_sel clk_matrix_500m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
10:6	RW	0x01	clk_matrix_500m_src_div Divide clk_matrix_500m_src by (div_con + 1).
5	RW	0x0	clk_matrix_450m_src_sel clk_matrix_450m_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x01	clk_matrix_450m_src_div NP5 division register. Divide clk_matrix_450m_src by ((2 * div_con + 3) / 2).

CRU_CLKSEL_CON05

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	clk_uart0_src_sel clk_uart0_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
4:0	RW	0x01	clk_uart0_src_div Divide clk_uart0_src by (div_con + 1).

CRU_CLKSEL_CON06

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart0_frac_div clk_uart0_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON07

Address: Operational Base + offset (0x031C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_uart1_src_sel clk_uart1_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_uart1_src_div Divide clk_uart1_src by (div_con + 1).
1:0	RW	0x2	sclk_uart0_src_sel sclk_uart0_src clock mux. 2'b00: clk_uart0_src 2'b01: clk_uart0_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON08

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart1_frac_div clk_uart1_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON09

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_uart2_src_sel clk_uart2_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_uart2_src_div Divide clk_uart2_src by (div_con + 1).
1:0	RW	0x2	sclk_uart1_src_sel sclk_uart1_src clock mux. 2'b00: clk_uart1_src 2'b01: clk_uart1_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart2_frac_div clk_uart2_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON11

Address: Operational Base + offset (0x032C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_uart3_src_sel clk_uart3_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_uart3_src_div Divide clk_uart3_src by (div_con + 1).
1:0	RW	0x2	sclk_uart2_src_sel sclk_uart2_src clock mux. 2'b00: clk_uart2_src 2'b01: clk_uart2_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON12

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart3_frac_div clk_uart3_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON13

Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_uart4_src_sel clk_uart4_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_uart4_src_div Divide clk_uart4_src by (div_con + 1).
1:0	RW	0x2	sclk_uart3_src_sel sclk_uart3_src clock mux. 2'b00: clk_uart3_src 2'b01: clk_uart3_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON14

Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart4_frac_div clk_uart4_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON15

Address: Operational Base + offset (0x033C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_uart5_src_sel clk_uart5_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_uart5_src_div Divide clk_uart5_src by (div_con + 1).
1:0	RW	0x2	sclk_uart4_src_sel sclk_uart4_src clock mux. 2'b00: clk_uart4_src 2'b01: clk_uart4_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON16

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_uart5_frac_div clk_uart5_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON17

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_i2s0_8ch_tx_src_sel clk_i2s0_8ch_tx_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_i2s0_8ch_tx_src_div Divide clk_i2s0_8ch_tx_src by (div_con + 1).
1:0	RW	0x2	sclk_uart5_src_sel sclk_uart5_src clock mux. 2'b00: clk_uart5_src 2'b01: clk_uart5_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON18

Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:0	RW	0x03355460	clk_i2s0_8ch_tx_frac_div clk_i2s0_8ch_tx_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON19

Address: Operational Base + offset (0x034C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_i2s0_8ch_rx_src_sel clk_i2s0_8ch_rx_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_i2s0_8ch_rx_src_div Divide clk_i2s0_8ch_rx_src by (div_con + 1).
1:0	RW	0x3	mclk_i2s0_8ch_tx_src_peri_sel mclk_i2s0_8ch_tx_src_peri clock mux. 2'b00: clk_i2s0_8ch_tx_src 2'b01: clk_i2s0_8ch_tx_frac 2'b10: i2s0_mclkin 2'b11: xin_osc0_half

CRU_CLKSEL_CON20

Address: Operational Base + offset (0x0350)

Bit	Attr	Reset Value	Description
31:0	RW	0x03355460	clk_i2s0_8ch_rx_frac_div clk_i2s0_8ch_rx_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON21

Address: Operational Base + offset (0x0354)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x2	i2s0_8ch_mclkout_sel i2s0_8ch_mclkout clock mux. 2'b00: mclk_i2s0_8ch_tx_src_peri 2'b01: mclk_i2s0_8ch_rx_src_peri 2'b10: xin_osc0_half
1:0	RW	0x3	mclk_i2s0_8ch_rx_src_peri_sel mclk_i2s0_8ch_rx_src_peri clock mux. 2'b00: clk_i2s0_8ch_rx_src 2'b01: clk_i2s0_8ch_rx_frac 2'b10: i2s0_mclkin 2'b11: xin_osc0_half

CRU_CLKSEL_CON23

Address: Operational Base + offset (0x035C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:9	RW	0x00	clk_ref_pvtpll_0_div Divide clk_ref_pvtpll_0 by (div_con + 1).
8	RW	0x0	dclk_vop_src_sel dclk_vop_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
7:3	RW	0x07	dclk_vop_src_div Divide dclk_vop_src by (div_con + 1).
2:0	RO	0x0	reserved

CRU_CLKSEL_CON24

Address: Operational Base + offset (0x0360)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x0	clk_testout_top2vepu_sel clk_testout_top2vepu clock mux. 2'b00: clk_gpll_mux 2'b01: clk_cpll_mux 2'b10: pclk_top_root
11:7	RW	0x0b	clk_testout_top2vepu_div Divide clk_testout_top2vepu by (div_con + 1).

Bit	Attr	Reset Value	Description
6:5	RW	0x0	pclk_top_root_sel pclk_top_root clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func
4:0	RW	0x00	clk_ref_pvtpll_1_div Divide clk_ref_pvtpll_1 by (div_con + 1).

CRU_CLKSEL_CON25

Address: Operational Base + offset (0x0364)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_ref_mipi0_src_sel clk_ref_mipi0_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_ref_mipi0_src_div Divide clk_ref_mipi0_src by (div_con + 1).
1:0	RO	0x0	reserved

CRU_CLKSEL_CON26

Address: Operational Base + offset (0x0368)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_ref_mipi0_frac_div clk_ref_mipi0_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON27

Address: Operational Base + offset (0x036C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_ref_mipi1_src_sel clk_ref_mipi1_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux

Bit	Attr	Reset Value	Description
6:2	RW	0x01	clk_ref_mipi1_src_div Divide clk_ref_mipi1_src by (div_con + 1).
1:0	RW	0x2	clk_ref_mipi0_out_sel clk_ref_mipi0_out clock mux. 2'b00: clk_ref_mipi0_src 2'b01: clk_ref_mipi0_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON28

Address: Operational Base + offset (0x0370)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_ref_mipi1_frac_div clk_ref_mipi1_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON29

Address: Operational Base + offset (0x0374)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_vicap_m0_src_sel clk_vicap_m0_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_vicap_m0_src_div Divide clk_vicap_m0_src by (div_con + 1).
1:0	RW	0x2	clk_ref_mipi1_out_sel clk_ref_mipi1_out clock mux. 2'b00: clk_ref_mipi1_src 2'b01: clk_ref_mipi1_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON30

Address: Operational Base + offset (0x0378)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_vicap_m0_frac_div clk_vicap_m0_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON31

Address: Operational Base + offset (0x037C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	clk_vicap_m1_src_sel clk_vicap_m1_src clock mux. 1'b0: clk_gpll_mux 1'b1: clk_cpll_mux
6:2	RW	0x01	clk_vicap_m1_src_div Divide clk_vicap_m1_src by (div_con + 1).
1:0	RW	0x2	clk_vicap_m0_out_sel clk_vicap_m0_out clock mux. 2'b00: clk_vicap_m0_src 2'b01: clk_vicap_m0_frac 2'b10: xin_osc0_func

CRU_CLKSEL_CON32

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:0	RW	0x001403de	clk_vicap_m1_frac_div clk_vicap_m1_frac fraction division register. High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x2	clk_vicap_m1_out_sel clk_vicap_m1_out clock mux. 2'b00: clk_vicap_m1_src 2'b01: clk_vicap_m1_frac 2'b10: xin_osc0_func

CRU_GATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_uart1_frac_en clk_uart1_frac clock gating control. When high, disable clock
14	RW	0x0	clk_uart1_en clk_uart1_src clock gating control. When high, disable clock
13	RW	0x0	sclk_uart0_en sclk_uart0_src clock gating control. When high, disable clock
12	RW	0x0	clk_uart0_frac_en clk_uart0_frac clock gating control. When high, disable clock
11	RW	0x0	clk_uart0_en clk_uart0_src clock gating control. When high, disable clock
10	RW	0x0	clk_matrix_500m_src_en clk_matrix_500m_src clock gating control. When high, disable clock
9	RW	0x0	clk_matrix_450m_src_en clk_matrix_450m_src clock gating control. When high, disable clock
8	RW	0x0	clk_matrix_400m_src_en clk_matrix_400m_src clock gating control. When high, disable clock
7	RW	0x0	clk_matrix_339m_src_en clk_matrix_339m_src clock gating control. When high, disable clock
6	RW	0x0	clk_matrix_300m_src_en clk_matrix_300m_src clock gating control. When high, disable clock
5	RW	0x0	clk_matrix_250m_src_en clk_matrix_250m_src clock gating control. When high, disable clock
4	RW	0x0	clk_matrix_200m_src_en clk_matrix_200m_src clock gating control. When high, disable clock
3	RW	0x0	clk_matrix_150m_src_en clk_matrix_150m_src clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	clk_matrix_100m_src_en clk_matrix_100m_src clock gating control. When high, disable clock
1	RW	0x0	clk_matrix_50m_src_en clk_matrix_50m_src clock gating control. When high, disable clock
0	RO	0x0	reserved

CRU_GATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mclk_i2s0_8ch_tx_src_en mclk_i2s0_8ch_tx_src_peri clock gating control. When high, disable clock
14	RW	0x0	clk_i2s0_8ch_frac_tx_en clk_i2s0_8ch_tx_frac clock gating control. When high, disable clock
13	RW	0x0	clk_i2s0_8ch_tx_en clk_i2s0_8ch_tx_src clock gating control. When high, disable clock
12	RW	0x0	sclk_uart5_en sclk_uart5_src clock gating control. When high, disable clock
11	RW	0x0	clk_uart5_frac_en clk_uart5_frac clock gating control. When high, disable clock
10	RW	0x0	clk_uart5_en clk_uart5_src clock gating control. When high, disable clock
9	RW	0x0	sclk_uart4_en sclk_uart4_src clock gating control. When high, disable clock
8	RW	0x0	clk_uart4_frac_en clk_uart4_frac clock gating control. When high, disable clock
7	RW	0x0	clk_uart4_en clk_uart4_src clock gating control. When high, disable clock
6	RW	0x0	sclk_uart3_en sclk_uart3_src clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
5	RW	0x0	clk_uart3_frac_en clk_uart3_frac clock gating control. When high, disable clock
4	RW	0x0	clk_uart3_en clk_uart3_src clock gating control. When high, disable clock
3	RW	0x0	sclk_uart2_en sclk_uart2_src clock gating control. When high, disable clock
2	RW	0x0	clk_uart2_frac_en clk_uart2_frac clock gating control. When high, disable clock
1	RW	0x0	clk_uart2_en clk_uart2_src clock gating control. When high, disable clock
0	RW	0x0	sclk_uart1_en sclk_uart1_src clock gating control. When high, disable clock

CRU_GATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	clk_cplltest_en clk_cplltest clock gating control. When high, disable clock
14	RW	0x1	clk_dplltest_en clk_dplltest clock gating control. When high, disable clock
13	RW	0x1	clk_aplltest_en clk_aplltest clock gating control. When high, disable clock
12	RW	0x0	clk_testout_en clk_testout_top2vepu clock gating control. When high, disable clock
11	RW	0x0	pclk_cru_biu_en pclk_cru_biu clock gating control. When high, disable clock
10	RW	0x0	pclk_cru_en pclk_cru clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	pclk_top_root_en pclk_top_root clock gating control. When high, disable clock
8	RW	0x0	clk_ref_pvtpll_1_en clk_ref_pvtpll_1 clock gating control. When high, disable clock
7	RW	0x0	clk_ref_pvtpll_0_en clk_ref_pvtpll_0 clock gating control. When high, disable clock
6	RW	0x0	dclk_vop_src_en dclk_vop_src clock gating control. When high, disable clock
5:3	RO	0x0	reserved
2	RW	0x0	mclk_i2s0_8ch_rx_en mclk_i2s0_8ch_rx_src_peri clock gating control. When high, disable clock
1	RW	0x0	clk_i2s0_8ch_frac_rx_en clk_i2s0_8ch_rx_frac clock gating control. When high, disable clock
0	RW	0x0	clk_i2s0_8ch_rx_en clk_i2s0_8ch_rx_src clock gating control. When high, disable clock

CRU_GATE_CON03

Address: Operational Base + offset (0x080C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_vicap_m1_out_en clk_vicap_m1_out clock gating control. When high, disable clock
14	RW	0x0	clk_vicap_m1_frac_en clk_vicap_m1_frac clock gating control. When high, disable clock
13	RW	0x0	clk_vicap_m1_src_en clk_vicap_m1_src clock gating control. When high, disable clock
12	RW	0x0	clk_vicap_m0_out_en clk_vicap_m0_out clock gating control. When high, disable clock
11	RW	0x0	clk_vicap_m0_frac_en clk_vicap_m0_frac clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
10	RW	0x0	clk_vicap_m0_src_en clk_vicap_m0_src clock gating control. When high, disable clock
9	RW	0x0	clk_ref_mipi1_out_en clk_ref_mipi1_out clock gating control. When high, disable clock
8	RW	0x0	clk_ref_mipi1_frac_en clk_ref_mipi1_frac clock gating control. When high, disable clock
7	RW	0x0	clk_ref_mipi1_src_en clk_ref_mipi1_src clock gating control. When high, disable clock
6	RW	0x0	clk_ref_mipi0_out_en clk_ref_mipi0_out clock gating control. When high, disable clock
5	RW	0x0	clk_ref_mipi0_frac_en clk_ref_mipi0_frac clock gating control. When high, disable clock
4	RW	0x0	clk_ref_mipi0_src_en clk_ref_mipi0_src clock gating control. When high, disable clock
3:1	RO	0x0	reserved
0	RW	0x1	clk_gplltest_en clk_gplltest clock gating control. When high, disable clock

CRU_SOFTRST_CON02

Address: Operational Base + offset (0x0A08)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	presetn_cru_biu When high, reset relative logic
10	RW	0x0	presetn_cru When high, reset relative logic
9	RO	0x0	reserved
8	RW	0x0	resetn_ref_pvtpll_1 When high, reset relative logic
7	RW	0x0	resetn_ref_pvtpll_0 When high, reset relative logic
6:0	RO	0x00	reserved

CRU_GLB_CNT_TH

Address: Operational Base + offset (0x0C00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0064	Reserved
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 (0x0C04)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RW	0x0	wdt0_src_st If High, glbrst by wdt0_src
10:7	RO	0x0	reserved
6	RW	0x0	glbrst_wdt0_rst If High, global reset by WDT0
5	RW	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
4	RW	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
3	RW	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
2	RW	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
1	RW	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	RW	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 (0x0C08)

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

CRU_GLB_SRST_SND

Address: Operational Base + offset (0x0C0C)

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

CRU_GLB_RST_CON

Address: Operational Base + offset (0x0C10)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RW	0x0	cru_wdt0_con 1'b0: Wdt0 trigger second global reset 1'b1: Wdt0 trigger first global reset
10:7	RO	0x0	reserved
6	RW	0x0	cru_wdt0_en 1'b0: Wdt0 trigger global reset disable 1'b1: Wdt0 trigger global reset enable
5	RO	0x0	reserved
4	RW	0x0	pmu_srst_wdt_en 0: Enable Wdt reset as pmu reset source 1: Disable Wdt reset as pmu reset source
3	RW	0x0	pmu_srst_glb_en 1'b0: global reset trigger pmu reset 1'b1: global reset dont trigger pmu reset
2	RW	0x0	pmu_srst_glb_ctrl 0: Enable first global reset as pmu reset source 1: Enable second global reset as pmu reset source effective when glb_rst_con[3] enable
1	RW	0x0	tsadc_glb_srst_en 1'b0: tsadc trigger global reset disable 1'b1: tsadc trigger global reset enable
0	RW	0x0	tsadc_glb_srst_ctrl 1'b0: tsadc trigger second global reset 1'b1: tsadc trigger first global reset

2.5 CORECRU Register Description

2.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CORECRU_CORECLKSEL_CON00	0x0300	W	0x00000081	Internal clock select and division register 0
CORECRU_CORECLKSEL_CON01	0x0304	W	0x00003847	Internal clock select and division register 1
CORECRU_CORECLKSEL_CON02	0x0308	W	0x00000005	Internal clock select and division register 2
CORECRU_CORECLKSEL_CON03	0x030C	W	0x000000C0	Internal clock select and division register 3
CORECRU_CORECLKSEL_CON04	0x0310	W	0x00000009	Internal clock select and division register 4
CORECRU_COREGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
CORECRU_COREGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
CORECRU_CORESOFTRST_CON00	0x0A00	W	0x00000100	Internal clock reset register 0
CORECRU_CORESOFTRST_CON01	0x0A04	W	0x0000001E	Internal clock reset register 1
CORECRU_AUTOCS_CORE_SRC_CON0	0x0D00	W	0x00040014	Pdcore auto clock swith control 0
CORECRU_AUTOCS_CORE_SRC_CON1	0x0D04	W	0x00000000	Pdcore auto clock swith control 1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.5.2 Detail Registers Description

CORECRU_CORECLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	clk_core_pvtpll_src_sel clk_core_pvtpll_src clock mux. 1'b0: clk_deepslow 1'b1: clk_core_pvtpll
11:7	RW	0x01	aclk_m_core_biu_div Divide aclk_m_core_biu by (div_con + 1).
6:5	RW	0x0	clk_core_src_sel clk_core_src clock mux. 2'b00: clk_apll_mux 2'b01: clk_core_pvtpll_src 2'b10: clk_core_gpll_src

Bit	Attr	Reset Value	Description
4:0	RW	0x01	clk_core_gpll_src_div Divide clk_core_gpll_src by (div_con + 1).

CORECRU_CORECLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RW	0x07	clk_scanhs_pclk_dbg_div Divide clk_scanhs_pclk_dbg by (div_con + 1).
10:6	RW	0x01	clk_scanhs_aclkm_core_div Divide clk_scanhs_aclkm_core by (div_con + 1).
5	RO	0x0	reserved
4:0	RW	0x07	pclk_dbg_div Divide pclk_dbg by (div_con + 1).

CORECRU_CORECLKSEL_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5	RW	0x0	hclk_cpu_biu_sel hclk_cpu_biu clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
4:0	RW	0x05	hclk_cpu_biu_div Divide hclk_cpu_biu by (div_con + 1).

CORECRU_CORECLKSEL_CON03

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_core_mcu_sel clk_core_mcu clock mux. 1'b0: clk_gpll_mux 1'b1: xin_osc0_func

Bit	Attr	Reset Value	Description
10:6	RW	0x03	clk_core_mcu_div Divide clk_core_mcu by (div_con + 1).
5	RW	0x0	clk_ref_pvtpll_core_sel clk_ref_pvtpll_core clock mux. 1'b0: xin_osc0_func 1'b1: clk_core
4:0	RW	0x00	clk_ref_pvtpll_core_div Divide clk_ref_pvtpll_core by (div_con + 1).

CORECRU_CORECLKSEL_CON04

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:5	RW	0x0	clk_testout_core2vi_sel clk_testout_core2vi clock mux. 2'b00: clk_core 2'b01: clk_core_mcu 2'b10: pclk_cpu_root 2'b11: hclk_cpu_biu
4:0	RW	0x09	clk_testout_core2vi_div Divide clk_testout_core2vi by (div_con + 1).

CORECRU_COREGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_pvtm_core_en pclk_pvtm_core clock gating control. When high, disable clock
14	RW	0x0	clk_pvtm_core_en clk_pvtm_core clock gating control. When high, disable clock
13	RW	0x0	pclk_cpu_biu_en pclk_cpu_biu clock gating control. When high, disable clock
12	RW	0x0	hclk_cpu_biu_en hclk_cpu_biu clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	pclk_core_grf_en pclk_core_grf clock gating control. When high, disable clock
10	RW	0x0	pclk_cpu_root_en pclk_cpu_root clock gating control. When high, disable clock
9	RW	0x0	clk_scanhs_core_en clk_scanhs_core clock gating control. When high, disable clock
8	RW	0x0	swclktck_en swclktck clock gating control. When high, disable clock
7	RO	0x0	reserved
6	RW	0x0	pclk_dbg_en pclk_dbg clock gating control. When high, disable clock
5	RW	0x0	aclk_m_core_biu_en aclk_m_core_biu clock gating control. When high, disable clock
4:3	RO	0x0	reserved
2	RW	0x0	clk_core_en clk_core clock gating control. When high, disable clock
1	RW	0x0	clk_core_src_en clk_core_src clock gating control. When high, disable clock
0	RW	0x0	clk_core_gpll_src_en clk_core_gpll_src clock gating control. When high, disable clock

CORECRU_COREGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk_testout_core_en clk_testout_core2vi clock gating control. When high, disable clock
12	RW	0x0	hclk_cache_en hclk_cache clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	pclk_core_sgrf_en pclk_core_sgrf clock gating control. When high, disable clock
10	RW	0x0	pclk_core_cru_en pclk_core_cru clock gating control. When high, disable clock
9	RW	0x0	pclk_intmux_en pclk_intmux clock gating control. When high, disable clock
8	RW	0x0	pclk_mailbox_en pclk_mailbox clock gating control. When high, disable clock
7	RW	0x0	clk_core_mcu_jtag_en clk_core_mcu_jtag clock gating control. When high, disable clock
6	RW	0x0	clk_core_mcu_rtc_en clk_core_mcu_rtc clock gating control. When high, disable clock
5	RW	0x0	clk_core_mcu_biu_en clk_core_mcu_biu clock gating control. When high, disable clock
4:2	RO	0x0	reserved
1	RW	0x0	clk_core_mcu_en clk_core_mcu clock gating control. When high, disable clock
0	RW	0x0	clk_ref_pvtpll_core_en clk_ref_pvtpll_core clock gating control. When high, disable clock

CORECRU_CORESOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_pvtm_core When high, reset relative logic
14	RW	0x0	resetn_pvtm_core When high, reset relative logic
13	RW	0x0	presetn_cpu_biu When high, reset relative logic
12	RW	0x0	hresetn_cpu_biu When high, reset relative logic

Bit	Attr	Reset Value	Description
11	RW	0x0	presetn_core_grf When high, reset relative logic
10:9	RO	0x0	reserved
8	RW	0x1	ntresetn_dbg When high, reset relative logic
7	RW	0x0	potresetn_dbg When high, reset relative logic
6	RW	0x0	presetn_dbg When high, reset relative logic
5	RW	0x0	aresetn_m_core_biu When high, reset relative logic
4	RW	0x0	nl2reset When high, reset relative logic
3	RW	0x0	ndbgreset When high, reset relative logic
2	RW	0x0	ncoreset When high, reset relative logic
1	RW	0x0	ncoreporeset When high, reset relative logic
0	RO	0x0	reserved

CORECRU_CORESOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	hresetn_cache When high, reset relative logic
11	RW	0x0	presetn_core_sgrf When high, reset relative logic
10	RW	0x0	presetn_core_cru When high, reset relative logic
9	RW	0x0	presetn_intmux When high, reset relative logic
8	RW	0x0	presetn_mailbox When high, reset relative logic
7:6	RO	0x0	reserved
5	RW	0x0	resetn_mcu_biu When high, reset relative logic
4	RW	0x1	tresetn_core_mcu_cpu When high, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x1	resetn_core_mcu_cpu When high, reset relative logic
2	RW	0x1	resetn_core_mcu_pwrup When high, reset relative logic
1	RW	0x1	resetn_core_mcu When high, reset relative logic
0	RW	0x0	resetn_ref_pvtpll_core When high, reset relative logic

CORECRU_AUTOCS_CORE_SRC_CON0

Address: Operational Base + offset (0x0D00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0004	clk_core_src_wait_th wait_th Wait time threshold, measured by original clk_core_src
15:0	RW	0x0014	clk_core_src_idle_th idle_th Idle time threshold, measured by original clk_core_src

CORECRU_AUTOCS_CORE_SRC_CON1

Address: Operational Base + offset (0x0D04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	clk_core_src_clksel_cfg Auto switch clock selection as clk_core_src 2'b00: Original clk_core_src 2'b01: xin_osc0_func_div 2'b10: clk_rtc_32k
13	RW	0x0	clk_core_src_switch_en switch_en 1'b1: Enable clk_core_src switch to lower frequency when module is inactive. 1'b0: Disable auto switch fuction.
12	RW	0x0	clk_core_src_autocs_en autocs_en 1'b1: Enable clk_core_src switch to lower frequency. 1'b0: Disable.
11:0	RW	0x000	clk_core_src_autocs_ctrl autocs_ctrl 12'hfff: Enable clk_core_src switch to lower frequency. 12'h000: Disable.

2.6 DDRCRU Register Description

2.6.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDRCRU_DDRCLKSEL_CON00	0x0300	W	0x00001000	Internal clock select and division register 0
DDRCRU_DDRGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
DDRCRU_DDRGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
DDRCRU_DDRSOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0
DDRCRU_DDRSOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.6.2 Detail Registers Description

DDRCRU_DDRCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_testout_ddr2peri_sel clk_testout_ddr2peri clock mux. 1'b0: aclk_ddr_root 1'b1: clk_core_ddrc
14:10	RW	0x04	clk_testout_ddr2peri_div Divide clk_testout_ddr2peri by (div_con + 1).
9:8	RW	0x0	aclk_ddr_root_sel aclk_ddr_root clock mux. 2'b00: clk_matrix_500m_src 2'b01: clk_matrix_300m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
7:2	RO	0x00	reserved
1:0	RW	0x0	pclk_ddr_root_sel pclk_ddr_root clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func

DDRCRU_DDRGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_ddr_grf_en pclk_ddr_grf clock gating control. When high, disable clock
14	RW	0x0	aclk_shrm_biu_en aclk_shrm_biu clock gating control. When high, disable clock
13	RW	0x0	aclk_sys_shrm_en aclk_sys_shrm clock gating control. When high, disable clock
12	RW	0x0	aclk_ddr_root_en aclk_ddr_root clock gating control. When high, disable clock
11	RW	0x0	pclk_dfictrl_en pclk_dfictrl clock gating control. When high, disable clock
10:9	RO	0x0	reserved
8	RW	0x0	clk_timer_ddrmon_en clk_timer_ddrmon clock gating control. When high, disable clock
7	RW	0x0	pclk_ddrmon_en pclk_ddrmon clock gating control. When high, disable clock
6	RO	0x0	reserved
5	RW	0x0	pclk_ddrc_en pclk_ddrc clock gating control. When high, disable clock
4:3	RO	0x0	reserved
2	RW	0x0	pclk_ddr_biu_en pclk_ddr_biu clock gating control. When high, disable clock
1	RO	0x0	reserved
0	RW	0x0	pclk_ddr_root_en pclk_ddr_root clock gating control. When high, disable clock

DDRCRU_DDRGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	mbist_clk_clk_ddr_biumem_en mbist_clk_clk_ddr_biumem clock gating control. When high, disable clock
5	RW	0x0	mbist_clk_clk_core_ddrc_en mbist_clk_clk_core_ddrc clock gating control. When high, disable clock
4	RO	0x0	reserved
3	RW	0x0	pclk_ddrphy_en pclk_ddrphy clock gating control. When high, disable clock
2	RW	0x0	pclk_ddr_hwlp_en pclk_ddr_hwlp clock gating control. When high, disable clock
1	RW	0x0	clk_ddr_testout_en clk_testout_ddr2peri clock gating control. When high, disable clock
0	RW	0x0	pclk_ddr_cru_en pclk_ddr_cru clock gating control. When high, disable clock

DDRCRU_DDRSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_ddr_grf When high, reset relative logic
14	RW	0x0	aresetn_shrm_biu When high, reset relative logic
13	RW	0x0	aresetn_sys_shrm When high, reset relative logic
12	RO	0x0	reserved
11	RW	0x0	presetn_dfictrl When high, reset relative logic
10:9	RO	0x0	reserved
8	RW	0x0	resetn_timer_ddrmon When high, reset relative logic

Bit	Attr	Reset Value	Description
7	RW	0x0	presetn_ddrmon When high, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	presetn_ddrc When high, reset relative logic
4:3	RO	0x0	reserved
2	RW	0x0	presetn_ddr_biu When high, reset relative logic
1:0	RO	0x0	reserved

DDRCRU_DDRSOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	presetn_ddrphy When high, reset relative logic
2	RW	0x0	presetn_ddr_hwlp When high, reset relative logic
1	RO	0x0	reserved
0	RW	0x0	presetn_ddr_cru When high, reset relative logic

2.7 SUBDDRCRU Register Description

2.7.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SUBDDRCRU_SUBDDRMODE_CON00	0x0280	W	0x00000000	Internal clock select and division register 0
SUBDDRCRU_SUBDDRCLKSEL_CON00	0x0300	W	0x00000001	Internal clock select and division register 0
SUBDDRCRU_SUBDDRGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
SUBDDRCRU_SUBDDRSOFTRST_CON00	0x0A00	W	0x00000040	Internal clock reset register 0

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.7.2 Detail Registers Description

SUBDDRCRU_SUBDDRMODE_CON00

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x0	clk_dpll_mode clk_dpll_mux clock mux. 2'b00: xin_osc0_func 2'b01: clk_dpll 2'b10: clk_deepslow

SUBDDRCRU_SUBDDRCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5	RW	0x0	clk_core_ddrc_src_sel clk_core_ddrc_src clock mux. 1'b0: clk_dpll_mux 1'b1: clk_matrix_300m_src
4:0	RW	0x01	clk_core_ddrc_src_div DT50 division register. Divide clk_core_ddrc_src by (div_con + 1).

SUBDDRCRU_SUBDDRGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	clk_ddr_phy_en clk_ddr_phy clock gating control. When high, disable clock
5	RW	0x0	clk_dfictrl_en clk_dfictrl clock gating control. When high, disable clock
4	RW	0x0	clk_ddrmon_en clk_ddrmon clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	clk_core_ddrc_en clk_core_ddrc clock gating control. When high, disable clock
2	RW	0x0	clk_core_ddrc_src_en clk_core_ddrc_src clock gating control. When high, disable clock
1	RW	0x0	aclk_ddrc_en aclk_ddrc clock gating control. When high, disable clock
0	RW	0x0	clk_msch_biu_en clk_msch_biu clock gating control. When high, disable clock

SUBDDRCRU_SUBDDRSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x1	resetn_ddr_phy When high, reset relative logic
5	RW	0x0	resetn_dfictrl When high, reset relative logic
4	RW	0x0	resetn_ddrmon When high, reset relative logic
3	RW	0x0	resetn_core_ddrc When high, reset relative logic
2	RO	0x0	reserved
1	RW	0x0	aresetn_ddrc When high, reset relative logic
0	RW	0x0	resetn_msch_biu When high, reset relative logic

2.8 NPUCRU Register Description

2.8.1 Registers Summary

Name	Offset	Size	Reset Value	Description
NPUCRU_NPUCLKSEL_CON00	0x0300	W	0x00000100	Internal clock select and division register 0
NPUCRU_NPUGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
NPUCRU_NPUGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1

Name	Offset	Size	Reset Value	Description
NPUCRU_NPUSOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.8.2 Detail Registers Description

NPUCRU_NPUCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:11	RW	0x0	clk_testout_npu2vi_sel clk_testout_npu2vi clock mux. 2'b00: aclk_npu_root 2'b01: hclk_npu_root 2'b10: pclk_npu_root
10:6	RW	0x04	clk_testout_npu2vi_div Divide clk_testout_npu2vi by (div_con + 1).
5:4	RW	0x0	pclk_npu_root_sel pclk_npu_root clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func
3:2	RW	0x0	aclk_npu_root_sel aclk_npu_root clock mux. 2'b00: clk_matrix_500m_src 2'b01: clk_matrix_300m_src 2'b10: clk_pvtpll_0 2'b11: clk_pvtpll_1
1:0	RW	0x0	hclk_npu_root_sel hclk_npu_root clock mux. 2'b00: clk_matrix_150m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func

NPUCRU_NPUGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_matrix_50m_src2vi_en clk_matrix_50m_src2vi clock gating control. When high, disable clock
14:13	RO	0x0	reserved
12	RW	0x0	clk_pvtpll_1_i2vi_en clk_pvtpll_1_i2vi clock gating control. When high, disable clock
11	RW	0x0	clk_pvtpll_0_i2vi_en clk_pvtpll_0_i2vi clock gating control. When high, disable clock
10	RW	0x0	aclk_rknn_en aclk_rknn clock gating control. When high, disable clock
9	RW	0x0	hclk_rknn_en hclk_rknn clock gating control. When high, disable clock
8	RW	0x0	pclk_npu_grf_en pclk_npu_grf clock gating control. When high, disable clock
7	RW	0x0	pclk_npu_sgrf_en pclk_npu_sgrf clock gating control. When high, disable clock
6	RW	0x0	pclk_npu_cru_en pclk_npu_cru clock gating control. When high, disable clock
5	RW	0x0	pclk_npu_biu_en pclk_npu_biu clock gating control. When high, disable clock
4	RW	0x0	aclk_npu_biu_en aclk_npu_biu clock gating control. When high, disable clock
3	RW	0x0	hclk_npu_biu_en hclk_npu_biu clock gating control. When high, disable clock
2	RW	0x0	pclk_npu_root_en pclk_npu_root clock gating control. When high, disable clock
1	RW	0x0	aclk_npu_root_en aclk_npu_root clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	hclk_npu_root_en hclk_npu_root clock gating control. When high, disable clock

NPUCRU_NPUGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_testout_npu_en clk_testout_npu2vi clock gating control. When high, disable clock
14	RW	0x0	clk_matrix_500m_src2peri_en clk_matrix_500m_src2peri clock gating control. When high, disable clock
13	RW	0x0	clk_matrix_400m_src2peri_en clk_matrix_400m_src2peri clock gating control. When high, disable clock
12	RW	0x0	clk_matrix_300m_src2peri_en clk_matrix_300m_src2peri clock gating control. When high, disable clock
11	RW	0x0	clk_matrix_200m_src2peri_en clk_matrix_200m_src2peri clock gating control. When high, disable clock
10	RW	0x0	clk_matrix_100m_src2peri_en clk_matrix_100m_src2peri clock gating control. When high, disable clock
9	RW	0x0	clk_matrix_50m_src2peri_en clk_matrix_50m_src2peri clock gating control. When high, disable clock
8:5	RO	0x0	reserved
4	RW	0x0	clk_matrix_400m_src2vi_en clk_matrix_400m_src2vi clock gating control. When high, disable clock
3	RW	0x0	clk_matrix_339m_src2vi_en clk_matrix_339m_src2vi clock gating control. When high, disable clock
2	RW	0x0	clk_matrix_200m_src2vi_en clk_matrix_200m_src2vi clock gating control. When high, disable clock
1	RW	0x0	clk_matrix_150m_src2vi_en clk_matrix_150m_src2vi clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	clk_matrix_100m_src2vi_en clk_matrix_100m_src2vi clock gating control. When high, disable clock

NPUCRU_NPUSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	aresetn_rknn When high, reset relative logic
9	RW	0x0	hresetn_rknn When high, reset relative logic
8	RW	0x0	presetn_npu_grf When high, reset relative logic
7	RW	0x0	presetn_npu_sgrf When high, reset relative logic
6	RW	0x0	presetn_npu_cru When high, reset relative logic
5	RW	0x0	presetn_npu_biu When high, reset relative logic
4	RW	0x0	aresetn_npu_biu When high, reset relative logic
3	RW	0x0	hresetn_npu_biu When high, reset relative logic
2:0	RO	0x0	reserved

2.9 PERICRU Register Description

2.9.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PERICRU_PERICLKSEL_CON01	0x0304	W	0x00000000	Internal clock select and division register 1
PERICRU_PERICLKSEL_CON02	0x0308	W	0x00000000	Internal clock select and division register 2
PERICRU_PERICLKSEL_CON05	0x0314	W	0x00000303	Internal clock select and division register 5
PERICRU_PERICLKSEL_CON06	0x0318	W	0x00000001	Internal clock select and division register 6
PERICRU_PERICLKSEL_CON07	0x031C	W	0x00002000	Internal clock select and division register 7

Name	Offset	Size	Reset Value	Description
PERICRU_PERICLKSEL_CON08	0x0320	W	0x00000303	Internal clock select and division register 8
PERICRU_PERICLKSEL_CON09	0x0324	W	0x00000000	Internal clock select and division register 9
PERICRU_PERICLKSEL_CON11	0x032C	W	0x00000000	Internal clock select and division register 11
PERICRU_PERIGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
PERICRU_PERIGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
PERICRU_PERIGATE_CON02	0x0808	W	0x00000000	Internal clock gate and division register 2
PERICRU_PERIGATE_CON03	0x080C	W	0x00000000	Internal clock gate and division register 3
PERICRU_PERIGATE_CON04	0x0810	W	0x00000000	Internal clock gate and division register 4
PERICRU_PERIGATE_CON05	0x0814	W	0x00000000	Internal clock gate and division register 5
PERICRU_PERIGATE_CON06	0x0818	W	0x00000000	Internal clock gate and division register 6
PERICRU_PERIGATE_CON07	0x081C	W	0x00000000	Internal clock gate and division register 7
PERICRU_PERISOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0
PERICRU_PERISOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1
PERICRU_PERISOFTRST_CON02	0x0A08	W	0x00000000	Internal clock reset register 2
PERICRU_PERISOFTRST_CON03	0x0A0C	W	0x00000000	Internal clock reset register 3
PERICRU_PERISOFTRST_CON04	0x0A10	W	0x00000000	Internal clock reset register 4
PERICRU_PERISOFTRST_CON05	0x0A14	W	0x00000000	Internal clock reset register 5
PERICRU_PERISOFTRST_CON06	0x0A18	W	0x00000000	Internal clock reset register 6
PERICRU_PERISOFTRST_CON07	0x0A1C	W	0x00000000	Internal clock reset register 7

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.9.2 Detail Registers Description

PERICRU_PERICLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	clk_i2c3_sel clk_i2c3 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
13:12	RW	0x0	clk_i2c2_sel clk_i2c2 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
11:10	RO	0x0	reserved
9:8	RW	0x0	clk_i2c0_sel clk_i2c0 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
7:6	RO	0x0	reserved
5:4	RW	0x0	hclk_peri_root_sel hclk_peri_root clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
3:2	RW	0x0	aclk_peri_root_sel aclk_peri_root clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
1:0	RW	0x0	pclk_peri_root_sel pclk_peri_root clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func

PERICRU_PERICLKSEL_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x0	clk_i2c4_sel clk_i2c4 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func

PERICRU_PERICLKSEL_CON05

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RW	0x0	clk_testout_peri2vi_1_sel clk_testout_peri2vi_1 clock mux. 3'b000: emmc_test_clkout_1 3'b001: sclk_uart2 3'b010: aclk_bus_root 3'b011: clk_core_crypto 3'b100: mclk_i2s0_8ch_tx
12:8	RW	0x03	clk_testout_peri2vi_1_div Divide clk_testout_peri2vi_1 by (div_con + 1).
7:5	RW	0x0	clk_testout_peri2vi_0_sel clk_testout_peri2vi_0 clock mux. 3'b000: emmc_test_clkout_0 3'b001: aclk_peri_root 3'b010: hclk_peri_root 3'b011: pclk_peri_root 3'b100: clk_testout_ddr2peri
4:0	RW	0x03	clk_testout_peri2vi_0_div Divide clk_testout_peri2vi_0 by (div_con + 1).

PERICRU_PERICLKSEL_CON06

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12:11	RW	0x0	clk_pwm2_peri_sel clk_pwm2_peri clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func
10:9	RW	0x0	clk_pwm1_peri_sel clk_pwm1_peri clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func
8:7	RW	0x0	clk_pka_crypto_sel clk_pka_crypto clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
6:5	RW	0x0	clk_core_crypto_sel clk_core_crypto clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
4:3	RW	0x0	clk_spi1_sel clk_spi1 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
2:0	RW	0x1	clk_saradc_div Divide clk_saradc by (div_con + 1).

PERICRU_PERICLKSEL_CON07

Address: Operational Base + offset (0x031C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	dclk_decom_sel dclk_decom clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func

Bit	Attr	Reset Value	Description
13:12	RW	0x2	sclk_sfc_sel sclk_sfc clock mux. 2'b00: clk_matrix_500m_src 2'b01: clk_matrix_300m_src 2'b10: clk_matrix_200m_src 2'b11: xin_osc0_func
11:7	RW	0x00	sclk_sfc_div Divide sclk_sfc by (div_con + 1).
6	RW	0x0	cclk_src_emmc_sel cclk_src_emmc clock mux. 1'b0: clk_matrix_400m_src 1'b1: xin_osc0_func
5:0	RW	0x00	cclk_src_emmc_div DT50 division register. Divide cclk_src_emmc by (div_con + 1).

PERICRU_PERICLKSEL_CON08

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x03	mclk_acodec_rx_div Divide mclk_acodec_rx by (div_con + 1).
7:0	RW	0x03	mclk_acodec_tx_div Divide mclk_acodec_tx by (div_con + 1).

PERICRU_PERICLKSEL_CON09

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x0	aclk_bus_root_sel aclk_bus_root clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func

PERICRU_PERICLKSEL_CON11

Address: Operational Base + offset (0x032C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1:0	RW	0x0	clk_pwm0_peri_sel clk_pwm0_peri clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func

PERICRU_PERIGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_stimer_en pclk_stimer clock gating control. When high, disable clock
14	RW	0x0	clk_timer5_en clk_timer5 clock gating control. When high, disable clock
13	RW	0x0	clk_timer4_en clk_timer4 clock gating control. When high, disable clock
12	RW	0x0	clk_timer3_en clk_timer3 clock gating control. When high, disable clock
11	RW	0x0	clk_timer2_en clk_timer2 clock gating control. When high, disable clock
10	RW	0x0	clk_timer1_en clk_timer1 clock gating control. When high, disable clock
9	RW	0x0	clk_timer0_en clk_timer0 clock gating control. When high, disable clock
8	RW	0x0	pclk_timer_en pclk_timer clock gating control. When high, disable clock
7	RW	0x0	hclk_bootrom_en hclk_bootrom clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	hclk_peri_biu_en hclk_peri_biu clock gating control. When high, disable clock
5	RW	0x0	aclk_peri_biu_en aclk_peri_biu clock gating control. When high, disable clock
4	RW	0x0	pclk_peri_biu_en pclk_peri_biu clock gating control. When high, disable clock
3	RW	0x0	clk_timer_root_en clk_timer_root clock gating control. When high, disable clock
2	RW	0x0	hclk_peri_root_en hclk_peri_root clock gating control. When high, disable clock
1	RW	0x0	aclk_peri_root_en aclk_peri_root clock gating control. When high, disable clock
0	RW	0x0	pclk_peri_root_en pclk_peri_root clock gating control. When high, disable clock

PERICRU_PERIGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_i2c4_en clk_i2c4 clock gating control. When high, disable clock
14	RW	0x0	pclk_i2c4_en pclk_i2c4 clock gating control. When high, disable clock
13	RW	0x0	clk_i2c3_en clk_i2c3 clock gating control. When high, disable clock
12	RW	0x0	pclk_i2c3_en pclk_i2c3 clock gating control. When high, disable clock
11	RW	0x0	clk_i2c2_en clk_i2c2 clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
10	RW	0x0	pclk_i2c2_en pclk_i2c2 clock gating control. When high, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	clk_i2c0_en clk_i2c0 clock gating control. When high, disable clock
6	RW	0x0	pclk_i2c0_en pclk_i2c0 clock gating control. When high, disable clock
5	RW	0x0	tclk_wdt_s_en tclk_wdt_s clock gating control. When high, disable clock
4	RW	0x0	pclk_wdt_s_en pclk_wdt_s clock gating control. When high, disable clock
3	RW	0x0	tclk_wdt_ns_en tclk_wdt_ns clock gating control. When high, disable clock
2	RW	0x0	pclk_wdt_ns_en pclk_wdt_ns clock gating control. When high, disable clock
1	RW	0x0	clk_stimer1_en clk_stimer1 clock gating control. When high, disable clock
0	RW	0x0	clk_stimer0_en clk_stimer0 clock gating control. When high, disable clock

PERICRU_PERIGATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_uart5_en pclk_uart5 clock gating control. When high, disable clock
14	RW	0x0	sclk_uart4_en sclk_uart4 clock gating control. When high, disable clock
13:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	pclk_uart4_en pclk_uart4 clock gating control. When high, disable clock
10	RW	0x0	sclk_uart3_en sclk_uart3 clock gating control. When high, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	pclk_uart3_en pclk_uart3 clock gating control. When high, disable clock
6	RW	0x0	sclk_uart2_en sclk_uart2 clock gating control. When high, disable clock
5:4	RO	0x0	reserved
3	RW	0x0	pclk_uart2_en pclk_uart2 clock gating control. When high, disable clock
2	RW	0x0	pclk_peri_ioc_en pclk_peri_ioc clock gating control. When high, disable clock
1	RW	0x0	dbclk_gpio4_en dbclk_gpio4 clock gating control. When high, disable clock
0	RW	0x0	pclk_gpio4_en pclk_gpio4 clock gating control. When high, disable clock

PERICRU_PERIGATE_CON03

Address: Operational Base + offset (0x080C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_pwm1_peri_en pclk_pwm1_peri clock gating control. When high, disable clock
14	RW	0x0	hclk_crypto_en hclk_crypto clock gating control. When high, disable clock
13	RW	0x0	aclk_crypto_en aclk_crypto clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
12	RW	0x0	clk_pka_crypto_en clk_pka_crypto clock gating control. When high, disable clock
11	RW	0x0	clk_core_crypto_en clk_core_crypto clock gating control. When high, disable clock
10	RW	0x0	hclk_trng_s_en hclk_trng_s clock gating control. When high, disable clock
9	RW	0x0	hclk_trng_ns_en hclk_trng_ns clock gating control. When high, disable clock
8	RW	0x0	sclk_in_spi1_en sclk_in_spi1 clock gating control. When high, disable clock
7	RW	0x0	clk_spi1_en clk_spi1 clock gating control. When high, disable clock
6	RW	0x0	pclk_spi1_en pclk_spi1 clock gating control. When high, disable clock
5	RO	0x0	reserved
4	RW	0x0	clk_saradc_en clk_saradc clock gating control. When high, disable clock
3	RW	0x0	pclk_saradc_en pclk_saradc clock gating control. When high, disable clock
2	RW	0x0	sclk_uart5_en sclk_uart5 clock gating control. When high, disable clock
1	RW	0x0	clk_testout_peri2vi_1_en clk_testout_peri2vi_1 clock gating control. When high, disable clock
0	RW	0x0	clk_testout_peri2vi_0_en clk_testout_peri2vi_0 clock gating control. When high, disable clock

PERICRU_PERIGATE_CON04

Address: Operational Base + offset (0x0810)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x0	hclk_sfc_en hclk_sfc clock gating control. When high, disable clock
13	RW	0x0	hclk_emmc_en hclk_emmc clock gating control. When high, disable clock
12	RW	0x0	cclk_src_emmc_en cclk_src_emmc clock gating control. When high, disable clock
11	RW	0x0	mbist_clk_aclk_usb_en mbist_clk_aclk_usb clock gating control. When high, disable clock
10	RW	0x0	aclk_bus_biu_en aclk_bus_biu clock gating control. When high, disable clock
9	RW	0x0	clk_utmi_usbotg_en clk_utmi_usbotg clock gating control. When high, disable clock
8	RW	0x0	clk_ref_usbotg_en clk_ref_usbotg clock gating control. When high, disable clock
7	RW	0x0	aclk_usbotg_en aclk_usbotg clock gating control. When high, disable clock
6	RW	0x0	pclk_peri_cru_en pclk_peri_cru clock gating control. When high, disable clock
5	RW	0x0	pclk_peri_grf_en pclk_peri_grf clock gating control. When high, disable clock
4	RW	0x0	clk_capture_pwm2_peri_en clk_capture_pwm2_peri clock gating control. When high, disable clock
3	RW	0x0	clk_pwm2_peri_en clk_pwm2_peri clock gating control. When high, disable clock
2	RW	0x0	pclk_pwm2_peri_en pclk_pwm2_peri clock gating control. When high, disable clock
1	RW	0x0	clk_capture_pwm1_peri_en clk_capture_pwm1_peri clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	clk_pwm1_peri_en clk_pwm1_peri clock gating control. When high, disable clock

PERICRU_PERIGATE_CON05

Address: Operational Base + offset (0x0814)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mclk_i2s0_8ch_tx_en mclk_i2s0_8ch_tx clock gating control. When high, disable clock
14	RW	0x0	mclk_sai_en mclk_sai clock gating control. When high, disable clock
13	RW	0x0	hclk_sai_en hclk_sai clock gating control. When high, disable clock
12	RW	0x0	pclk_peri_sgrf_en pclk_peri_sgrf clock gating control. When high, disable clock
11	RW	0x0	dclk_decom_en dclk_decom clock gating control. When high, disable clock
10	RW	0x0	pclk_decom_en pclk_decom clock gating control. When high, disable clock
9	RW	0x0	aclk_decom_en aclk_decom clock gating control. When high, disable clock
8	RW	0x0	aclk_dmac_en aclk_dmac clock gating control. When high, disable clock
7:3	RO	0x00	reserved
2	RW	0x0	clk_ref_usbphy_en clk_ref_usbphy clock gating control. When high, disable clock
1	RW	0x0	pclk_usbphy_en pclk_usbphy clock gating control. When high, disable clock
0	RW	0x0	sclk_sfc_en sclk_sfc clock gating control. When high, disable clock

PERICRU_PERIGATE_CON06

Address: Operational Base + offset (0x0818)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_uart1_en pclk_uart1 clock gating control. When high, disable clock
14	RW	0x0	sclk_uart0_en sclk_uart0 clock gating control. When high, disable clock
13:12	RO	0x0	reserved
11	RW	0x0	pclk_uart0_en pclk_uart0 clock gating control. When high, disable clock
10	RW	0x0	aclk_ive_en aclk_ive clock gating control. When high, disable clock
9	RW	0x0	hclk_ive_en hclk_ive clock gating control. When high, disable clock
8	RW	0x0	aclk_bus_root_en aclk_bus_root clock gating control. When high, disable clock
7	RW	0x0	pclk_dft2apb_en pclk_dft2apb clock gating control. When high, disable clock
6	RW	0x0	mclk_i2s0_8ch_rx_en mclk_i2s0_8ch_rx clock gating control. When high, disable clock
5	RW	0x0	mclk_acodec_rx_en mclk_acodec_rx clock gating control. When high, disable clock
4	RW	0x0	mclk_acodec_tx_en mclk_acodec_tx clock gating control. When high, disable clock
3	RW	0x0	pclk_acodec_en pclk_acodec clock gating control. When high, disable clock
2	RW	0x0	pclk_dsm_en pclk_dsm clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	mclk_dsm_en mclk_dsm clock gating control. When high, disable clock
0	RW	0x0	hclk_i2s0_en hclk_i2s0 clock gating control. When high, disable clock

PERICRU_PERIGATE_CON07

Address: Operational Base + offset (0x081C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	mbist_clk_decom_400m_en mbist_clk_decom_400m clock gating control. When high, disable clock
10	RW	0x0	mbist_clk_ive_400m_en mbist_clk_ive_400m clock gating control. When high, disable clock
9	RW	0x0	mbist_clk_clk_emmc_en mbist_clk_clk_emmc clock gating control. When high, disable clock
8	RW	0x0	mbist_clk_aclk_crypto_en mbist_clk_aclk_crypto clock gating control. When high, disable clock
7	RW	0x0	mbist_clk_clk_pka_crypto_en mbist_clk_clk_pka_crypto clock gating control. When high, disable clock
6	RW	0x0	mbist_clk_clk_bootrom_en mbist_clk_clk_bootrom clock gating control. When high, disable clock
5	RW	0x0	clk_capture_pwm0_peri_en clk_capture_pwm0_peri clock gating control. When high, disable clock
4	RW	0x0	clk_pwm0_peri_en clk_pwm0_peri clock gating control. When high, disable clock
3	RW	0x0	pclk_pwm0_peri_en pclk_pwm0_peri clock gating control. When high, disable clock
2	RW	0x0	sclk_uart1_en sclk_uart1 clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
1:0	RO	0x0	reserved

PERICRU_PERISOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_stimer When high, reset relative logic
14	RW	0x0	resetn_timer5 When high, reset relative logic
13	RW	0x0	resetn_timer4 When high, reset relative logic
12	RW	0x0	resetn_timer3 When high, reset relative logic
11	RW	0x0	resetn_timer2 When high, reset relative logic
10	RW	0x0	resetn_timer1 When high, reset relative logic
9	RW	0x0	resetn_timer0 When high, reset relative logic
8	RW	0x0	presetn_timer When high, reset relative logic
7	RW	0x0	hresetn_bootrom When high, reset relative logic
6	RW	0x0	hresetn_peri_biu When high, reset relative logic
5	RW	0x0	aresetn_peri_biu When high, reset relative logic
4	RW	0x0	presetn_peri_biu When high, reset relative logic
3:0	RO	0x0	reserved

PERICRU_PERISOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_i2c4 When high, reset relative logic

Bit	Attr	Reset Value	Description
14	RW	0x0	presetn_i2c4 When high, reset relative logic
13	RW	0x0	resetn_i2c3 When high, reset relative logic
12	RW	0x0	presetn_i2c3 When high, reset relative logic
11	RW	0x0	resetn_i2c2 When high, reset relative logic
10	RW	0x0	presetn_i2c2 When high, reset relative logic
9:8	RO	0x0	reserved
7	RW	0x0	resetn_i2c0 When high, reset relative logic
6	RW	0x0	presetn_i2c0 When high, reset relative logic
5	RW	0x0	tresetn_wdt_s When high, reset relative logic
4	RW	0x0	presetn_wdt_s When high, reset relative logic
3	RW	0x0	tresetn_wdt_ns When high, reset relative logic
2	RW	0x0	presetn_wdt_ns When high, reset relative logic
1	RW	0x0	resetn_stimer1 When high, reset relative logic
0	RW	0x0	resetn_stimer0 When high, reset relative logic

PERICRU_PERISOFTRST_CON02

Address: Operational Base + offset (0x0A08)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_uart5 When high, reset relative logic
14	RW	0x0	sresetn_uart4 When high, reset relative logic
13:12	RO	0x0	reserved
11	RW	0x0	presetn_uart4 When high, reset relative logic
10	RW	0x0	sresetn_uart3 When high, reset relative logic
9:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	presetn_uart3 When high, reset relative logic
6	RW	0x0	sresetn_uart2 When high, reset relative logic
5:4	RO	0x0	reserved
3	RW	0x0	presetn_uart2 When high, reset relative logic
2	RW	0x0	presetn_peri_ioc When high, reset relative logic
1	RW	0x0	dbresetn_gpio4 When high, reset relative logic
0	RW	0x0	presetn_gpio4 When high, reset relative logic

PERICRU_PERISOFTRST_CON03

Address: Operational Base + offset (0x0A0C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_pwm1_peri When high, reset relative logic
14	RW	0x0	hresetn_crypto When high, reset relative logic
13	RW	0x0	aresetn_crypto When high, reset relative logic
12	RW	0x0	resetn_pka_crypto When high, reset relative logic
11	RW	0x0	resetn_core_crypto When high, reset relative logic
10	RW	0x0	hresetn_trng_s When high, reset relative logic
9	RW	0x0	hresetn_trng_ns When high, reset relative logic
8	RO	0x0	reserved
7	RW	0x0	resetn_spi1 When high, reset relative logic
6	RW	0x0	presetn_spi1 When high, reset relative logic
5	RW	0x0	resetn_saradc_phy When high, reset relative logic
4	RW	0x0	resetn_saradc When high, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	presetn_saradc When high, reset relative logic
2	RW	0x0	sresetn_uart5 When high, reset relative logic
1:0	RO	0x0	reserved

PERICRU_PERISOFTRST_CON04

Address: Operational Base + offset (0x0A10)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	hresetn_sfc When high, reset relative logic
13	RW	0x0	hresetn_emmc When high, reset relative logic
12:11	RO	0x0	reserved
10	RW	0x0	aresetn_bus_biu When high, reset relative logic
9:8	RO	0x0	reserved
7	RW	0x0	aresetn_usbotg When high, reset relative logic
6	RW	0x0	presetn_peri_cru When high, reset relative logic
5	RW	0x0	presetn_peri_grf When high, reset relative logic
4	RO	0x0	reserved
3	RW	0x0	resetn_pwm2_peri When high, reset relative logic
2	RW	0x0	presetn_pwm2_peri When high, reset relative logic
1	RO	0x0	reserved
0	RW	0x0	resetn_pwm1_peri When high, reset relative logic

PERICRU_PERISOFTRST_CON05

Address: Operational Base + offset (0x0A14)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RW	0x0	mresetn_i2s0_8ch_tx When high, reset relative logic
14	RW	0x0	mresetn_sai When high, reset relative logic
13	RW	0x0	hresetn_sai When high, reset relative logic
12	RW	0x0	presetn_peri_sgrf When high, reset relative logic
11	RW	0x0	dresetn_decom When high, reset relative logic
10	RW	0x0	presetn_decom When high, reset relative logic
9	RW	0x0	aresetn_decom When high, reset relative logic
8	RW	0x0	aresetn_dmac When high, reset relative logic
7:4	RO	0x0	reserved
3	RW	0x0	resetn_usbphy_otg When high, reset relative logic
2	RW	0x0	resetn_usbphy_por When high, reset relative logic
1	RW	0x0	presetn_usbphy When high, reset relative logic
0	RW	0x0	sresetn_sfc When high, reset relative logic

PERICRU_PERISOFTRST_CON06

Address: Operational Base + offset (0x0A18)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_uart1 When high, reset relative logic
14	RW	0x0	sresetn_uart0 When high, reset relative logic
13:12	RO	0x0	reserved
11	RW	0x0	presetn_uart0 When high, reset relative logic
10	RW	0x0	aresetn_ive When high, reset relative logic
9	RW	0x0	hresetn_ive When high, reset relative logic
8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	presetn_dft2apb When high, reset relative logic
6	RW	0x0	mresetn_i2s0_8ch_rx When high, reset relative logic
5:4	RO	0x0	reserved
3	RW	0x0	presetn_acodec When high, reset relative logic
2	RW	0x0	presetn_dsm When high, reset relative logic
1	RW	0x0	mresetn_dsm When high, reset relative logic
0	RW	0x0	hresetn_i2s0 When high, reset relative logic

PERICRU_PERISOFTRST_CON07

Address: Operational Base + offset (0x0A1C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	resetn_pwm0_peri When high, reset relative logic
3	RW	0x0	presetn_pwm0_peri When high, reset relative logic
2	RW	0x0	sresetn_uart1 When high, reset relative logic
1:0	RO	0x0	reserved

2.10 PMUCRU Register Description

2.10.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMUCRU_PMUCLKSEL_CON00	0x0300	W	0x00006101	Internal clock select and division register 0
PMUCRU_PMUCLKSEL_CON01	0x0304	W	0x00000000	Internal clock select and division register 1
PMUCRU_PMUCLKSEL_CON06	0x0318	W	0x0040B71B	Internal clock select and division register 6
PMUCRU_PMUCLKSEL_CON07	0x031C	W	0x00000000	Internal clock select and division register 7
PMUCRU_PMUGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0

Name	Offset	Size	Reset Value	Description
PMUCRU_PMUGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
PMUCRU_PMUGATE_CON02	0x0808	W	0x00000000	Internal clock gate and division register 2
PMUCRU_PMUSOFTRST_CON00	0x0A00	W	0x00001E00	Internal clock reset register 0
PMUCRU_PMUSOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1
PMUCRU_PMUSOFTRST_CON02	0x0A08	W	0x00000000	Internal clock reset register 2

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.10.2 Detail Registers Description

PMUCRU_PMUCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dbclk_pmu_gpio0_sel dbclk_pmu_gpio0 clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
14:13	RW	0x3	clk_testout_pmu_sel clk_testout_pmu clock mux. 2'b00: hclk_pmu_root 2'b01: pclk_pmu_root 2'b10: clk_deepslow 2'b11: xin_osc0_div
12:8	RW	0x01	clk_testout_pmu_div Divide clk_testout_pmu by (div_con + 1).
7:6	RW	0x0	clk_i2c1_sel clk_i2c1 clock mux. 2'b00: clk_matrix_200m_pmusrc 2'b01: clk_matrix_100m_pmusrc 2'b10: xin_osc0_func 2'b11: clk_deepslow
5:4	RW	0x0	hclk_pmu_root_sel hclk_pmu_root clock mux. 2'b00: clk_matrix_200m_pmusrc 2'b01: clk_matrix_100m_pmusrc 2'b10: xin_osc0_func

Bit	Attr	Reset Value	Description
3	RW	0x0	pclk_pmu_root_sel pclk_pmu_root clock mux. 1'b0: clk_matrix_100m_pmusrc 1'b1: xin_osc0_func
2:0	RW	0x1	clk_matrix_100m_pmusrc_div Divide clk_matrix_100m_pmusrc by (div_con + 1).

PMUCRU_PMUCLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4:0	RW	0x00	clk_pvtm_pmu_div Divide clk_pvtm_pmu by (div_con + 1).

PMUCRU_PMUCLKSEL_CON06

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:0	RW	0x0040b71b	xin_osc0_div_div xin_osc0_div fraction division register. High 16-bit for numerator Low 16-bit for denominator

PMUCRU_PMUCLKSEL_CON07

Address: Operational Base + offset (0x031C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x0	tclk_pmu_wdt_sel tclk_pmu_wdt clock mux. 1'b0: xin_osc0_func 1'b1: clk_deepslow
1:0	RW	0x0	clk_deepslow_sel clk_deepslow clock mux. 2'b00: xin_osc0_div 2'b01: xin_osc1_32k 2'b10: clk_pvtm_32k

PMUCRU_PMUGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_scanhs_clk_pmu_mcu_en clk_scanhs_clk_pmu_mcu clock gating control. When high, disable clock
14	RW	0x0	clk_pmu_mcu_jtag_en clk_pmu_mcu_jtag clock gating control. When high, disable clock
13	RW	0x0	clk_pmu_mcu_rtc_en clk_pmu_mcu_rtc clock gating control. When high, disable clock
12:10	RO	0x0	reserved
9	RW	0x0	clk_pmu_mcu_en clk_pmu_mcu clock gating control. When high, disable clock
8	RW	0x0	hclk_pmu_sram_en hclk_pmu_sram clock gating control. When high, disable clock
7	RW	0x0	pclk_pmu_biu_en pclk_pmu_biu clock gating control. When high, disable clock
6	RW	0x0	hclk_pmu_biu_en hclk_pmu_biu clock gating control. When high, disable clock
5	RW	0x0	clk_testout_pmu_en clk_testout_pmu clock gating control. When high, disable clock
4	RW	0x0	clk_i2c1_en clk_i2c1 clock gating control. When high, disable clock
3	RW	0x0	pclk_i2c1_en pclk_i2c1 clock gating control. When high, disable clock
2	RW	0x0	hclk_pmu_root_en hclk_pmu_root clock gating control. When high, disable clock
1	RW	0x0	pclk_pmu_root_en pclk_pmu_root clock gating control. When high, disable clock
0	RO	0x0	reserved

PMUCRU_PMUGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_ddr_fail_safe_en clk_ddr_fail_safe clock gating control. When high, disable clock
14	RW	0x0	xin_osc0_div_en xin_osc0_div clock gating control. When high, disable clock
13:6	RO	0x00	reserved
5	RW	0x0	pclk_pvtm_pmu_en pclk_pvtm_pmu clock gating control. When high, disable clock
4	RW	0x0	clk_pvtm_pmu_en clk_pvtm_pmu clock gating control. When high, disable clock
3	RW	0x0	dbclk_pmu_gpio0_en dbclk_pmu_gpio0 clock gating control. When high, disable clock
2	RW	0x0	pclk_pmu_gpio0_en pclk_pmu_gpio0 clock gating control. When high, disable clock
1	RW	0x0	pclk_pmu_en pclk_pmu clock gating control. When high, disable clock
0	RW	0x0	clk_pmu_en clk_pmu clock gating control. When high, disable clock

PMUCRU_PMUGATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk_refout_en clk_refout clock gating control. When high, disable clock
12	RW	0x0	occ_scanclk_pmu_gpio_en occ_scanclk_pmu_gpio clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	occ_scanclk_pmu_mcu_jtag_en occ_scanclk_pmu_mcu_jtag clock gating control. When high, disable clock
10	RW	0x0	pclk_pmu_mailbox_en pclk_pmu_mailbox clock gating control. When high, disable clock
9	RW	0x0	tclk_pmu_wdt_en tclk_pmu_wdt clock gating control. When high, disable clock
8	RW	0x0	pclk_pmu_wdt_en pclk_pmu_wdt clock gating control. When high, disable clock
7	RO	0x0	reserved
6	RW	0x0	pclk_pmu_sgrf_en pclk_pmu_sgrf clock gating control. When high, disable clock
5	RW	0x0	pclk_pmu_grf_en pclk_pmu_grf clock gating control. When high, disable clock
4	RW	0x0	pclk_pmu_cru_en pclk_pmu_cru clock gating control. When high, disable clock
3	RW	0x0	pclk_pmu_ioc_en pclk_pmu_ioc clock gating control. When high, disable clock
2	RW	0x0	clk_pmu_32k_hp_timer_en clk_pmu_32k_hp_timer clock gating control. When high, disable clock
1	RW	0x0	clk_pmu_hp_timer_en clk_pmu_hp_timer clock gating control. When high, disable clock
0	RW	0x0	pclk_pmu_hp_timer_en pclk_pmu_hp_timer clock gating control. When high, disable clock

PMUCRU_PMUSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x1	tresetn_pmu_mcu_cpu When high, reset relative logic

Bit	Attr	Reset Value	Description
11	RW	0x1	resetn_pmu_mcu_cpu When high, reset relative logic
10	RW	0x1	resetn_pmu_mcu_pwrup When high, reset relative logic
9	RW	0x1	resetn_pmu_mcu When high, reset relative logic
8	RW	0x0	hresetn_pmu_sram When high, reset relative logic
7	RW	0x0	presetn_pmu_biu When high, reset relative logic
6	RW	0x0	hresetn_pmu_biu When high, reset relative logic
5	RO	0x0	reserved
4	RW	0x0	resetn_i2c1 When high, reset relative logic
3	RW	0x0	presetn_i2c1 When high, reset relative logic
2:0	RO	0x0	reserved

PMUCRU_PMUSOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_ddr_fail_safe When high, reset relative logic
14:6	RO	0x000	reserved
5	RW	0x0	presetn_pvtm_pmu When high, reset relative logic
4	RW	0x0	resetn_pvtm_pmu When high, reset relative logic
3	RW	0x0	dbresetn_pmu_gpio0 When high, reset relative logic
2	RW	0x0	presetn_pmu_gpio0 When high, reset relative logic
1:0	RO	0x0	reserved

PMUCRU_PMUSOFTRST_CON02

Address: Operational Base + offset (0x0A08)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	presetn_pmu_mailbox When high, reset relative logic
9	RW	0x0	tresetn_pmu_wdt When high, reset relative logic
8	RW	0x0	presetn_pmu_wdt When high, reset relative logic
7	RW	0x0	presetn_pmu_sgrf_remap When high, reset relative logic
6	RW	0x0	presetn_pmu_sgrf When high, reset relative logic
5	RW	0x0	presetn_pmu_grf When high, reset relative logic
4	RW	0x0	presetn_pmu_cru When high, reset relative logic
3	RW	0x0	presetn_pmu_ioc When high, reset relative logic
2	RW	0x0	resetn_pmu_32k_hp_timer When high, reset relative logic
1	RW	0x0	resetn_pmu_hp_timer When high, reset relative logic
0	RW	0x0	presetn_pmu_hp_timer When high, reset relative logic

2.11 VEPUCRU Register Description

2.11.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VEPUCRU_VEPUCLKSEL_CON00	0x0300	W	0x00000000	Internal clock select and division register 0
VEPUCRU_VEPUCLKSEL_CON01	0x0304	W	0x00000181	Internal clock select and division register 1
VEPUCRU_VEPUGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
VEPUCRU_VEPUGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
VEPUCRU_VEPUGATE_CON02	0x0808	W	0x00000000	Internal clock gate and division register 2
VEPUCRU_VEPUSOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0

Name	Offset	Size	Reset Value	Description
VEPUCRU_VEPUSOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.11.2 Detail Registers Description

VEPUCRU_VEPUCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x0	clk_spi0_sel clk_spi0 clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
11:10	RW	0x0	clk_core_vepu_dvbm_sel clk_core_vepu_dvbm clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
9:8	RW	0x0	clk_core_vepu_sel clk_core_vepu clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_300m_src 2'b10: clk_pvtpll_0 2'b11: clk_pvtpll_1
7:6	RW	0x0	pclk_vepu_root_sel pclk_vepu_root clock mux. 2'b00: clk_matrix_100m_src 2'b01: clk_matrix_50m_src 2'b10: xin_osc0_func
5:4	RW	0x0	aclk_vepu_root_sel aclk_vepu_root clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func

Bit	Attr	Reset Value	Description
3:2	RW	0x0	aclk_vepu_com_root_sel aclk_vepu_com_root clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
1:0	RW	0x0	hclk_vepu_root_sel hclk_vepu_root clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func

VEPUCRU_VEPUCLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x0	clk_testout_vepu2vo_1_sel clk_testout_vepu2vo_1 clock mux. 2'b00: clk_core_vepu 2'b01: clk_core_vepu_dvbm 2'b10: clk_testout_top2vepu
11:7	RW	0x03	clk_testout_vepu2vo_1_div Divide clk_testout_vepu2vo_1 by (div_con + 1).
6:5	RW	0x0	clk_testout_vepu2vo_0_sel clk_testout_vepu2vo_0 clock mux. 2'b00: hclk_vepu_root 2'b01: aclk_vepu_com_root 2'b10: aclk_vepu_root 2'b11: pclk_vepu_root
4:0	RW	0x01	clk_testout_vepu2vo_0_div Divide clk_testout_vepu2vo_0 by (div_con + 1).

VEPUCRU_VEPUGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RW	0x0	pclk_gpio1_en pclk_gpio1 clock gating control. When high, disable clock
14	RW	0x0	pclk_vicap_vepu_en pclk_vicap_vepu clock gating control. When high, disable clock
13	RW	0x0	clk_core_vepu_dvbm_en clk_core_vepu_dvbm clock gating control. When high, disable clock
12	RW	0x0	aclk_vepu_pp_en aclk_vepu_pp clock gating control. When high, disable clock
11	RW	0x0	hclk_vepu_pp_en hclk_vepu_pp clock gating control. When high, disable clock
10	RW	0x0	clk_core_vepu_en clk_core_vepu clock gating control. When high, disable clock
9	RW	0x0	aclk_vepu_en aclk_vepu clock gating control. When high, disable clock
8	RW	0x0	hclk_vepu_en hclk_vepu clock gating control. When high, disable clock
7	RW	0x0	pclk_vepu_biu_en pclk_vepu_biu clock gating control. When high, disable clock
6	RW	0x0	aclk_vepu_com_biu_en aclk_vepu_com_biu clock gating control. When high, disable clock
5	RW	0x0	aclk_vepu_biu_en aclk_vepu_biu clock gating control. When high, disable clock
4	RW	0x0	hclk_vepu_biu_en hclk_vepu_biu clock gating control. When high, disable clock
3	RW	0x0	pclk_vepu_root_en pclk_vepu_root clock gating control. When high, disable clock
2	RW	0x0	aclk_vepu_root_en aclk_vepu_root clock gating control. When high, disable clock
1	RW	0x0	aclk_vepu_com_root_en aclk_vepu_com_root clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	hclk_vepu_root_en hclk_vepu_root clock gating control. When high, disable clock

VEPUCRU_VEPUGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_matrix_300m_src2npu_en clk_matrix_300m_src2npu clock gating control. When high, disable clock
14	RW	0x0	clk_matrix_200m_src2npu_en clk_matrix_200m_src2npu clock gating control. When high, disable clock
13	RW	0x0	clk_matrix_150m_src2npu_en clk_matrix_150m_src2npu clock gating control. When high, disable clock
12	RW	0x0	clk_matrix_100m_src2npu_en clk_matrix_100m_src2npu clock gating control. When high, disable clock
11	RW	0x0	clk_matrix_50m_src2npu_en clk_matrix_50m_src2npu clock gating control. When high, disable clock
10	RW	0x0	clk_pvtpll_1_i2npu_en clk_pvtpll_1_i2npu clock gating control. When high, disable clock
9	RW	0x0	clk_pvtpll_0_i2npu_en clk_pvtpll_0_i2npu clock gating control. When high, disable clock
8	RW	0x0	clk_uart_detn_flt_en clk_uart_detn_flt clock gating control. When high, disable clock
7	RW	0x0	pclk_vepu_grf_en pclk_vepu_grf clock gating control. When high, disable clock
6	RW	0x0	pclk_vepu_sgrf_en pclk_vepu_sgrf clock gating control. When high, disable clock
5	RW	0x0	pclk_vepu_cru_en pclk_vepu_cru clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	sclk_in_spi0_en sclk_in_spi0 clock gating control. When high, disable clock
3	RW	0x0	clk_spi0_en clk_spi0 clock gating control. When high, disable clock
2	RW	0x0	pclk_spi0_en pclk_spi0 clock gating control. When high, disable clock
1	RW	0x0	pclk_vepu_ioc_en pclk_vepu_ioc clock gating control. When high, disable clock
0	RW	0x0	dbclk_gpio1_en dbclk_gpio1 clock gating control. When high, disable clock

VEPUCRU_VEPUGATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_testout_vepu_1_en clk_testout_vepu2vo_1 clock gating control. When high, disable clock
13	RW	0x0	clk_testout_vepu_0_en clk_testout_vepu2vo_0 clock gating control. When high, disable clock
12	RW	0x0	clk_matrix_400m_src2vo_en clk_matrix_400m_src2vo clock gating control. When high, disable clock
11	RW	0x0	clk_matrix_300m_src2vo_en clk_matrix_300m_src2vo clock gating control. When high, disable clock
10	RW	0x0	clk_matrix_200m_src2vo_en clk_matrix_200m_src2vo clock gating control. When high, disable clock
9	RW	0x0	clk_matrix_150m_src2vo_en clk_matrix_150m_src2vo clock gating control. When high, disable clock
8	RW	0x0	clk_matrix_100m_src2vo_en clk_matrix_100m_src2vo clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_matrix_50m_src2vo_en clk_matrix_50m_src2vo clock gating control. When high, disable clock
6	RW	0x0	clk_matrix_500m_src2ddr_en clk_matrix_500m_src2ddr clock gating control. When high, disable clock
5	RW	0x0	clk_matrix_300m_src2ddr_en clk_matrix_300m_src2ddr clock gating control. When high, disable clock
4	RW	0x0	clk_matrix_100m_src2ddr_en clk_matrix_100m_src2ddr clock gating control. When high, disable clock
3	RW	0x0	clk_matrix_50m_src2ddr_en clk_matrix_50m_src2ddr clock gating control. When high, disable clock
2	RW	0x0	clk_matrix_500m_src2npu_en clk_matrix_500m_src2npu clock gating control. When high, disable clock
1	RW	0x0	clk_matrix_400m_src2npu_en clk_matrix_400m_src2npu clock gating control. When high, disable clock
0	RW	0x0	clk_matrix_339m_src2npu_en clk_matrix_339m_src2npu clock gating control. When high, disable clock

VEPUCRU_VEPUSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_gpio1 When high, reset relative logic
14	RW	0x0	presetn_vicap_vepu When high, reset relative logic
13	RW	0x0	resetn_core_vepu_dvbm When high, reset relative logic
12	RW	0x0	aresetn_vepu_pp When high, reset relative logic
11	RW	0x0	hresetn_vepu_pp When high, reset relative logic
10	RW	0x0	resetn_core_vepu When high, reset relative logic

Bit	Attr	Reset Value	Description
9	RW	0x0	aresetn_vepu When high, reset relative logic
8	RW	0x0	hresetn_vepu When high, reset relative logic
7	RW	0x0	presetn_vepu_biu When high, reset relative logic
6	RW	0x0	aresetn_vepu_com_biu When high, reset relative logic
5	RW	0x0	aresetn_vepu_biu When high, reset relative logic
4	RW	0x0	hresetn_vepu_biu When high, reset relative logic
3:0	RO	0x0	reserved

VEPUCRU_VEPUSOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	resetn_uart_detn_flt When high, reset relative logic
7	RW	0x0	presetn_vepu_grf When high, reset relative logic
6	RW	0x0	presetn_vepu_sgrf When high, reset relative logic
5	RW	0x0	presetn_vepu_cru When high, reset relative logic
4	RO	0x0	reserved
3	RW	0x0	resetn_spi0 When high, reset relative logic
2	RW	0x0	presetn_spi0 When high, reset relative logic
1	RW	0x0	presetn_vepu_ioc When high, reset relative logic
0	RW	0x0	dbresetn_gpio1 When high, reset relative logic

2.12 VICRU Register Description

2.12.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VICRU_VICLKSEL_CON00	0x0300	W	0x00000000	Internal clock select and division register 0
VICRU_VICLKSEL_CON01	0x0304	W	0x00000000	Internal clock select and division register 1
VICRU_VICLKSEL_CON02	0x0308	W	0x00000006	Internal clock select and division register 2
VICRU_VICLKSEL_CON03	0x030C	W	0x00000003	Internal clock select and division register 3
VICRU_VIGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
VICRU_VIGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
VICRU_VIGATE_CON02	0x0808	W	0x00000000	Internal clock gate and division register 2
VICRU_VISOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0
VICRU_VISOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1
VICRU_VISOFTRST_CON02	0x0A08	W	0x00000000	Internal clock reset register 2

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.12.2 Detail Registers Description

VICRU_VICLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:9	RW	0x0	dclk_vicap_sel dclk_vicap clock mux. 2'b00: clk_matrix_339m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
8:7	RW	0x0	clk_core_isp3p2_sel clk_core_isp3p2 clock mux. 2'b00: clk_matrix_339m_src 2'b01: clk_matrix_200m_src 2'b10: clk_pvtpll_0 2'b11: clk_pvtpll_1

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_vi_rtc_root_sel pclk_vi_rtc_root clock mux. 1'b0: clk_matrix_50m_src 1'b1: xin_osc0_func
5:4	RW	0x0	pclk_vi_root_sel pclk_vi_root clock mux. 2'b00: clk_matrix_150m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
3:2	RW	0x0	aclk_vi_root_sel aclk_vi_root clock mux. 2'b00: clk_matrix_339m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
1:0	RW	0x0	hclk_vi_root_sel hclk_vi_root clock mux. 2'b00: clk_matrix_150m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func

VICRU_VICLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	cclk_src_sdmmc_sel cclk_src_sdmmc clock mux. 1'b0: clk_matrix_400m_src 1'b1: xin_osc0_func
13:8	RW	0x00	cclk_src_sdmmc_div DT50 division register. Divide cclk_src_sdmmc by (div_con + 1).
7:0	RO	0x00	reserved

VICRU_VICLKSEL_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8:6	RW	0x0	clk_testout_vi2peri_0_sel clk_testout_vi2peri_0 clock mux. 3'b000: sdmmc_test_clkout_0 3'b001: aclk_vi_root 3'b010: hclk_vi_root 3'b011: pclk_vi_root 3'b100: dclk_vicap
5:1	RW	0x03	clk_testout_vi2peri_0_div Divide clk_testout_vi2peri_0 by (div_con + 1).
0	RO	0x0	reserved

VICRU_VICLKSEL_CON03

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:5	RW	0x0	clk_testout_vi2peri_1_sel clk_testout_vi2peri_1 clock mux. 3'b000: sdmmc_test_clkout_1 3'b001: pclk_vi_rtc_root 3'b010: clk_core_isp3p2 3'b011: cclk_src_sdmmc 3'b100: clk_testout_core2vi 3'b101: clk_testout_npu2vi
4:0	RW	0x03	clk_testout_vi2peri_1_div Divide clk_testout_vi2peri_1 by (div_con + 1).

VICRU_VIGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	i1clk_vicap_en i1clk_vicap clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
14	RW	0x0	i0clk_vicap_en i0clk_vicap clock gating control. When high, disable clock
13	RW	0x0	hclk_vicap_en hclk_vicap clock gating control. When high, disable clock
12	RW	0x0	aclk_vicap_en aclk_vicap clock gating control. When high, disable clock
11	RW	0x0	pclk_vicap_en pclk_vicap clock gating control. When high, disable clock
10	RW	0x0	dclk_vicap_en dclk_vicap clock gating control. When high, disable clock
9	RW	0x0	clk_core_isp3p2_en clk_core_isp3p2 clock gating control. When high, disable clock
8	RW	0x0	aclk_isp3p2_en aclk_isp3p2 clock gating control. When high, disable clock
7	RW	0x0	hclk_isp3p2_en hclk_isp3p2 clock gating control. When high, disable clock
6	RW	0x0	pclk_vi_biu_en pclk_vi_biu clock gating control. When high, disable clock
5	RW	0x0	aclk_vi_biu_en aclk_vi_biu clock gating control. When high, disable clock
4	RW	0x0	hclk_vi_biu_en hclk_vi_biu clock gating control. When high, disable clock
3	RW	0x0	pclk_vi_rtc_root_en pclk_vi_rtc_root clock gating control. When high, disable clock
2	RW	0x0	pclk_vi_root_en pclk_vi_root clock gating control. When high, disable clock
1	RW	0x0	aclk_vi_root_en aclk_vi_root clock gating control. When high, disable clock
0	RW	0x0	hclk_vi_root_en hclk_vi_root clock gating control. When high, disable clock

VICRU_VIGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_gpio3_en pclk_gpio3 clock gating control. When high, disable clock
14	RW	0x0	pclk_mipicsiphy_en pclk_mipicsiphy clock gating control. When high, disable clock
13	RW	0x0	clk_sdmmc_detn_flt_en clk_sdmmc_detn_flt clock gating control. When high, disable clock
12	RW	0x0	hclk_sdmmc_en hclk_sdmmc clock gating control. When high, disable clock
11	RW	0x0	cclk_src_sdmmc_en cclk_src_sdmmc clock gating control. When high, disable clock
10	RW	0x0	mbist_clk_clk_core_isp3p2_en mbist_clk_clk_core_isp3p2 clock gating control. When high, disable clock
9	RW	0x0	mbist_clk_aclk_isp3p2_en mbist_clk_aclk_isp3p2 clock gating control. When high, disable clock
8	RW	0x0	mbist_clk_dclk_vicap_en mbist_clk_dclk_vicap clock gating control. When high, disable clock
7	RW	0x0	mbist_clk_aclk_vicap_en mbist_clk_aclk_vicap clock gating control. When high, disable clock
6	RW	0x0	clk_rxbyteclkhs_1_en clk_rxbyteclkhs_1 clock gating control. When high, disable clock
5	RW	0x0	pclk_csihost1_en pclk_csihost1 clock gating control. When high, disable clock
4	RW	0x0	clk_rxbyteclkhs_0_en clk_rxbyteclkhs_0 clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	pclk_csihost0_en pclk_csihost0 clock gating control. When high, disable clock
2	RW	0x0	isp0clk_vicap_en isp0clk_vicap clock gating control. When high, disable clock
1	RW	0x0	rx1pclk_vicap_en rx1pclk_vicap clock gating control. When high, disable clock
0	RW	0x0	rx0pclk_vicap_en rx0pclk_vicap clock gating control. When high, disable clock

VICRU_VIGATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_testout_vi2peri_1_en clk_testout_vi2peri_1 clock gating control. When high, disable clock
10	RW	0x0	clk_testout_vi2peri_0_en clk_testout_vi2peri_0 clock gating control. When high, disable clock
9:8	RO	0x0	reserved
7	RW	0x0	pclk_vi_rtc_biu_en pclk_vi_rtc_biu clock gating control. When high, disable clock
6	RW	0x0	pclk_vi_rtc_phy_en pclk_vi_rtc_phy clock gating control. When high, disable clock
5	RW	0x0	pclk_vi_rtc_test_en pclk_vi_rtc_test clock gating control. When high, disable clock
4	RW	0x0	pclk_vi_cru_en pclk_vi_cru clock gating control. When high, disable clock
3	RW	0x0	pclk_vi_sgrf_en pclk_vi_sgrf clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	pclk_vi_grf_en pclk_vi_grf clock gating control. When high, disable clock
1	RW	0x0	pclk_vi_ioc_en pclk_vi_ioc clock gating control. When high, disable clock
0	RW	0x0	dbclk_gpio3_en dbclk_gpio3 clock gating control. When high, disable clock

VICRU_VISOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	i1resetn_vicap When high, reset relative logic
14	RW	0x0	i0resetn_vicap When high, reset relative logic
13	RW	0x0	hresetn_vicap When high, reset relative logic
12	RW	0x0	aresetn_vicap When high, reset relative logic
11	RW	0x0	presetn_vicap When high, reset relative logic
10	RW	0x0	dresetn_vicap When high, reset relative logic
9	RW	0x0	resetn_core_isp3p2 When high, reset relative logic
8:7	RO	0x0	reserved
6	RW	0x0	presetn_vi_biu When high, reset relative logic
5	RW	0x0	aresetn_vi_biu When high, reset relative logic
4	RW	0x0	hresetn_vi_biu When high, reset relative logic
3:0	RO	0x0	reserved

VICRU_VISOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_gpio3 When high, reset relative logic
14	RW	0x0	presetn_mipicsiphy When high, reset relative logic
13	RW	0x0	resetn_sdmmc_detn_flt When high, reset relative logic
12	RW	0x0	hresetn_sdmmc When high, reset relative logic
11:7	RO	0x00	reserved
6	RW	0x0	presetn_csihost1 When high, reset relative logic
5	RO	0x0	reserved
4	RW	0x0	presetn_csihost0 When high, reset relative logic
3	RO	0x0	reserved
2	RW	0x0	isp0resetn_vicap When high, reset relative logic
1	RW	0x0	rx1presetn_vicap When high, reset relative logic
0	RW	0x0	rx0presetn_vicap When high, reset relative logic

VICRU_VISOFTRST_CON02

Address: Operational Base + offset (0x0A08)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	presetn_vi_rtc_biu When high, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	presetn_vi_rtc_test When high, reset relative logic
4	RW	0x0	presetn_vi_cru When high, reset relative logic
3	RW	0x0	presetn_vi_sgrf When high, reset relative logic
2	RW	0x0	presetn_vi_grf When high, reset relative logic

Bit	Attr	Reset Value	Description
1	RW	0x0	presetn_vi_ioc When high, reset relative logic
0	RW	0x0	dbresetn_gpio3 When high, reset relative logic

2.13 VOCRU Register Description

2.13.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VOCRU_VOCLKSEL_CON00	0x0300	W	0x000000C0	Internal clock select and division register 0
VOCRU_VOCLKSEL_CON01	0x0304	W	0x00000003	Internal clock select and division register 1
VOCRU_VOCLKSEL_CON02	0x0308	W	0x00000000	Internal clock select and division register 2
VOCRU_VOCLKSEL_CON03	0x030C	W	0x00002433	Internal clock select and division register 3
VOCRU_VOGATE_CON00	0x0800	W	0x00000000	Internal clock gate and division register 0
VOCRU_VOGATE_CON01	0x0804	W	0x00000000	Internal clock gate and division register 1
VOCRU_VOGATE_CON02	0x0808	W	0x00000000	Internal clock gate and division register 2
VOCRU_VOGATE_CON03	0x080C	W	0x00000000	Internal clock gate and division register 3
VOCRU_VOSOFTRST_CON00	0x0A00	W	0x00000000	Internal clock reset register 0
VOCRU_VOSOFTRST_CON01	0x0A04	W	0x00000000	Internal clock reset register 1
VOCRU_VOSOFTRST_CON02	0x0A08	W	0x00000000	Internal clock reset register 2
VOCRU_VOSOFTRST_CON03	0x0A0C	W	0x00000000	Internal clock reset register 3

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

2.13.2 Detail Registers Description

VOCRU_VOCLKSEL_CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:11	RW	0x0	clk_testout_vo_0_sel clk_testout_vo_0 clock mux. 3'b000: sdio_test_clkout_0 3'b001: clk_testout_vepu2vo_0 3'b010: aclk_vo_root 3'b011: hclk_vo_root 3'b100: clk_core_rga2e 3'b101: aclk_vop_root
10:6	RW	0x03	clk_testout_vo_0_div Divide clk_testout_vo_0 by (div_con + 1).
5:4	RW	0x0	pclk_vo_root_sel pclk_vo_root clock mux. 2'b00: clk_matrix_150m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
3:2	RW	0x0	hclk_vo_root_sel hclk_vo_root clock mux. 2'b00: clk_matrix_200m_src 2'b01: clk_matrix_100m_src 2'b10: clk_matrix_50m_src 2'b11: xin_osc0_func
1:0	RW	0x0	aclk_vo_root_sel aclk_vo_root clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func

VOCRU_VOCLKSEL_CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x0	aclk_mac_root_sel aclk_mac_root clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func

Bit	Attr	Reset Value	Description
11:10	RW	0x0	aclk_vop_root_sel aclk_vop_root clock mux. 2'b00: clk_matrix_300m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
9:8	RW	0x0	clk_core_rga2e_sel clk_core_rga2e clock mux. 2'b00: clk_matrix_400m_src 2'b01: clk_matrix_200m_src 2'b10: clk_matrix_100m_src 2'b11: xin_osc0_func
7:5	RW	0x0	clk_testout_vo_1_sel clk_testout_vo_1 clock mux. 3'b000: sdio_test_clkout_1 3'b001: clk_testout_vepu2vo_1 3'b010: aclk_mac_root 3'b011: pclk_vo_root 3'b100: cclk_src_sdio 3'b101: dclk_vop
4:0	RW	0x03	clk_testout_vo_1_div Divide clk_testout_vo_1 by (div_con + 1).

VOCRU_VOCLKSEL_CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	cclk_src_sdio_sel cclk_src_sdio clock mux. 1'b0: clk_matrix_400m_src 1'b1: xin_osc0_func
12:7	RW	0x00	cclk_src_sdio_div DT50 division register. Divide cclk_src_sdio by (div_con + 1).
6:1	RW	0x00	clk_gmac0_tx_50m_o_div Divide clk_gmac0_tx_50m_o by (div_con + 1).
0	RO	0x0	reserved

VOCRU_VOCLKSEL_CON03

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RW	0x1	clk_user_otpc_s_div Divide clk_user_otpc_s by (div_con + 1).
12:10	RW	0x1	clk_user_otpc_ns_div Divide clk_user_otpc_ns by (div_con + 1).
9:5	RW	0x01	clk_tsadc_tsen_div Divide clk_tsadc_tsen by (div_con + 1).
4:0	RW	0x13	clk_tsadc_div Divide clk_tsadc by (div_con + 1).

VOCRU_VOGATE_CON00

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	aclk_vop_en aclk_vop clock gating control. When high, disable clock
14	RW	0x0	dclk_vop_en dclk_vop clock gating control. When high, disable clock
13	RW	0x0	hclk_vop_en hclk_vop clock gating control. When high, disable clock
12	RW	0x0	aclk_vop_biu_en aclk_vop_biu clock gating control. When high, disable clock
11	RW	0x0	aclk_vop_root_en aclk_vop_root clock gating control. When high, disable clock
10	RW	0x0	pclk_vo_grf_en pclk_vo_grf clock gating control. When high, disable clock
9	RW	0x0	clk_core_rga2e_en clk_core_rga2e clock gating control. When high, disable clock
8	RW	0x0	aclk_rga2e_en aclk_rga2e clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	hclk_rga2e_en hclk_rga2e clock gating control. When high, disable clock
6	RW	0x0	clk_testout_vo_1_en clk_testout_vo_1 clock gating control. When high, disable clock
5	RW	0x0	clk_testout_vo_0_en clk_testout_vo_0 clock gating control. When high, disable clock
4	RW	0x0	hclk_vo_biu_en hclk_vo_biu clock gating control. When high, disable clock
3	RW	0x0	aclk_vo_biu_en aclk_vo_biu clock gating control. When high, disable clock
2	RW	0x0	pclk_vo_root_en pclk_vo_root clock gating control. When high, disable clock
1	RW	0x0	hclk_vo_root_en hclk_vo_root clock gating control. When high, disable clock
0	RW	0x0	aclk_vo_root_en aclk_vo_root clock gating control. When high, disable clock

VOCRU_VOGATE_CON01

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hclk_sdio_en hclk_sdio clock gating control. When high, disable clock
14	RW	0x0	cclk_src_sdio_en cclk_src_sdio clock gating control. When high, disable clock
13	RW	0x0	pclk_vo_cru_en pclk_vo_cru clock gating control. When high, disable clock
12	RW	0x0	pclk_vo_sgrf_en pclk_vo_sgrf clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	mbist_clk_aclk_mac_en mbist_clk_aclk_mac clock gating control. When high, disable clock
10	RO	0x0	reserved
9	RW	0x0	pclk_mac_en pclk_mac clock gating control. When high, disable clock
8	RW	0x0	aclk_mac_en aclk_mac clock gating control. When high, disable clock
7	RW	0x0	aclk_mac_biu_en aclk_mac_biu clock gating control. When high, disable clock
6	RW	0x0	pclk_mac_biu_en pclk_mac_biu clock gating control. When high, disable clock
5	RO	0x0	reserved
4	RW	0x0	aclk_mac_root_en aclk_mac_root clock gating control. When high, disable clock
3	RW	0x0	mbist_clk_cclk_rga_en mbist_clk_cclk_rga clock gating control. When high, disable clock
2	RW	0x0	mbist_clk_aclk_rga_en mbist_clk_aclk_rga clock gating control. When high, disable clock
1	RW	0x0	mbist_clk_aclk_vop_en mbist_clk_aclk_vop clock gating control. When high, disable clock
0	RW	0x0	mbist_clk_mac_phy_en mbist_clk_mac_phy clock gating control. When high, disable clock

VOCRU_VOGATE_CON02

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_pmc_otp_en clk_pmc_otp clock gating control. When high, disable clock

Bit	Attr	Reset Value	Description
14	RW	0x0	pclk_otp_mask_en pclk_otp_mask clock gating control. When high, disable clock
13	RW	0x0	clk_macphy_en clk_macphy clock gating control. When high, disable clock
12	RO	0x0	reserved
11	RW	0x0	clk_otpc_arb_en clk_otpc_arb clock gating control. When high, disable clock
10	RW	0x0	clk_user_otpc_s_en clk_user_otpc_s clock gating control. When high, disable clock
9	RW	0x0	clk_sbpi_otpc_s_en clk_sbpi_otpc_s clock gating control. When high, disable clock
8	RO	0x0	reserved
7	RW	0x0	pclk_otpc_s_en pclk_otpc_s clock gating control. When high, disable clock
6	RW	0x0	clk_user_otpc_ns_en clk_user_otpc_ns clock gating control. When high, disable clock
5	RW	0x0	clk_sbpi_otpc_ns_en clk_sbpi_otpc_ns clock gating control. When high, disable clock
4	RO	0x0	reserved
3	RW	0x0	pclk_otpc_ns_en pclk_otpc_ns clock gating control. When high, disable clock
2	RW	0x0	clk_tsadc_tsen_en clk_tsadc_tsen clock gating control. When high, disable clock
1	RW	0x0	clk_tsadc_en clk_tsadc clock gating control. When high, disable clock
0	RW	0x0	pclk_tsadc_en pclk_tsadc clock gating control. When high, disable clock

VOCRU_VOGATE_CON03

Address: Operational Base + offset (0x080C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x0	pclk_vo_ioc_en pclk_vo_ioc clock gating control. When high, disable clock
1	RW	0x0	dbclk_gpio2_en dbclk_gpio2 clock gating control. When high, disable clock
0	RW	0x0	pclk_gpio2_en pclk_gpio2 clock gating control. When high, disable clock

VOCRU_VOSOFTRST_CON00

Address: Operational Base + offset (0x0A00)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	aresetn_vop When high, reset relative logic
14	RW	0x0	dresetn_vop When high, reset relative logic
13	RW	0x0	hresetn_vop When high, reset relative logic
12	RW	0x0	aresetn_vop_biu When high, reset relative logic
11	RO	0x0	reserved
10	RW	0x0	presetn_vo_grf When high, reset relative logic
9	RW	0x0	resetn_core_rga2e When high, reset relative logic
8	RW	0x0	aresetn_rga2e When high, reset relative logic
7	RW	0x0	hresetn_rga2e When high, reset relative logic
6:5	RO	0x0	reserved
4	RW	0x0	hresetn_vo_biu When high, reset relative logic
3	RW	0x0	aresetn_vo_biu When high, reset relative logic
2:0	RO	0x0	reserved

VOCRU_VOSOFTRST_CON01

Address: Operational Base + offset (0x0A04)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hresetn_sdio When high, reset relative logic
14	RO	0x0	reserved
13	RW	0x0	presetn_vo_cru When high, reset relative logic
12	RW	0x0	presetn_vo_sgrf When high, reset relative logic
11:9	RO	0x0	reserved
8	RW	0x0	aresetn_mac When high, reset relative logic
7	RW	0x0	aresetn_mac_biu When high, reset relative logic
6	RW	0x0	presetn_mac_biu When high, reset relative logic
5:0	RO	0x00	reserved

VOCRU_VOSOFTRST_CON02

Address: Operational Base + offset (0x0A08)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_pmc_otp When high, reset relative logic
14	RW	0x0	presetn_otp_mask When high, reset relative logic
13	RW	0x0	resetn_macphy When high, reset relative logic
12	RO	0x0	reserved
11	RW	0x0	resetn_otpc_arb When high, reset relative logic
10	RW	0x0	resetn_user_otpc_s When high, reset relative logic
9	RW	0x0	resetn_sbpi_otpc_s When high, reset relative logic
8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	presetn_otpc_s When high, reset relative logic
6	RW	0x0	resetn_user_otpc_ns When high, reset relative logic
5	RW	0x0	resetn_sbpi_otpc_ns When high, reset relative logic
4	RO	0x0	reserved
3	RW	0x0	presetn_otpc_ns When high, reset relative logic
2	RO	0x0	reserved
1	RW	0x0	resetn_tsadc When high, reset relative logic
0	RW	0x0	presetn_tsadc When high, reset relative logic

VOCRU_VOSOFTRST_CON03

Address: Operational Base + offset (0x0A0C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x0	presetn_vo_ioc When high, reset relative logic
1	RW	0x0	dbresetn_gpio2 When high, reset relative logic
0	RW	0x0	presetn_gpio2 When high, reset relative logic

2.14 Application Notes

2.14.1 PLL Usage

2.14.1.1 Start-up Operation

Set the PLL registers at time t3 after setting the power from 0V to VDD at time t2.Set PLLEN=0 before PLL starts to operate. Then set PLLEN=1 at t4 to initialize PLL. Note that PLL generates a clock signal with the desired frequency only after Lock time has been passed.PLLEN goes hign 1us after power, refence clock and all inputs are stable, that means t4-t2 should be more than 1us.

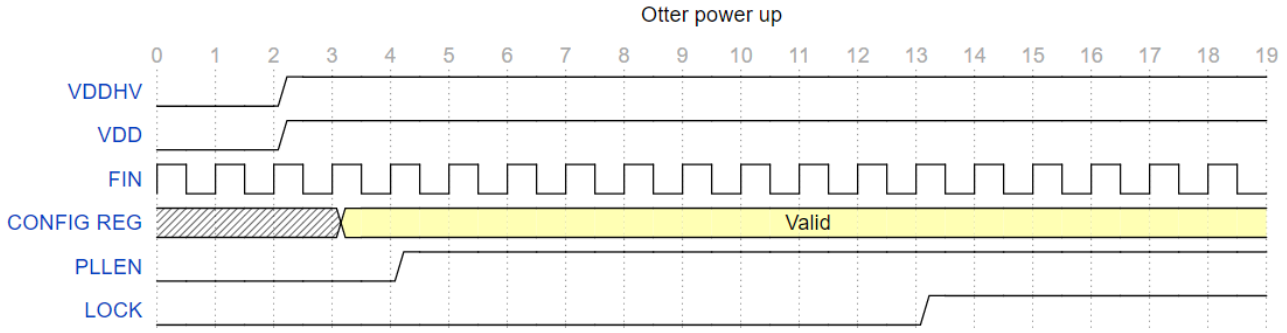


Fig. 2-5 PLL Start-up Operation

To change PLL config, you must set PLLEN=0 first. The other input pins are not allowed to be changed after PLL is locked.

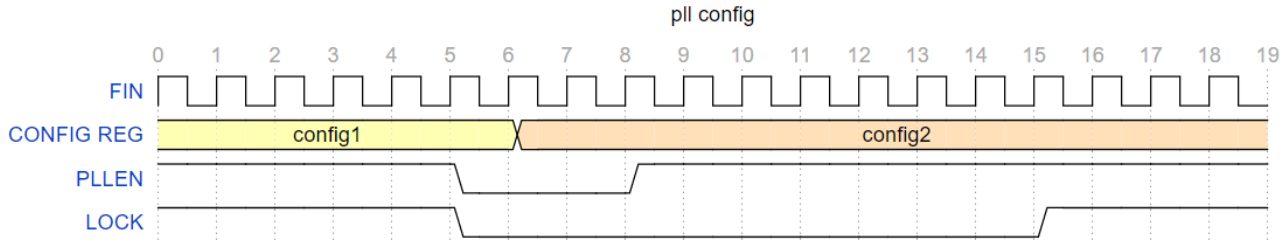


Fig. 2-6 PLL Config Change Operation

2.14.1.2 Bypass Operation

Bypass mode functions only when PLL is operating (PLLEN=1). In bypass mode, the frequency of FOUT will be set to be equal to FIN. When PLL get locked (LOCK = 1), PLL generates a stable clock signal with desired target frequency.

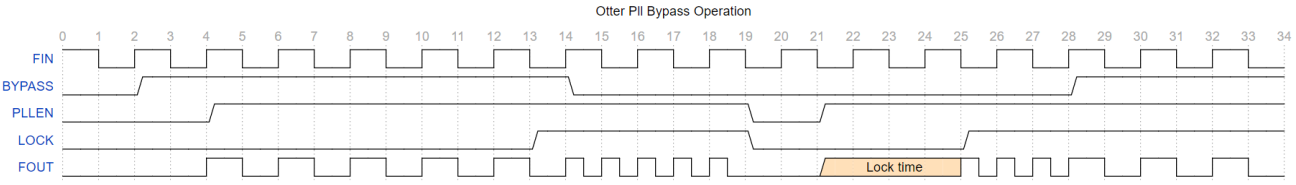


Fig. 2-7 PLL Bypass Operation

2.14.1.3 Setting Guide on PLLEN and BYPASS

PLLEN is a reset signal of PLL. PLL enters the power down mode when PLLEN= 0.
BYPASS enables the bypass mode of PLL. When BYPASS= 1, PLL enters the bypass mode and FOUT is equal to FIN.

2.14.2 Fractional PLL Usage

2.14.2.1 PLL 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.(x= D, G)

- 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

- POSTDIV1=0, POSTDIV2=0 are unused. We should make sure that POSTDIV1>=POSTDIV2.

2.14.2.2 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 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.

2.14.2.3 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_xPLL_CON1[10] (x= D, G) register. The lock state is high when both original hardware PLL lock and PLL counter lock are high.
The max delay time is 500*REFDIV/FREF.
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 128 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 128*REFDIV/FREF.

2.14.3 Integer PLL Usage

2.14.3.1 PLL Frequency Configuration

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_xPLL_CON0.
DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_xPLL_CON1(x=A, C).
DSMPD should always be set to 1 (CLKSSCG is power down), and REFDIV can only be 1 or 2.
FOUTVCO = (FREF / REFDIV) * FBDIV
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
POSTDIV1=0, POSTDIV2=0 are unused.
We should make sure that POSTDIV1>=POSTDIV2.

2.14.3.2 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_xPLL_CON1[10] (x=A, C) register. The lock state is high when both original hardware PLL lock and PLL counter lock are high.
The max delay time is 1500*REFDIV/FREF.

2.14.4 Divider Usage

CRU supports multi-dividers for different clock requirement.

- Divider free divider
- Fractional divider
- DivfreeDT50 divider(DT50)
- DivfreeNP5 divider(NP5)

2.14.4.1 DivFree Divider Usage

Using this divider, freq_out =freq_src/(n+1).

2.14.4.2 Fractional Divider Usage

To get specific frequency, clocks of I2S, UART, MIPI and VICAP can be generated by fractional divider. Generally you should ensure that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock.

2.14.4.3 DivfreeDT50 Divider Usage

Some modules like EMMC, SDIO, SDMMC need clock of 50% duty cycle, divfreeDT50 can generate clock of 50% duty cycle even when divisor value is odd.

2.14.4.4 DivFreeNP5 Divider Usage

Some modules need special frequency can use this divider. Using this divider, freq_out = freq_src/(n+1.5).

2.14.5 Global Software Reset

Two global software resets are designed in the chip. These two software resets are self-de-asserted by hardware. Hold time of global software reset (glb_srstn_1, glb_srstn_2, wdt_rstn, tsadc_rstn) can be programed up to 1ms.

- CRU_GLB_SRST_FST_VALUE[15:0] = 0xfdb9 to assert the first global software reset glb_srstn_1
- CRU_GLB_SRST_SND_VALUE[15:0] = 0xeca8 to assert the second global software reset glb_srstn_2
- glb_srstn_1 resets almost all logic except some registers just supporting hardware reset
- glb_srstn_2 resets almost all logic except GRFs and GPIOs

Reset for IP in PD_PMU can be hold if its reset_hold_enable in PMUGRF_PMU_SOC_CON1 or PMUGRF_PMU_SOC_CON2 is high even if glb_srstn_1 or glb_srstn_2 active.

2.14.6 SSCG Usage

There are some scenes where SSCG should be enabled. One scene is in communication where a fixed frequency is required. Another scene is that the system requires a clock with

low long-term jitter. When SSCG is used, the PLL should be configured to fractional mode first for spread spectrum capability.

2.14.6.1 SSCG Use InternalPoint Table

User can use SSCG with internal point table as following steps:

- Setting ssmod_spread (CRU_XPLL_CON3[12:8]) and ssmod_downspread (CRU_XPLL_CON3[3])

The modulation amplitude is controlled by the value of ssmod_spread. A ssmod_spread value of 5'd0 turns off the modulation. A ssmod_spread value of 5'd31 (5'b11111) gives maximum modulation while a value of 5'd1 gives minimum modulation.
The modulation amplitude can be calculated from the value of modulation by:

Modulation Amplitude = ±5'd(ssmod_spread)*0.1%

The modulation direction is determined by the ssmod_downspread bit.

- ssmod_downspread = 1'b1, down spread mode is used. If ssmod_spread = 5'd29, the maximum PLL frequency is the nominally programmed value F_{NOM}, and the minimum value is given by F_{NOM}* (1–0.029).
- ssmod_downspread = 1'b0, center spread mode is used. If ssmod_spread = 5'd29, the maximum PLL frequency would be determined by F_{NOM}* (1+0.029) and the minimum frequency by F_{NOM} * (1–0.029).

Setting the style of modulation (center versus down) and the modulation amplitude depend on the amount of EMI reduction desired and the timing margin for circuits running on the spread clock domain. The larger the spread value, the greater the reduction in EMI amplitude. However, the larger the spread value, the more timing margin needed for correct circuit operation.

- Setting ssmod_sel_ext_wave (CRU_XPLL_CON4[0])=1'b0

When use internal point table, the frequency will change as following during 128 point:

- Change from minimum value to maximum value uniformly within 64 points
- Change from maximum value to minimum value uniformly within 64 points

Spread spectrum modulator is implemented by repeating as above.

- Setting ssmod_divval (CRU_XPLL_CON3[7:4])

The frequency of modulation FMOD= FREF / (Point number*REFDIV * ssmod_divval). The FMOD is typically set above 32KHz and below the maximum frequency for modulation fidelity, which isdetermined by the PLL bandwidth. The maximum modulation frequency is conservatively set at FREF / (200*REFDIV).
When FREF=24MHz and REFDIV= 1, the value of ssmod_divval can be 5, then the FMOD is 37.5KHz.

- Setting ssmod_bp(CRU_XPLL_CON3[0])=1'b0
- Setting ssmod_disable_sscg(CRU_XPLL_CON3[1])=1'b0
- Setting ssmod_reset(CRU_XPLL_CON3[2])=1'b0

2.14.6.2 SSCG Use External Point Table

In addition to the internal shape table, an external shape table can be used.
This enables customization tables in both shape and the number of sample points for theenvelope wave formup to 128 data points. The external table of 128 data points can be configured from CRU_SSCGTBL_CON0~CRU_SSCGTBL_CON31.
User can use SSMOD with external point table as following steps:

- Setting ssmod_spread(CRU_XPLL_CON3[12:8]) andssmod_downspread(CRU_XPLL_CON3[3]), same with internal point table usage.
- Setting ssmod_sel_ext_wave (CRU_XPLL_CON4[0])=1'b1
- Setting ssmod_ext_maxaddr(CRU_XPLL_CON4[15:8]) and table0 ~ table127 (CRU_SSCGTBL_CON0 ~ CRU_SSCGTBL_CON31)

ssmod_ext_maxaddr is the maximum table address. For example, if the number of points describing the envelope shape is 128, the ssmod_ext_maxaddr should be configured to 127. The table address circulate over the range 0 to 127.
The table0 ~ table127 must be 8 bit numbers in the form of sign and magnitude.

- 1.00 is represented by 8'b01111111, it is corresponded to maximum frequency.
- -1.00 is represented in the table by 8'b11111111, it is corresponded to minimum frequency.
- 0.5 is represented in the table by 8'b00111111.

- -0.5 is represented in the table by 8'b10111111.

The frequency will change base table0~table127 within 128 points, and then repeat.

- Setting ssmod_divval(CRU_XPLL_CON3[7:4])

The frequency of modulation FMOD= FREF/(Point number*REFDIV*ssmod_divval). The point number equals to ssmod_ext_maxaddr+1.

- Setting ssmod_bp(CRU_XPLL_CON3[0])=1'b0
- Setting ssmod_disable_sscg(CRU_XPLL_CON3[1])=1'b0
- Setting ssmod_reset(CRU_XPLL_CON3[2])=1'b0

2.14.7 BIU Clock gating reliance

A part of biu clocks have a dependence on another biu clock in order to sharing the internal bus. When these clocks are in use, another biu clock must be opened, and cannot be gated. These clocks and the special clock on which they are relied are as following:

Table 2-3 BIU Clocks dependency

Clocks which have dependency	The clock which can not be gated
pclk_cpu_biu	hclk_cpu_biu aclk_cpu_biu
hclk_cpu_biu	aclk_cpu_biu
hclk_npu_biu	hclk_cpu_biu
pclk_npu_biu	hclk_npu_biu
hclk_vi_biu	hclk_cpu_biu
pclk_vi_biu	hclk_vi_biu
pclk_vi_rtc_biu	hclk_vi_biu
pclk_ddr_biu	hclk_cpu_biu
hclk_peri_biu	hclk_cpu_biu
pclk_peri_biu	hclk_peri_biu
hclk_vo_biu	hclk_cpu_biu
pclk_vo_biu	hclk_vo_biu
aclk_vepu_cpm_biu	aclk_vo_biu
hclk_vepu_biu	hclk_cpu_biu
pclk_vepu_biu	hclk_cpu_biu
hclk_pmu_biu	hclk_cpu_biu
pclk_pmu_biu	hclk_pmu_biu
pclk_cru_biu	hclk_cpu_biu

Chapter 3 System Debug

3.1 Overview

The chip uses the DAPLITE Technology to support real-time debug.

3.1.1 Features

- Invasive debug with core halted
- SW-DP

3.1.2 Debug Components Address Map

The following table shows the debug components address in memory map:

Module	Base Address
DAP_ROM	0xff200000

3.2 Block Diagram

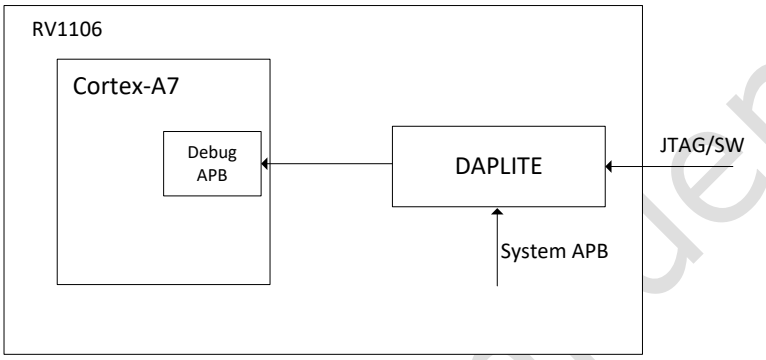


Fig. 3-1 Debug system structure

3.3 Function Description

3.3.1 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 document CoreSight_DAPLite_TRM.pdf for the debug detail description.

3.4 Register Description

Please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

3.5 Interface Description

3.5.1 DAP SWJ-DP Interface

The following figure is the DAP SWJ-DP interface, the SWJ-DP is a combined JTAG-DP and SW-DP that enable you connect either a Serial Wire Debug(SWJ) to JTAG probe to a target.

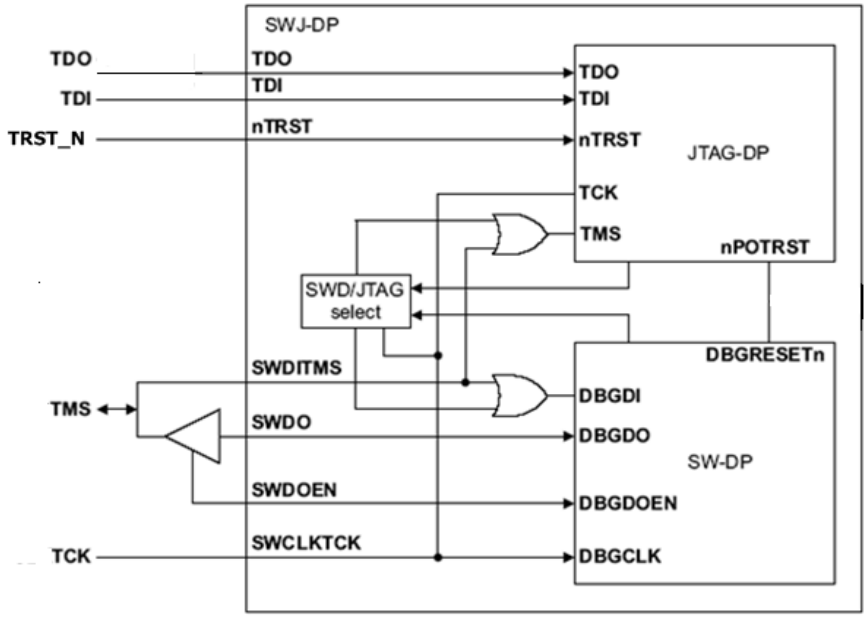


Fig. 3-2 DAP SWJ interface

3.5.2 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 serial wire interface, and the DAP internal bus transfers. It shows when the target responds with a WAIT acknowledgement.

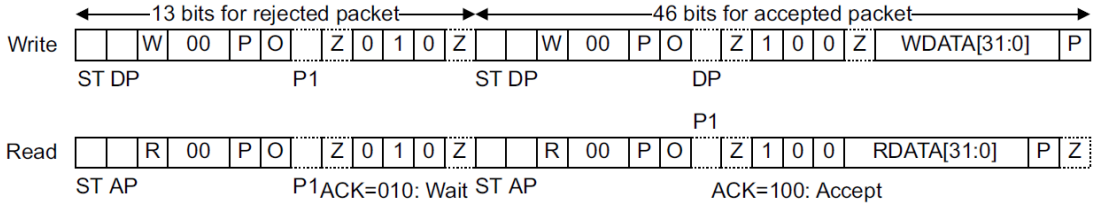


Fig. 3-3 SW-DP acknowledgement timing

Table 3-1 DAP-Lite Interface Description

Module pin	Direction	Pad name	IOMUX
jtag_tckm0	I	HPMCU_JTAG_TCK_M1/A7_JTAG_TCK_M0/UART5_RX_M0/SDMMC0_D2/GPIO3_A7_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[14:12] = 3'h3
jtag_tmsm0	I/O	HPMCU_JTAG_TMS_M1/A7_JTAG_TMS_M0/UART5_TX_M0/SDMMC0_D3/GPIO3_A6_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[11:8] = 3'h3
jtag_tckm1	I	LPMCU_JTAG_TCK_M0/HPMCU_JTAG_TCK_M0/UART2_TX_M1/A7_JTAG_TCK_M1/GPIO1_B2_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[11:8] = 3'h1
jtag_tmsm1	I/O	LPMCU_JTAG_TMS_M0/HPMCU_JTAG_TMS_M0/UART2_RX_M1/A7_JTAG_TMS_M1/GPIO1_B3_u	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[14:12] = 3'h1

By default, jtag_tckm0 and jtag_tmsm0 are forced to connect to corresponding io pad, we don't need to set IOMUX again.

Chapter 4 General Register Files (GRF)

4.1 Overview

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

4.2 Function Description

The function of general register file is:

- GPIO IOMUX control
- GPIO PAD control
- Common system control
- Record the system state

Table 4-1 GRF Address Mapping Table

Name	Address Base
PERI_GRF	0xFF000000
VENC_GRF	0xFF010000
NPU_GRF	0xFF018000
PMU_GRF	0xFF020000
DDR_GRF	0xFF030000
CORE_GRF	0xFF040000
VI_GRF	0xFF050000
VO_GRF	0xFF060000
GPIO0_IOC	0xFF388000
GPIO1_IOC	0xFF538000
GPIO2_IOC	0xFF548000
GPIO3_IOC	0xFF558000
GPIO4_IOC	0xFF568000

4.3 PERI_GRF Register Description

4.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PERI_GRF_PERI_CON0	0x0000	W	0x00000000	Peripheral system control register 0
PERI_GRF_PERI_CON1	0x0004	W	0x00002000	Peripheral system control register 1
PERI_GRF_PERI_STATUS	0x0010	W	0x00000000	Peripheral system status register
PERI_GRF_EMMC_CON0	0x0020	W	0x00000004	EMMC control register 0
PERI_GRF_EMMC_CON1	0x0024	W	0x00000000	EMMC control register 1
PERI_GRF_USBOTG_CON0	0x0030	W	0x00002000	USB OTG control register 0
PERI_GRF_USBOTG_CON1	0x0034	W	0x00000100	USB OTG control register 1
PERI_GRF_USBOTG_STATUS0	0x0040	W	0x00000000	USB OTG status register 0
PERI_GRF_USBOTG_STATUS1	0x0044	W	0x00000000	USB OTG status register 1
PERI_GRF_USBOTG_STATUS2	0x0048	W	0x00000000	USB OTG status register 2
PERI_GRF_USBPHY_CON0	0x0050	W	0x00000C52	USB PHY control register 0
PERI_GRF_USBPHY_CON1	0x0054	W	0x00000080	USB PHY control register 1
PERI_GRF_USBPHY_CON2	0x0058	W	0x00000000	USB PHY control register 2

Name	Offset	Size	Reset Value	Description
PERI_GRF_USBPHY_CON3	0x005C	W	0x00000000	USB PHY control register 3
PERI_GRF_USBPHY_STATUS	0x0060	W	0x00000000	USB PHY status register
PERI_GRF_SARADC_CON	0x0070	W	0x00000000	SARADC control register
PERI_GRF_PERI_MEM_CON	0x0080	W	0x00001015	Peripheral system memory control register
PERI_GRF_BIU_CON0	0x0090	W	0x00000000	BIU control register 0
PERI_GRF_BIU_CON1	0x0094	W	0x00000000	BIU control register 1
PERI_GRF_BIU_STATUS0	0x0098	W	0x00000000	BIU status register 0
PERI_GRF_BIU_STATUS1	0x009C	W	0x00000000	BIU status register 1
PERI_GRF_USBOTG_SIG_DETECT_CON	0x0100	W	0x00000000	USB OTG detection control register
PERI_GRF_USBOTG_SIG_DETECT_STATUS	0x0104	W	0x00000000	USB OTG detection status register
PERI_GRF_USBOTG_SIG_DETECT_CLR	0x0108	W	0x00000000	USB OTG detection clear register
PERI_GRF_USBOTG_LINESTATE_DETECT_CON	0x0110	W	0x00030100	USB OTG linestate control register
PERI_GRF_USBOTG_DISCONNECT_DETECT_CON	0x0114	W	0x00030100	USB OTG disconnect control register
PERI_GRF_USBOTG_BVALID_DETECT_CON	0x0118	W	0x00030100	USB OTG bvalid control register
PERI_GRF_USBOTG_ID_DETECT_CON	0x011C	W	0x00030100	USB OTG id control register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.3.2 Detail Registers Description

PERI_GRF_PERI_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	uart5_cts_inv Polarity selection for uart5_cts. 1'b0: Low active 1'b1: High active
10	RW	0x0	uart5_rts_inv Polarity selection for uart5_rts. 1'b0: Low active 1'b1: High active
9	RW	0x0	uart4_cts_inv Polarity selection for uart4_cts. 1'b0: Low active 1'b1: High active
8	RW	0x0	uart4_rts_inv Polarity selection for uart4_rts. 1'b0: Low active 1'b1: High active

Bit	Attr	Reset Value	Description
7	RW	0x0	uart3_cts_inv Polarity selection for uart3_cts. 1'b0: Low active 1'b1: High active
6	RW	0x0	uart3_rts_inv Polarity selection for uart3_rts. 1'b0: Low active 1'b1: High active
5	RW	0x0	uart2_cts_inv Polarity selection for uart2_cts. 1'b0: Low active 1'b1: High active
4	RW	0x0	uart2_rts_inv Polarity selection for uart2_rts. 1'b0: Low active 1'b1: High active
3	RW	0x0	uart1_cts_inv Polarity selection for uart1_cts. 1'b0: Low active 1'b1: High active
2	RW	0x0	uart1_rts_inv Polarity selection for uart1_rts. 1'b0: Low active 1'b1: High active
1	RW	0x0	uart0_cts_inv Polarity selection for uart0_cts. 1'b0: Low active 1'b1: High active
0	RW	0x0	uart0_rts_inv Polarity selection for uart0_rts. 1'b0: Low active 1'b1: High active

PERI_GRF_PERI_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	wdt_ns_pause_en No-secure WDT pause enable. 1'b0: Disable 1'b1: Enable
13	RW	0x1	emmc_ram_clkgat_disable EMMC ram clock gating control. 1'b0: Clock gating enable 1'b1: Clock gating disable
12	RW	0x0	trng_ns_rst_sync_bypass Enable No-secure TRNG reset synchronization. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
11	RW	0x0	trng_ns_ctrl_reseed Enable No-secure TRNG reseed operation. 1'b0: Disable 1'b1: Enable When it is asserted, the No-secure TRNG responds by stopping any reseed operation that is currently underway and initiating a new reseed operation.
10	RW	0x0	trng_ns_ctrl_zeroize Enable No-secure TRNG zeroize operation. 1'b0: Disable 1'b1: Enable
9	RW	0x0	wdt_ns_glb_reset_en No-secure WDT global reset enable. 1'b0: Disable. No-secure WDT reset output cannot reset SoC system. 1'b1: Enable. No-secure WDT reset output will reset SoC system.
8	RW	0x0	dsm_ena DSM enable. 1'b0: Disable 1'b1: Enable. Audio controller works with DSM.
7	RW	0x0	acodec_ad2da_loop ACODEC ad2da_loop enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	acodec_ena ACODEC enable. 1'b0: Disable. Audio controller works with external ACODEC through PAD. 1'b1: Enable. Audio controller works with internal ACODEC.
5	RW	0x0	i2s_mclk_oe_n i2s_mclk output enable. 1'b0: Output disable 1'b1: Output enable
4	RW	0x0	i2s_sdo3_oe_n i2s_sdo3 output enable. 1'b0: Output disable 1'b1: Output enable
3	RW	0x0	i2s_sdo2_oe_n i2s_sdo2 output enable. 1'b0: Output disable 1'b1: Output enable
2	RW	0x0	i2s_sdo1_oe_n i2s_sdo1 output enable. 1'b0: Output disable 1'b1: Output enable
1	RW	0x0	audio_tx_rx_sel Audio transmit and receive selection. 1'b0: Transmit is selected 1'b1: Receive is selected
0	RW	0x0	i2s_sai_sel Audio controller selection. 1'b0: I2S is selected as audio controller 1'b1: SAI selected as audio controller

PERI_GRF_PERI_STATUS

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RO	0x0	acodec_dac_master_en ACODEC DAC master enable status. 1'b0: Inactive 1'b1: Active
20	RO	0x0	acodec_adc_master_en ACODEC ADC master enable status. 1'b0: Inactive 1'b1: Active
19	RO	0x0	trng_s_ctrl_rand_vld Secure TRNG serial random bit valid output. 1'b0: Inactive 1'b1: Active
18	RO	0x0	trng_s_ctrl_rand_bit Secure TRNG serial random bit output. 1'b0: Inactive 1'b1: Active
17	RO	0x0	trng_s_ctrl_seeded Secure TRNG reseeding/seeding completion output. 1'b0: Inactive 1'b1: Active
16	RO	0x0	trng_s_ctrl_reseeding Secure TRNG reseeding activity output. 1'b0: Inactive 1'b1: Active
15	RO	0x0	trng_s_ctrl_reminder Secure TRNG reseed reminder output. 1'b0: Inactive 1'b1: Active
14	RO	0x0	trng_s_ctrl_secure Secure TRNG secure mode output. 1'b0: Inactive 1'b1: Active
13	RO	0x0	trng_ns_ctrl_rand_vld No-secure TRNG serial random bit valid output. 1'b0: Inactive 1'b1: Active
12	RO	0x0	trng_ns_ctrl_rand_bit No-secure TRNG serial random bit output. 1'b0: Inactive 1'b1: Active
11	RO	0x0	trng_ns_ctrl_seeded No-secure TRNG reseeding/seeding completion output. 1'b0: Inactive 1'b1: Active
10	RO	0x0	trng_ns_ctrl_reseeding No-secure TRNG reseeding activity output. 1'b0: Inactive 1'b1: Active
9	RO	0x0	trng_ns_ctrl_reminder No-secure TRNG reseed reminder output. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
8	RO	0x0	trng_ns_ctrl_secure No-secure TRNG secure mode output. 1'b0: Inactive 1'b1: Active
7	RO	0x0	stimer1_en_status Secure TIMER1 active status. 1'b0: Inactive 1'b1: Active
6	RO	0x0	stimer0_en_status Secure TIMER0 active status. 1'b0: Inactive 1'b1: Active
5	RO	0x0	timer5_en_status TIMER5 active status. 1'b0: Inactive 1'b1: Active
4	RO	0x0	timer4_en_status TIMER4 active status. 1'b0: Inactive 1'b1: Active
3	RO	0x0	timer3_en_status TIMER3 active status. 1'b0: Inactive 1'b1: Active
2	RO	0x0	timer2_en_status TIMER2 active status. 1'b0: Inactive 1'b1: Active
1	RO	0x0	timer1_en_status TIMER1 active status. 1'b0: Inactive 1'b1: Active
0	RO	0x0	timer0_en_status TIMER0 active status. 1'b0: Inactive 1'b1: Active

PERI_GRF_EMMC_CON0

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	test_clkout_0_sel EMMC test clock 0 output selection. 1'b0: Drive clock output to test clock 0 1'b1: Function clock output to test clock 0
11	RW	0x0	drv_sel Selection for drive clock. 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.

Bit	Attr	Reset Value	Description
10:3	RW	0x00	drv_delaynum Delay element number configuration for drive clock. It can be modified when init_state=1 and should be stable when init_state=0.
2:1	RW	0x2	drv_degree Phase selection for drive clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
0	RW	0x0	init_state Enable initialization for clock source. 1'b0: Disable 1'b1: Enable When init_state=1, the host clocks including drive clock and sample clock are inactive.

PERI_GRF_EMMC_CON1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	test_clkout_1_sel EMMC test clock 1 output selection. 1'b0: Sample clock output to test clock 1 1'b1: Function clock output to test clock 1
11	RW	0x0	sample_sel Selection for sample clock. 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.
10:3	RW	0x00	sample_delaynum Delay element number configuration for sample clock. It can be modified when init_state=1 and should be stable when init_state=0.
2:1	RW	0x0	sample_degree Phase selection for sample clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
0	RO	0x0	reserved

PERI_GRF_USBOTG_CON0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	usbotg_host_u2_port_disable USB2.0 port disable control. 1'b0: Port enabled 1'b1: Port disabled, stops reporting connect/disconnect events the port and keeps the port in disabled state.
14	RW	0x0	usbotg_host_port_power_control_present This indicates whether the host controller implementation includes port power control. 1'b0: Indicates that the port does not have port power switches 1'b1: Indicates that the port has port power switches
13:8	RW	0x20	usbotg_fladj_30mhz_reg USB OTG fladj_30mhz_reg configuration
7	RO	0x0	reserved
6	RW	0x0	usbotg_hub_port_perm_attach Indicates if the device attached to a downstream port is permanently attached or not. 1'b0: Not permanently attached 1'b1: Permanently attached
5	RO	0x0	reserved
4	RW	0x0	usbotg_hub_port_overcurrent Overcurrent indication of the root-hub ports. 1'b0: No overcurrent 1'b1: overcurrent
3:0	RW	0x0	usbotg_bus_filter_bypass The function of each bit is: usbotg_bus_filter_bypass[3]: Bypass the filter for utmiotg_iddig usbotg_bus_filter_bypass[2]: Bypass the filters for utmisrp_bvalid and utmisrp_sessend usbotg_bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports usbotg_bus_filter_bypass[0]: Bypass the filter for utmiotg_vbusvalid all U2 ports In non-OTG Host-only mode, internal bus filters are not needed. Values: 1'b0: Bus filter(s) enabled 1'b1: Bus filter(s) disabled It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the core.

PERI_GRF_USBOTG_CON1

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:8	RW	0x1	usbotg_host_num_u2_port xHCI host USB2 Port number, default as 1.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	usbotg_host_legacy_smi_bar Use this register to support SMI on BAR defined in xHCI spec. SW must set this register, then clear this register to indicate Base Address Register written.
4	RW	0x0	usbotg_host_legacy_smi_pci_cmd Use this register to support SMI on PCI Command defined in xHCI spec. SW must set this register, then clear this register to indicate PCI command register written.
3:1	RO	0x0	reserved
0	RW	0x0	usbotg_pme_en Enable the core to assert pme_generation. 1'b0: Disable 1'b1: Enable

PERI_GRF_USBOTG_STATUS0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	usbotg_logic_analyzer_trace_31_0 usbotg_logic_analyzer_trace bit[31:0] status

PERI_GRF_USBOTG_STATUS1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	usbotg_logic_analyzer_trace_63_32 usbotg_logic_analyzer_trace bit[63:32] status

PERI_GRF_USBOTG_STATUS2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RO	0x000	usbotg_host_current_belt usbotg_host_current_belt bit[11:0] status

PERI_GRF_USBPHY_CON0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	usbotg_utmi_dischrgvbus usbotg_utmi_dischrgvbus configuration.
12	RW	0x0	usbotg_utmi_chrgvbus usbotg_utmi_chrgvbus configuration.
11	RW	0x1	usbotg_utmi_idpullup usbotg_utmi_idpullup configuration.
10	RW	0x1	usbotg_utmi_iddig Software configuration for usbotg_utmi_iddig.
9	RW	0x0	usbotg_utmi_iddig_sel usbotg_utmi_iddig selection. 1'b0: usbotg_utmi_iddig comes from USB PHY 1'b1: usbotg_utmi_iddig comes from USBPHY_CON0[10]

Bit	Attr	Reset Value	Description
8	RW	0x0	usbotg_utmi_dmpulldown Software configuration for usbotg_utmi_dmpulldown when usbphy_sw_en is high.
7	RW	0x0	usbotg_utmi_dppulldown Software configuration for usbotg_utmi_dppulldown when usbphy_sw_en is high.
6	RW	0x1	usbotg_utmi_termselect Software configuration for usbotg_utmi_termselect when usbphy_sw_en is high.
5:4	RW	0x1	usbotg_utmi_xcvrselect Software configuration for usbotg_utmi_xcvrselect when usbphy_sw_en is high.
3:2	RW	0x0	usbotg_utmi_opmode Software configuration for usbotg_utmi_opmode when usbphy_sw_en is high.
1	RW	0x1	usbotg_utmi_suspend_n Software configuration for usbotg_utmi_suspend_n when usbphy_sw_en is high.
0	RW	0x0	usbphy_sw_en USB PHY software configuration enable. 1'b0: Disable 1'b1: Enable

PERI_GRF_USBPHY_CON1

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	usbotg_utmi_drvvbus Software configuration for usbotg_utmi_drvvbus.
10	RW	0x0	usbotg_utmi_drvvbus_sel usbotg_utmi_drvvbus selection. 1'b0: usbotg_utmi_drvvbus comes from USB PHY 1'b1: usbotg_utmi_drvvbus comes from PERI_GRF_USBPHY_CON1[11]
9	RW	0x0	usbotg_utmi_fs_se0 Software configuration for usbotg_utmi_fs_se0.
8	RW	0x0	usbotg_utmi_fs_data Software configuration for usbotg_utmi_fs_data.
7	RW	0x1	usbotg_utmi_fs_oe Software configuration for usbotg_utmi_fs_oe.
6	RW	0x0	usbotg_utmi_fs_xver_own Software configuration for usbotg_utmi_fs_xver_own.
5:3	RO	0x0	reserved
2	RW	0x0	usbotg_utmi_suspend_n Software configuration for usbotg_utmi_suspend_n.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	usbotg_utmi_suspend_n_sel usbotg_utmi_suspend_n selection. 2'b00: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b01: PERI_GRF_USBPHY_CON1[2] 2'b10: ~usbotg_utmi_suspend_com_n & ~usbotg_utmi_l1_suspend_com_n 2'b11: ~usbotg_uspend_n & ~usbotg_utmi_l1_suspend_n

PERI_GRF_USBPHY_CON2

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	usbotg_utmi_bvalid_sel usbotg_utmi_bvalid selection. 1'b0: usbotg_utmi_bvalid comes from USB PHY 1'b1: usbotg_utmi_bvalid comes from PERI_GRF_USBPHY_CON2[14]
14	RW	0x0	usbotg_utmi_bvalid Software configuration for usbotg_utmi_bvalid.
13	RO	0x0	reserved
12	RW	0x0	usbphy_vdm_src_en Software configuration for usbphy_vdm_src_en.
11	RW	0x0	usbphy_vdp_src_en Software configuration for usbphy_vdp_src_en.
10	RW	0x0	usbphy_rdm_pdwn_en Software configuration for usbphy_rdm_pdwn_en.
9	RW	0x0	usbphy_idp_src_en Software configuration for usbphy_idp_src_en.
8	RW	0x0	usbphy_idm_sink_en Software configuration for usbphy_idm_sink_en.
7	RW	0x0	usbphy_idp_sink_en Software configuration for usbphy_idp_sink_en.
6:0	RO	0x00	reserved

PERI_GRF_USBPHY_CON3

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9:8	RW	0x0	usbphy_bc_weakpulldown_tune Software configuration for usbphy_bc_weakpulldown_tune.
7:6	RW	0x0	usbphy_bc_weakpullup_tune Software configuration for usbphy_bc_weakpullup_tune.
5	RW	0x0	usbphy_bc_weakpulldown_en Software configuration for usbphy_bc_weakpulldown_en.
4	RW	0x0	usbphy_bc_weakpullup_en Software configuration for usbphy_bc_weakpullup_en.

Bit	Attr	Reset Value	Description
3	RW	0x0	usbphy_force_dcp_det Software configuration for usbphy_force_dcp_det.
2	RW	0x0	usbphy_force_cp_det Software configuration for usbphy_force_cp_det.
1	RW	0x0	usbphy_force_dp_attached Software configuration for usbphy_force_dp_attached.
0	RW	0x0	usbphy_charge_det_byp Software configuration for usbphy_charge_det_byp.

PERI_GRF_USBPHY_STATUS

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RO	0x0	usbphy_dp_floatdet usbphy_dp_floatdet status, high active.
15	RO	0x0	usbphy_dm_floatdet usbphy_dm_floatdet status, high active.
14	RO	0x0	usbphy_dp_attached usbphy_dp_attached status, high active.
13	RO	0x0	usbphy_cp_detected usbphy_cp_detected status, high active.
12	RO	0x0	usbphy_dcp_detected usbphy_dcp_detected status, high active.
11	RO	0x0	usbotg_phy_ls_fs_rcv usbotg_phy_ls_fs_rcv status, high active.
10	RO	0x0	usbotg_utmi_avalid usbotg_utmi_avalid status, high active.
9	RO	0x0	usbotg_utmi_bvalid usbotg_utmi_bvalid status, high active.
8	RO	0x0	usbotg_utmi_fs_xver_own usbotg_utmi_fs_xver_own status, high active.
7	RO	0x0	usbotg_utmi_hostdisconnect usbotg_utmi_hostdisconnect status, high active.
6	RO	0x0	usbotg_utmi_iddig usbotg_utmi_iddig status, high active.
5:4	RO	0x0	usbotg_utmi_linestate usbotg_utmi_linestate status, high active.
3	RO	0x0	usbotg_utmi_sessend usbotg_utmi_sessend status, high active.
2	RO	0x0	usbotg_utmi_vbusvalid usbotg_utmi_vbusvalid status, high active.
1	RO	0x0	usbotg_utmi_vmi usbotg_utmi_vmi status, high active.
0	RO	0x0	usbotg_utmi_vpi usbotg_utmi_vpi status, high active.

PERI_GRF_SARADC_CON

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	saradc_ana_reg SARADC ana_reg configuration

PERI_GRF_PERI_MEM_CON

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x1	peri_mem_cfg_uhdspra [1:0]: RTSEL [3:2]: WTSEL
11:6	RW	0x00	peri_mem_cfg_uhddpra [1:0]: RTSEL [3:2]: WTSEL [5:4]: PTSEL
5:0	RW	0x15	peri_mem_cfg_rom [1:0]: RTSEL [3:2]: PTSEL [5:4]: TRB

PERI_GRF_BIU_CON0

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	vo_to_vepu_stall Response type when BIU_VO or BIU_VEPU in idle state and VO access VEPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
14	RW	0x0	vi_to_peri_stall Response type when BIU_VI or BIU_PERI in idle state and VI access PERI. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
13	RW	0x0	vepu_to_cru_stall Response type when BIU_VEPU or BIU_CRU in idle state and VEPU access CRU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
12	RW	0x0	vepu_to_ddr_stall Response type when BIU_VEPU or BIU_DDR in idle state and VEPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
11	RW	0x0	shrm_to_npu_stall Response type when BIU_SHRM or BIU_NPU in idle state and SHRM access NPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
10	RW	0x0	peri_to_ddr_stall Response type when BIU_PERI or BIU_DDR in idle state and PERI access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted

Bit	Attr	Reset Value	Description
9	RW	0x0	npu_to_vepu_stall Response type when BIU_NPU or BIU_VEPU in idle state and NPU access VEPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
8	RW	0x0	cpu_to_ddr_reg_stall Response type when BIU_CPU or BIU_DDR_REG in idle state and CPU access DDR_REG. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
7	RW	0x0	npu_to_ddr_stall Response type when BIU_NPU or BIU_DDR in idle state and NPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
6	RW	0x0	ddr_to_shrm_stall Response type when BIU_SHRM or BIU_DDR in idle state and DDR access SHRM. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
5	RW	0x0	ddr_to_peri_stall Response type when BIU_DDR or BIU_PERI in idle state and DDR access PERI. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
4	RW	0x0	cru_to_pmu_stall Response type when BIU_CRU or BIU_PMU in idle state and CRU access PMU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
3	RW	0x0	cpu_to_vi_stall Response type when BIU_CPU or BIU_VI in idle state and CPU access VI. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
2	RW	0x0	cpu_to_vo_stall Response type when BIU_CPU or BIU_VO in idle state and CPU access VO. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
1	RW	0x0	cpu_to_npu_stall Response type when BIU_CPU or BIU_NPU in idle state and CPU access NPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
0	RW	0x0	cpu_to_ddr_stall Response type when BIU_CPU or BIU_DDR in idle state and CPU access DDR. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted

PERI_GRF_BIU_CON1

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	cpu_to_npu_srv_stall Response type when BIU_CPU or BIU_NPU in idle state and CPU access NPU_SRV. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
2	RW	0x0	dma_to_other_slv_stall Response type when DMA access other slave. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
1	RW	0x0	vepupp_vepu_stall Response type when BIU_VEPU_PP or BIU_VEPU in idle state and VEPU_PP access VEPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted
0	RW	0x0	venc_to_vepu_stall Response type when BIU_VENC or BIU_VEPU in idle state and VENC access VEPU. 1'b0: BIU return ERROR response 1'b1: BIU hold the bus until the NIU idle state is de-asserted

PERI_GRF_BIU_STATUS0

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18:0	RO	0x00000	active_status BIU access active status, high active. [0]: cpu_to_ddr_active [1]: cpu_to_npu_active [2]: cpu_to_vo_active [3]: cpu_to_vi_active [4]: cru_to_pmu_active [5]: ddr_to_peri_active [6]: ddr_to_shrm_active [7]: npu_to_ddr_active [8]: cpu_to_ddr_reg_active [9]: npu_to_vepu_active [10]: peri_to_ddr_active [11]: shrm_to_npu_active [12]: vepu_to_ddr_active [13]: vepu_to_cru_active [14]: vi_to_peri_active [15]: vo_to_vepu_active [16]: venc_to_vepu_active [17]: vepupp_vepu_active [18]: dma_to_other_slv_active

PERI_GRF_BIU_STATUS1

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
15	RO	0x0	decom_nopendingtrans No pending transfer status for DECOM. 1'b0: Inactive 1'b1: Active
14	RO	0x0	usbotg_nopendingtrans No pending transfer status for USBOTG. 1'b0: Inactive 1'b1: Active
13	RO	0x0	dmac_nopendingtrans No pending transfer status for DMAC. 1'b0: Inactive 1'b1: Active
12	RO	0x0	crypto_nopendingtrans No pending transfer status for CRYPTO. 1'b0: Inactive 1'b1: Active
11	RO	0x0	ive_wr_nopendingtrans No pending transfer status for IVE_WR. 1'b0: Inactive 1'b1: Active
10	RO	0x0	ive_rd_nopendingtrans No pending transfer status for IVE_RD. 1'b0: Inactive 1'b1: Active
9	RO	0x0	vop_nopendingtrans No pending transfer status for VOP. 1'b0: Inactive 1'b1: Active
8	RO	0x0	vicap_nopendingtrans No pending transfer status for VICAP. 1'b0: Inactive 1'b1: Active
7	RO	0x0	vepu_pp_nopendingtrans No pending transfer status for VEPU_PP. 1'b0: Inactive 1'b1: Active
6	RO	0x0	venc_nopendingtrans No pending transfer status for VENC. 1'b0: Inactive 1'b1: Active
5	RO	0x0	rga_wr_nopendingtrans No pending transfer status for RGA_WR. 1'b0: Inactive 1'b1: Active
4	RO	0x0	rga_rd_nopendingtrans No pending transfer status for RGA_RD. 1'b0: Inactive 1'b1: Active
3	RO	0x0	npu_nopendingtrans No pending transfer status for NPU. 1'b0: Inactive 1'b1: Active
2	RO	0x0	mac_nopendingtrans No pending transfer status for GMAC. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
1	RO	0x0	isp_nopendingtrans No pending transfer status for ISP. 1'b0: Inactive 1'b1: Active
0	RO	0x0	cpu_nopendingtrans No pending transfer status for CPU. 1'b0: Inactive 1'b1: Active

PERI_GRF_USBOTG_SIG_DETECT_CON

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	otg_disconnect_fall_irq_en otg_disconnect falling edge irq enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	otg_disconnect_rise_irq_en otg_disconnect rising edge irq enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	otg_id_fall_irq_en otg_id falling edge irq enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	otg_id_rise_irq_en otg_id rising edge irq enable. 1'b0: Disable 1'b1: Enable
3	RW	0x0	otg_bvalid_fall_irq_en otg_bvalid falling edge irq enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	otg_bvalid_rise_irq_en otg_bvalid rising edge irq enable. 1'b0: Disable 1'b1: Enable
1	RO	0x0	reserved
0	RW	0x0	otg_linestate_irq_en otg_linestate change irq enable. 1'b0: Disable 1'b1: Enable

PERI_GRF_USBOTG_SIG_DETECT_STATUS

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	otg_disconnect_fall_irq otg_disconnect falling edge irq status. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
6	RO	0x0	otg_disconnect_rise_irq otg_disconnect rising edge irq status. 1'b0: Inactive 1'b1: Active
5	RO	0x0	otg_id_fall_irq otg_id falling edge irq status. 1'b0: Inactive 1'b1: Active
4	RO	0x0	otg_id_rise_irq otg_id rising edge irq status. 1'b0: Inactive 1'b1: Active
3	RO	0x0	otg_bvalid_fall_irq otg_bvalid falling edge irq status. 1'b0: Inactive 1'b1: Active
2	RO	0x0	otg_bvalid_rise_irq otg_bvalid rising edge irq status. 1'b0: Inactive 1'b1: Active
1	RO	0x0	reserved
0	RO	0x0	otg_linestate_irq otg_linestate change irq status. 1'b0: Inactive 1'b1: Active

PERI_GRF_USBOTG_SIG_DETECT_CLR

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	otg_disconnect_fall_irq_clr otg_disconnect falling edge irq clear. Write 1 to clear irq status.
6	RW	0x0	otg_disconnect_rise_irq_clr otg_disconnect rising edge irq clear. Write 1 to clear irq status.
5	RW	0x0	otg_id_fall_irq_clr otg_id falling edge irq clear. Write 1 to clear irq status.
4	RW	0x0	otg_id_rise_irq_clr otg_id rising edge irq clear. Write 1 to clear irq status.
3	RW	0x0	otg_bvalid_fall_irq_clr otg_bvalid falling edge irq clear. Write 1 to clear irq status.
2	RW	0x0	otg_bvalid_rise_irq_clr otg_bvalid rising edge irq clear. Write 1 to clear irq status.
1	RO	0x0	reserved
0	WO	0x0	otg_linestate_irq_clr otg_linestate change irq clear. Write 1 to clear irq status.

PERI_GRF_USBOTG_LINESTATE_DETECT_CON

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved

Bit	Attr	Reset Value	Description
19:0	RW	0x30100	linestate_detect_con OTG linestate filter time control register

PERI_GRF_USBOTG_DISCONNECT_DETECT_CON

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x30100	disconnect_detect_con OTG disconnect filter time control register

PERI_GRF_USBOTG_BVALID_DETECT_CON

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x30100	bvalid_detect_con OTG bvalid filter time control register

PERI_GRF_USBOTG_ID_DETECT_CON

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:0	RW	0x0030100	id_detect_con OTG id filter time control register

4.4 VENC_GRF Register Description

4.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VENC_GRF_VENC_MEMCFG_UHDSPRA	0x0000	W	0x00000001	
VENC_GRF_UART2RX_LOW_DLY	0x0004	W	0xFFFFFFFF	
VENC_GRF_CIF_IO_WRAPPER	0x0008	W	0x00000000	
VENC_GRF_VOP_IO_WRAPPER	0x000C	W	0x00000000	
VENC_GRF_VCCIO6_VD_STATUS	0x0010	W	0x00000000	

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.4.2 Detail Registers Description

VENC_GRF_VENC_MEMCFG_UHDSPRA

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x0	wtsel Timing adjustment setting for debug purpose
1:0	RW	0x1	rtsel Timing adjustment setting for debug purpose

VENC_GRF_UART2RX_LOW_DLY

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	uart2rx_low_dly uart2_rx low count

VENC_GRF_CIF_IO_WRAPPER

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	cif_clk_inv_sel_m1 cif invert clk select for m1

VENC_GRF_VOP_IO_WRAPPER

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x0	lcd_clk_inv_sel Lcd invert clk select
1:0	RW	0x3	io_bypass_sel 2'b00: Use two flip-flop pipe in VOP LCD IO in vccio6. 2'b11: Bypass two flip-flop pipe in VOP LCD IO in vccio6. Other: Reserved

VENC_GRF_VCCIO6_VD_STATUS

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	vccio6_vd Indicates vccio6 voltage status, vccio6_vd = 1 when VCCIO > 2.19v. vccio6_vd = 0 when VCCIO < 2.02v.

4.5 NPU_GRF Register Description

4.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
NPUGRF_NPU_MEM	0x0000	W	0x00000001	NPU memory configuration
NPUGRF_NPU_QOS	0x0004	W	0x00000000	NPU Qos configuration

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.5.2 Detail Registers Description

NPUGRF_NPU_MEM

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x0	wtesl Configuration for wtsel for memory for RKNN.
1:0	RW	0x1	rtsel Configuration for rtsel for memory for RKNN.

NPUGRF_NPU_QOS

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
2:0	RW	0x0	aw_qos Aw Qos for RKNN.

4.6 PMU_GRF Register Description

4.6.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_GRF_SOC_CON0	0x0000	W	0x0000301C	PMU System control register 0
PMU_GRF_SOC_CON1	0x0004	W	0x00000000	PMU System control register 1
PMU_GRF_SOC_CON2	0x0008	W	0x00000000	PMU System control register 2
PMU_GRF_SOC_CON3	0x000C	W	0x00000000	PMU System control register 3
PMU_GRF_SOC_CON4	0x0010	W	0x00000000	PMU System control register 4
PMU_GRF_SOC_CON5	0x0014	W	0x00000000	PMU System control register 5
PMU_GRF_SOC_STATUS	0x0020	W	0x00000012	PMU System status register
PMU_GRF_PMUIO_CON	0x0030	W	0x00000004	PMUIO voltage control register
PMU_GRF_MEM_CON	0x0040	W	0x00000001	PMU System memory control register
PMU_GRF_OS_REG0	0x0200	W	0x00000000	OS register 0
PMU_GRF_OS_REG1	0x0204	W	0x00000000	OS register 1
PMU_GRF_OS_REG2	0x0208	W	0x00000000	OS register 2
PMU_GRF_OS_REG3	0x020C	W	0x00000000	OS register 3
PMU_GRF_OS_REG4	0x0210	W	0x00000000	OS register 4
PMU_GRF_OS_REG5	0x0214	W	0x00000000	OS register 5
PMU_GRF_OS_REG6	0x0218	W	0x00000000	OS register 6
PMU_GRF_OS_REG7	0x021C	W	0x00000000	OS register 7
PMU_GRF_OS_REG8	0x0220	W	0x00000000	OS register 9
PMU_GRF_OS_REG9	0x0224	W	0x00000000	OS register 9
PMU_GRF_OS_REG10	0x0228	W	0x00000000	OS register 10
PMU_GRF_OS_REG11	0x022C	W	0x00000000	OS register 11
PMU_GRF_RSTFUNC_STATUS	0x0230	W	0x00000000	System reset status register
PMU_GRF_RSTFUNC_CLR	0x0234	W	0x00000000	System reset status clear register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.6.2 Detail Registers Description

PMU_GRF_SOC_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	upctl_a_sysreq_cfg Enable for DDRCTL AXI hardware low-power request by software. 1'b0: Disable 1'b1: Enable
13	RW	0x1	ddrc_a_gating_en Enable DDRCTL's AXI-clock auto clock gating for DDR fail safe. 1'b0: Disable 1'b1: Enable
12	RW	0x1	sref_a_enter_en Enable DDR self-refresh mode for AXI-clock domain by hardware for DDR fail safe. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
11	RW	0x0	ddrphy_bufferen_sel1 DDR PHY bufferen selection 1. 1'b0: bufferen determined by ddrphy_bufferen_sel0 1'b1: bufferen comes from hwlp_ddrphy_bufferen
10	RW	0x0	ddrphy_bufferen_core 1'b0: Enable DDRPHY io retention 1'b1: Disable DDRPHY io retention
9	RW	0x0	ddrphy_bufferen_sel0 DDRPHY bufferen selection 0. 1'b0: bufferen comes from PMU or DDR_FAIL_SAFE 1'b1: bufferen comes from ddrphy_bufferen_core
8	RW	0x0	upctl_c_sysreq_cfg Enable for DDRCTL hardware low-power clock request by software. 1'b0: Disable 1'b1: Enable
7	RO	0x0	reserved
6	RW	0x0	ddr_io_ret_cfg 1'b0: DDR IO retention managed by hardware automatically 1'b1: Enable DDR IO retention manually
5	RW	0x0	ddr_io_ret_de_req Enable DDR IO exit retention mode by hardware for DDR fail safe. 1'b0: Disable 1'b1: Enable
4	RW	0x1	ddrc_c_gating_en Enable DDRCTL's core-clock auto clock gating for DDR fail safe. 1'b0: Disable 1'b1: Enable
3	RW	0x1	sref_c_enter_en Enable DDR self-refresh mode for core-clock domain by hardware for DDR fail safe. 1'b0: Disable 1'b1: Enable
2	RW	0x1	ddrio_ret_en Enable DDR IO enter retention mode by hardware for DDR fail safe. 1'b0: Disable 1'b1: Enable
1	RW	0x0	wdt_reset_trigger_en Enable WDT reset trigger for DDR fail safe. 1'b0: Disable 1'b1: Enable
0	RW	0x0	tsadc_shut_reset_trigger_en Enable TSADC shut reset trigger for DDR fail safe. 1'b0: Disable 1'b1: Enable

PMU_GRF_SOC_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RW	0x0	rtc_clk_32k_h_clamp_n Clamp enable for rtc_clk_32k_h from RTC high drive path. 1'b0: Clamp enable, rtc_clk_32k_h is inactive to PMU system. 1'b1: Clamp disable, rtc_clk_32k_h is active to PMU system. When RTC is off, this bit should be set to 0.
14	RW	0x0	rtc_32k_sel 32K clock source selection. 1'b0: 32K clock from RTC nomal path 1'b1: 32K clock from RTC high drive path
13	RW	0x0	pmuwdt_mcu_reset_en PMU WDT reset LPMCU enable. 1'b0: Disable. PMU WDT reset output cannot reset LPMCU. 1'b1: Enable. PMU WDT reset output will just reset LPMCU.
12	RW	0x0	pmuwdt_glb_reset_en PMU WDT global reset enable. 1'b0: Disable. PMU WDT reset output cannot reset SoC system. 1'b1: Enable. PMU WDT reset output will reset SoC system.
11	RW	0x0	pmuwdt_pause_en PMU WDT pause enable. 1'b0: Disable 1'b1: Enable
10	RO	0x0	reserved
9	RW	0x0	lpmcu_soft_irq LPMCU software interrupt. 1'b0: No interrupt 1'b0: Valid interrupt
8	RW	0x0	pmusram_clkgat_disable Disable auto clock gating for PMU SRAM. 1'b0: Enable auto clock gating. PMU SRAM clock is auto gated if there is no access. 1'b1: Disable auto clock gating. PMU SRAM clock is not auto gated if there is no access.
7	RW	0x0	pwm6_switch_ena PWM6 output source swith enable. 1'b0: Disable, PWM6 output source is from PWM module. 1'b1: Enable, PWM6 output source is from PMU module.
6	RW	0x0	pwm5_switch_ena PWM5 output source swith enable. 1'b0: Disable, PWM5 output source is from PWM module. 1'b1: Enable, PWM5 output source is from PMU module.
5	RW	0x0	pwm3_switch_ena PWM3 output source swith enable. 1'b0: Disable, PWM3 output source is from PWM module. 1'b1: Enable, PWM3 output source is from PMU module.
4	RW	0x0	pwm2_switch_ena PWM2 output source swith enable. 1'b0: Disable, PWM2 output source is from PWM module. 1'b1: Enable, PWM2 output source is from PMU module.
3	RW	0x0	pwm1_switch_ena PWM1 output source swith enable. 1'b0: Disable, PWM1 output source is from PWM module. 1'b1: Enable, PWM1 output source is from PMU module.

Bit	Attr	Reset Value	Description
2	RW	0x0	pmic_sleep_m1_pol pmic_sleep_m1 polarity. 1'b0: pmic_sleep_m1 is equal to PMIC_SLEEP. 1'b1: pmic_sleep_m1 is equal to the inverse value of PMIC_SLEEP.
1	RW	0x0	pmic_sleep_m0_pol pmic_sleep_m0 polarity. 1'b0: pmic_sleep_m0 is equal to PMIC_SLEEP. 1'b1: pmic_sleep_m0 is equal to the inverse value of PMIC_SLEEP.
0	RW	0x0	pmic_sleep_sel PMIC_SLEEP source selection. 1'b0: npor_out2chip_rst, from reset pulse generator. 1'b1: pmu_sleep, from PMU block.

PMU_GRF_SOC_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	out2chip_rst_init out2chip_rst width configuration.

PMU_GRF_SOC_CON3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:0	RW	0x000	pmupvtm_clkout_div Divider for PMU PVTM output clock to generate 32KHz clock. Frequency is divided by (pmupvtm_clkout_div*4+1).

PMU_GRF_SOC_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_ddr_fail_safe_hold_ena Hold resetn_ddr_fail_safe. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
14	RW	0x0	presetn_pmu_pvtm_hold_ena Hold presetn_pmu_pvtm. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
13	RW	0x0	presetn_pmu_wdt_hold_ena Hold presetn_pmu_wdt. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
12	RW	0x0	presetn_pmu_sgrf_hold_ena Hold presetn_pmu_sgrf. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
11	RW	0x0	presetn_pmu_sgrf_remap_hold_ena Hold presetn_pmu_sgrf_remap. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
10	RW	0x0	presetn_pmu_mailbox_hold_ena Hold presetn_pmu_mailbox. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
9	RW	0x0	presetn_pmu_ioc_hold_ena Hold presetn_pmu_ioc. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
8	RW	0x0	presetn_pmu_hptimer_hold_ena Hold presetn_pmu_hptimer. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
7	RW	0x0	presetn_pmu_cru_hold_ena Hold presetn_pmu_cru. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
6:5	RO	0x0	reserved
4	RW	0x0	presetn_pmu_biu_hold_ena Hold presetn_pmu_biu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
3	RW	0x0	presetn_i2c1_hold_ena Hold presetn_i2c1. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
2	RW	0x0	hresetn_pmu_sram_hold_ena Hold phresetn_pmu_sram. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
1	RW	0x0	hresetn_pmu_biu_hold_ena Hold hresetn_pmu_biu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	dbresetn_pmu_gpio0_hold_ena Hold dbresetn_pmu_gpio0. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

PMU_GRF_SOC_CON5

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	tresetn_pmu_wdt_hold_ena Hold tresetn_pmu_wdt. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
7	RW	0x0	tresetn_lpmcu_cpu_hold_ena Hold tresetn_lpmcu_cpu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
6	RW	0x0	resetn_pmupvtm_hold_ena Hold resetn_pmupvtm. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
5	RW	0x0	resetn_lpmcu_hold_ena Hold resetn_lpmcu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
4	RW	0x0	resetn_lpmcu_pwrup_hold_ena Hold resetn_lpmcu_pwrup. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
3	RW	0x0	resetn_lpmcu_cpu_hold_ena Hold resetn_lpmcu_cpu. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
2	RW	0x0	resetn_pmu_hptimer_hold_ena Hold resetn_pmu_hptimer. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable
1	RW	0x0	resetn_pmu_32k_hptimer_hold_ena Hold resetn_pmu_32k_hptimer. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	resetn_i2c1_hold_ena Hold resetn_i2c1. When enabled, the relative logic cannot be reset. 1'b0: Disable 1'b1: Enable

PMU_GRF_SOC_STATUS

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x1	npor_powergood NPOR power state. 1'b0: Power badl state 1'b1: Power good state
3	RO	0x0	core_standbywfil2 CORE L2 standbywfi state. 1'b0: CORE L2 in normal state 1'b1: CORE L2 in standbywfi state
2	RO	0x0	core_standbywfi CORE standbywfi state. 1'b0: CORE in normal state 1'b1: CORE in standbywfi state
1	RO	0x1	lpmcu_rst_out LPMCU Non-DN reset from debug module. 1'b0: Reset state 1'b1: Normal state
0	RO	0x0	lpmcu_wfi_halted LPMCU wfi halted state. 1'b0: LPMCU in normal state 1'b1: LPMCU in wfi state

PMU_GRF_PMUIO_CON

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	poc_pmuio_iddq For IDDQ testing
2	RW	0x1	poc_pmuio_cle PMUIO clamping receiver. Set to 1 for 3.3v operation.
1	RW	0x0	poc_pmuio_sel25 Set to 1 for PMUIO VDDO=2.5v operation.
0	RW	0x0	poc_pmuio_sel18 Set to 1 for PMUIO VDDO=1.8v operation.

PMU_GRF_MEM_CON

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:4	RO	0x000	reserved
3:0	RW	0x1	mem_cfg_uhdspra [1:0]: RTSEL [3:2]: WTSEL

PMU_GRF_OS_REG0

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG1

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG2

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG3

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG4

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG5

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG6

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG7

Address: Operational Base + offset (0x021C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG8

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG9

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG10

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_OS_REG11

Address: Operational Base + offset (0x022C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg OS register

PMU_GRF_RSTFUNC_STATUS

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RO	0x0	ddr_fail_safe_src_stat Reset state. When high, ddr_fail_safe is active.
2	RO	0x0	tsadc_shut_reset_src_stat Reset state. When high, reset by TSADC shut trigger.
1	RO	0x0	wdt_reset_src_stat Reset state. When high, reset by WDT trigger.
0	RO	0x0	first_reset_src_stat Reset state. When high, reset by first reset trigger.

PMU_GRF_RSTFUNC_CLR

Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	WO	0x0	ddr_fail_safe_src_clr Clear bit for ddr_fail_safe. 1'b1: Clear enable 1'b0: Clear disable
2	WO	0x0	tsadc_shut_reset_src_clr Clear bit for reset by tsadc shut trigger. 1'b1: Clear enable 1'b0: Clear disable
1	WO	0x0	wdt_reset_src_clr Clear bit for reset by wdt trigger. 1'b1: Clear enable 1'b0: Clear disable

Bit	Attr	Reset Value	Description
0	WO	0x0	first_reset_src_clr Clear bit for reset by first reset trigger. 1'b1: Clear enable 1'b0: Clear disable

4.7 DDR_GRF Register Description

4.7.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDRGRF_CON0	0x0000	W	0x00000080	DDR GRF Config Register 0
DDRGRF_CON1	0x0004	W	0x00002400	DDR GRF Config Register 1
DDRGRF_CON2	0x0008	W	0x000003FF	DDR GRF Config Register 2
DDRGRF_CON3	0x000C	W	0x00000001	DDR GRF Config Register 3
DDRGRF_STATUS0	0x0100	W	0x00000000	DDR GRF Status Register 0
DDRGRF_STATUS1	0x0104	W	0x00000000	DDR GRF Status Register 1
DDRGRF_STATUS2	0x0108	W	0x00000000	DDR GRF Status Register 2
DDRGRF_STATUS3	0x010C	W	0x00000000	DDR GRF Status Register 3
DDRGRF_STATUS4	0x0110	W	0x00000000	DDR GRF Status Register 4
DDRGRF_STATUS5	0x0114	W	0x00000000	DDR GRF Status Register 5
DDRGRF_STATUS6	0x0118	W	0x00000000	DDR GRF Status Register 6
DDRGRF_STATUS7	0x011C	W	0x00000000	DDR GRF Status Register 7
DDRGRF_STATUS8	0x0120	W	0x0000001F	DDR GRF Status Register 8
DDRGRF_STATUS9	0x0124	W	0x00000000	DDR GRF Status Register 9
DDRGRF_PHY_STATUS	0x0160	W	0x00000000	DDR PHY Status Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.7.2 Detail Registers Description

DDRGRF_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:17	RW	0x0000	write_enable Write enable for lower 15bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
16:15	RO	0x0	reserved
14	RW	0x0	grf_con_awpoison_0 Connected to the UMCTL2's input signal, awpoison_0. The awpoison_0 is a off-band signal to indicate an invalid write transaction.
13	RW	0x0	grf_con_awurgent_0 Connected to the UMCTL2's input AXI write urgent signal, awurgent_0.
12	RW	0x0	grf_con_arpoison_0 Connected to the UMCTL2's input signal, arpoison_0. The arpoison_0 is a sideband signal to indicate an invalid read transaction.
11	RW	0x0	grf_con_arurgent_0 Connected to the UMCTL2's input AXI read urgent signal, arurgent_0.
10	RW	0x0	grf_con_pa_wmask Connected to the UMCTL2's input write port mask signal, pa_wmask

Bit	Attr	Reset Value	Description
9:8	RW	0x0	grf_con_pa_rmask Connected to UMCTL2's input read port mask signal, pa_rmask
7	RW	0x1	grf_con_dfi_init_complete When grf_con_dfi_init_complete_sel==1'b1, this signal is selected as UMCTL2's input signal, dfi_init_complete.
6	RW	0x0	grf_con_dfi_init_complete_sel 1'b0: UMCTL2's input signal, dfi_init_complete, is generated by DDR PHY. 1'b1: UMCTL2's input signal, dfi_init_complete, is connected from grf_con_dfi_init_complete.
5	RW	0x0	grf_con_dfi_init_start When grf_con_dfi_init_start_sel==1'b1, this signal is selected as DDR PHY's input signal, dfi_init_start.
4	RW	0x0	grf_con_dfi_init_start_sel 1'b0: DDR PHY's input signal, dfi_init_start, is generated by HW lowpower module, dfi_ctl module or UMCTL2. 1'b1: DDR PHY's input signal, dfi_init_start, is connected from grf_con_dfi_init_start.
3:1	RO	0x0	reserved
0	RW	0x0	grf_con_upctl_slverr_enable 1'b0: Disable DDRC apb pslverr 1'b1: Enable DDRC apb pslverr

DDRGRF_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:18	RW	0x0000	write_enable Write enable for lower 14bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
17:14	RO	0x0	reserved
13	RW	0x1	grf_con_hwlp_ddrphy_io_ret_en Reserved
12	RO	0x0	reserved
11:8	RW	0x4	grf_con_ddrc_auto_sr_dly When SDRAM is in self-refresh state whose type is indicated by grf_con_selfref_type2_en, the core clk of UMCTL2 will be gated after grf_con_ddrc_auto_sr_dly clocks.
7:6	RO	0x0	reserved
5	RW	0x0	grf_con_upctl2_pdsrlp_cg_en Reserved
4	RW	0x0	grf_con_upctl2_syscreq_cg_en 1'b0: Core clk of UMCTL2 is allowed to be gated when csysreq_ddrc^csysack_ddrc==1. 1'b1: Core clk of UMCTL2 would not be gated when csysreq_ddrc^csysack_ddrc==1.
3	RW	0x0	grf_con_selfref_type2_en 1'b0: Only allowed core clk of UMCTL2 to be gated when SDRAM's self-refresh is caused by automatic self-refresh. 1'b1: Allowed core clk of UMCTL2 to be gated when SDRAM's self-refresh is not caused by automatic self-refresh.

Bit	Attr	Reset Value	Description
2	RW	0x0	grf_con_upctl2_core_cg_en 1'b0: Do not gate core clk of UMCTL2 1'b1: Gate core clk of UMCTL2 unless following two case: Case1: grf_con_upctl2_apb_cg_en==1'b1 and the psel of UMCTL2 is asserted Case2: grf_con_upctl2_syscreq_cg_en==1'b1 and csysreq_ddrc^csysack_ddrc==1
1	RW	0x0	grf_con_upctl2_apb_cg_en 1'b0: Axi clk and core clk of UMCTL2 is allowed to be gated when psel of UMCTL2 is asserted. 1'b1: Axi clk and core clk of UMCTL2 would not be gated when psel of UMCTL2 is asserted.
0	RW	0x0	grf_con_upctl2_axi_cg_en 1'b0: Do not gate axi clk of UMCTL2 1'b1: Gate axi clk of UMCTL2 unless grf_con_upctl2_apb_cg_en==1'b1 and the psel of UMCTL2 is asserted.

DDRGRF_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RW	0x0000	write_enable Write enable for lower 15bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
16:15	RO	0x0	reserved
14	RW	0x0	interconnect_mem_ckg_disable Indicates whether to enable the clock gating function of the msch memory. 1'b0: Enable clock gating function 1'b1: Disable clock gating function
13	RW	0x0	grf_con_lpddr45_en Should set to 0 as lpddr4/lpddr5 is not supported.
12	RW	0x0	grf_con_awautopre_0 Connected to the UMCTL2's input signal, awautopre_0. The awautopre_0 is AXI auto-precharge signal for write command.
11	RW	0x0	grf_con_arautopre_0 Connected to the UMCTL2's input signal, arautopre_0. The arautopre_0 is AXI auto-precharge signal for read command.
10	RO	0x0	reserved
9:0	RW	0x3ff	grf_con_ddr_clk_gate The clock gate signals for clocks inside DDRC and Arbiter block. Each bit corresponds to a clock.

DDRGRF_CON3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:20	RW	0x0000	write_enable Write enable for lower 12bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
19:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	grf_con_csysreq_axi_hwlp 1'b0: The hardware low power request signal of AXI block, cysreq_0, is not related to HW low power module. 1'b1: The hardware low power request signal of AXI block, csysreq_0, can be generated by HW low power module.
10	RW	0x0	grf_con_csysreq_ddrc_hwlp 1'b0: The hardware low power request signal of DDRC, cysreq_ddrc, is not related to HW low power module. 1'b1: The hardware low power request signal of DDRC, cysreq_ddrc, can be generated by HW low power module.
9	RW	0x0	grf_con_csysreq_axi_pmu 1'b0: The hardware low power request signal of AXI block, cysreq_0, is not related to PMU. 1'b1: The hardware low power request signal of AXI block, csysreq_0, can be generated by PMU.
8	RW	0x0	grf_con_csysreq_ddrc_pmu 1'b0: The hardware low power request signal of DDRC, cysreq_ddrc, is not related to PMU. 1'b1: The hardware low power request signal of DDRC, cysreq_ddrc, can be generated by PMU.
7:4	RO	0x0	reserved
3:0	RW	0x1	grf_shrm_mem_cfg_uhdspra [1:0]: Connected to system sram's input signal, RTSEL. [3:2]: Connected to system sram's input signal, WTSEL.

DDRGRF_STATUS0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	grf_st_mrr_data0_31_0 The low 32 bit of multi-purpose register 0 (MPR0) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data0_63_32 The high 32 bit of multi-purpose register 0 (MPR0) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data1_31_0 The low 32 bit of multi-purpose register 1 (MPR1) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS3

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data1_63_32 The high 32 bit of multi-purpose register 1 (MPR1) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data2_31_0 The low 32 bit of multi-purpose register 2 (MPR2) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS5

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data2_63_32 The high 32 bit of multi-purpose register 2 (MPR2) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS6

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data3_31_0 The low 32 bit of multi-purpose register 3 (MPR3) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS7

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	grf_st_mrr_data3_63_32 The high 32 bit of multi-purpose register 3 (MPR3) read data. The width of MPR read data is equal to 64 bit.

DDRGRF_STATUS8

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RO	0x0	vi_port_probe_mainstatalarm Reserved
14	RO	0x0	vi_port_probe_maintracealarm Reserved
13	RO	0x0	vepu_port_probe_mainstatalarm Reserved
12	RO	0x0	vepu_port_probe_maintracealarm Reserved
11	RO	0x0	cpu_port_probe_mainstatalarm Reserved
10	RO	0x0	cpu_port_probe_maintracealarm Reserved
9	RO	0x0	npu_port_probe_mainstatalarm Reserved
8	RO	0x0	npu_port_probe_maintracealarm Reserved
7:6	RO	0x0	grf_st_stat_ddrc_reg_selfref_type Current self-refresh status and type. Equivalent to STAT.selfref_type register.

Bit	Attr	Reset Value	Description
5	RO	0x0	cactive_axi State of AXI low-power clock active, cactive_axi
4	RO	0x1	csysack_axi State of AXI low-power request acknowledge, csysack_axi
3	RO	0x1	csysreq_axi State of AXI low-power request, csysreq_axi
2	RO	0x1	cactive_ddrc State of DDRC hardware low-power clock active, cactive_ddrc
1	RO	0x1	csysack_ddrc State of DDRC hardware low-power request acknowledge, cactive_ddrc
0	RO	0x1	csysreq_ddrc State of DDRC hardware low-power request, csysreq_ddrc

DDRGRF_STATUS9

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:16	RO	0x00	grf_st_wr_credit_cnt Value of wr_credit_cnt. It indicates the number of available write CAM slots.
15	RO	0x0	reserved
14:8	RO	0x00	grf_st_hpr_credit_cnt Value of hpr_credit_cnt. It Indicates the number of available High priority read CAM slots.
7	RO	0x0	reserved
6:0	RO	0x00	grf_st_lpr_credit_cnt Value of lpr_credit_cnt. It indicates the number of available Low priority read CAM slots.

DDRGRF_PHY_STATUS

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	ddr_cmd_plllock PLL lock state of DDR PHY. 1'b0: Lock 1'b1: Not lock

4.8 CORE_GRF Register Description

4.8.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CORE_GRF_PVTPLL_CON0_L	0x0000	W	0x00000000	pvtpll control register0 low part
CORE_GRF_PVTPLL_CON0_H	0x0004	W	0x00000000	pvtpll control register0 high part
CORE_GRF_PVTPLL_CON1	0x0008	W	0x00000018	
CORE_GRF_PVTPLL_CON2	0x000C	W	0x00000004	
CORE_GRF_PVTPLL_CON3	0x0010	W	0x00000000	
CORE_GRF_PVTPLL_OSC_CNT	0x0018	W	0x00000000	

Name	Offset	Size	Reset Value	Description
CORE_GRF_PVTPLL_OSC_CNT_AVG	0x001C	W	0x00000000	
CORE_GRF_CACHE_PERI_ADDR_START	0x0024	W	0x00000000	
CORE_GRF_CACHE_PERI_ADDR_END	0x0028	W	0x00000000	
CORE_GRF_MCU_CACHE_MISC	0x002C	W	0x00000000	
CORE_GRF_MCU_CACHE_STATUS	0x0030	W	0x00000000	
CORE_GRF_CPU_STATUS	0x0034	W	0x00000000	
CORE_GRF_CPU_CON0	0x0038	W	0x00000C01	
CORE_GRF_CORE_MEMCFG_UHDSPRA	0x003C	W	0x00000001	

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.8.2 Detail Registers Description

CORE_GRF_PVTPLL_CON0_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	bypass 1'b1: Not support glitch-free frequency switch 1'b0: Support glitch-free frequency switch
14:13	RW	0x0	clk_div_osc Frequency divsion factor for osc_clk
12:11	RW	0x0	clk_div_ref Frequency divsion factor for ref_clk
10:8	RW	0x0	osc_ring_sel Osc ring channel select
7:3	RO	0x00	reserved
2	RW	0x0	out_polar 1'b1: Need to increase when out = 1 1'b0: Need to increase when out = 0
1	RW	0x0	osc_en 1'b1: Osc_ring enable 1'b0: Osc_ring disable
0	RW	0x0	start 1'b1: Pvtpll monitor start 1'b0: Pvtpll monitor doesn't start

CORE_GRF_PVTPLL_CON0_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:0	RW	0x00	ring_length_sel Osc ring inverter length select

CORE_GRF_PVTPLL_CON1

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000018	cal_cnt Frequency measurement period

CORE_GRF_PVTPLL_CON2

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	ckg_val Clk gating interval control count value
15:0	RW	0x0004	threshold Count differency threshold value

CORE_GRF_PVTPLL_CON3

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ref_cnt Target reference frequency value

CORE_GRF_PVTPLL_OSC_CNT

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	osc_clk_cnt_val Osc_clk counter value

CORE_GRF_PVTPLL_OSC_CNT_AVG

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	osc_clk_cnt_ave_value Osc_clk counter average value

CORE_GRF_CACHE_PERI_ADDR_START

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	peri_addr_start Cache periphral high 20-bit base address,the address should be put in low 20-bit when configure the reg

CORE_GRF_CACHE_PERI_ADDR_END

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	periphral_addr_end Cache periphral high 20-bit end address,the address should be put in low 20-bit when configure the reg

CORE_GRF_MCU_CACHE_MISC

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	scr1_soft_irq mcu software irq
2:1	RW	0x0	cache_slv_memattr cache slave memory attribute
0	RW	0x0	cache_flush_req cache flush request

CORE_GRF_MCU_CACHE_STATUS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	ndm_rst_n_out Non-debug module reset output
2	RW	0x0	wfi_halted mcu is in wfi state.
1	RW	0x0	cache_flush_ack cache flush ack
0	RW	0x0	cache_idle cache is in idle state, high active

CORE_GRF_CPU_STATUS

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	jtagnsw High if JTAG selected, low if SWD selected SWJ-DP
5	RW	0x0	jtagtop Indicates TAP in TLR,RTI,Sel-DR or Sel-IR states for SWJ-DP
4	RW	0x0	evento_rising_edge Indicates an event output
3	RW	0x0	standbywfil2 L2 memory system is in wfi state.
2	RW	0x0	smpnamp It signals SMP mode or AMP mode for core.
1	RW	0x0	standbywfi Core is in wfi state.
0	RW	0x0	standbywfe Core is in wfe state.

CORE_GRF_CPU_CON0

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RW	0x0	wakeup_int_pmu_sel It is used to select the interrupt source 1'b1: select interrupt from IPs directly 1'b0: select interrupt from GIC output
14	RW	0x0	po_srst_wfien wfi enable for core power-on softrst
13	RW	0x0	srst_wfien wfi enable for core softrst
12	RW	0x0	evento_clr Clear the event output status to 1'b0
11	RW	0x1	dbgselfaddrv Debug self-address offset valid
10	RW	0x1	dbgromaddrv Debug ROM physical address valid
9	RW	0x0	eventi Event input for processor wake-up from WFE state.
8	RW	0x0	l2rstdisable Disable automatic L2 cache invalidate at reset
7	RW	0x0	l1rstdisable Disable automatic data cache,instruction cache and TLB invalidate at reset
6:3	RO	0x0	reserved
2	RW	0x0	cfgte Controls process state for exception handling(TE bit) at reset 1'b1: Thumb instruction set for exception handling 1'b0: ARM instruction set for exception handling
1	RW	0x0	cfgend Controls endianness during dta exception handling(EE-bit) at reset 1'b1: Big-endian data during exception handling 1'b0: Little-endian data during exception handling
0	RW	0x1	vinithi Sets the base address of the vector table 1'b1: A vector table base address of 0xFFFF0000 1'b0: A vector table base address of 0x00000000

CORE_GRF_CORE_MEMCFG_UHDSPRA

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x0	wtsel Timing adjustment setting for debug purpose
1:0	RW	0x1	rtsel Timing adjustment setting for debug purpose

4.9 VI_GRF Register Description

4.9.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VI_GRF_VI_MISC_CON0	0x0000	W	0x00000080	
VI_GRF_VI_SDMMC_CON0	0x0004	W	0x00000004	
VI_GRF_VI_SDMMC_CON1	0x0008	W	0x00000000	
VI_GRF_VI_STATUS_CON4	0x0010	W	0x00000000	
VI_GRF_VI_CSIPHY_CON5	0x0014	W	0x00000000	
VI_GRF_VI_CSIPHY_STATUS	0x0018	W	0x00000000	
VI_GRF_VI_MEMCFG_UHDSPRA	0x001C	W	0x00000001	
VI_GRF_SDMMC_DET_CNT	0x0024	W	0x00000000	
VI_GRF_SDMMC_SIG_DETECT_CON	0x0028	W	0x00000000	
VI_GRF_SDMMC_SIG_DETECT_STATUS	0x002C	W	0x00000000	
VI_GRF_SDMMC_STATUS_CLR	0x0030	W	0x00000000	
VI_GRF_SDMMC_DET_FLT_CON	0x0034	W	0x00000000	
VI_GRF_CIF_CON	0x0038	W	0x00000000	

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.9.2 Detail Registers Description

VI_GRF_VI_MISC_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	sdmmc_detn_io_pol Indicates SDMMC detectn io pin polar 1'b0: Low active 1'b1: High active
7	RW	0x1	sdmmc_ram_clkgat_disable SDMMC ram hardware clk gating disable 1'b0: Gate SDMMC ram clk when idle 1'b1: SDMMC ram clk is always on
6	RW	0x0	rtc_clamp_en Rtc clamp enable
5	RW	0x0	isp_shutter_trig Mechanical shutter external trigger pulse
4	RO	0x0	reserved
3	RW	0x0	isp_fl_trig Flash light external trigger pulse
2	RW	0x0	vicap_lvds_sel_2lane Vicap lvds select the lane0 and lane1.
1	RO	0x0	reserved
0	RW	0x0	csi2host0_ppi_sel_2lane Csi2host0 ppi select the lane0 and lane1.

VI_GRF_VI_SDMMC_CON0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	test_clkout_0_sel clk output to io for debug 1'b1: cclk_in 1'b0: cclk_drv
11	RW	0x0	drv_sel cclk_in_drv source select: 1'b0: just use clk after phase_shift 1'b1: use clk after phase_shift and delayline
10:3	RW	0x00	drv_delaynum The element number in delayline for cclk_in_drv
2:1	RW	0x2	drv_degree SDMMC driver clock phase for phase shift: 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RW	0x0	init_state Soft initial state for phase shift

VI_GRF_VI_SDMMC_CON1

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	test_clkout_1_sel clk output to io for debug 1'b1: cclk_in 1'b0: cclk_sample
11	RW	0x0	sample_sel cclk_in_sample source select: 1'b0: just use clk after phase_shift 1'b1: use clk after phase_shift and delayline
10:3	RW	0x00	sample_delaynum The element number used in delayline for cclk_in_sample
2:1	RW	0x0	sample_degree SDMMC sample clk phase for phase shift 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RO	0x0	reserved

VI_GRF_VI_STATUS_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	vccio4_vd Indicates vccio6 voltage status, vccio4_vd = 1 when VCCIO > 2.19v. vccio4_vd = 0 when VCCIO < 2.02v.
4	RW	0x0	rtc_ext_off_o Indicates the chip power off,high active
3	RW	0x0	rtc_dig_ldo_rdy Power on reset when low, reset deassert when high.
2	RW	0x0	isp_flash_trig Hold signal for flash light
1	RW	0x0	isp_prelight_trig Hold signal for prelight
0	RW	0x0	isp_shutter_open Hold signal for shutter open

VI_GRF_VI_CSIPHY_CON5

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	rxbyteclkhs_1_inv_sel It is used to select invert rxbyteclkhs_1.
10	RW	0x0	enable_ck_1 D-PHY clk lane1 enable signal
9	RW	0x0	rxbyteclkhs_0_inv_sel It is used to select invert rxbyteclkhs.
8	RW	0x0	enable_ck D-PHY clk lane enable signal, high active
7:4	RW	0x0	lane_enable Enable lane module, This active high signal forces the lane module out of shutdown.
3:0	RW	0x0	force_rx_mode Force lane module into receive mode/wait for stop state.

VI_GRF_VI_CSIPHY_STATUS

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:19	RW	0x0	csiphy_err_control Contorl error This active high signal is asserted when an incorrect line state sequence is detected. For example, if a turnaround request or escape mode request is immediately followed by a Stop state instead of the required Bridge state, this signal is asserted and remains asserted until the next change in line state
18	RW	0x0	csiphy_direction Transmit/Receive direction This signal is used to indicate the current direction of the Lane interconnects. When pin_direction =0, the Lane is in transmit mode. When pin_direction =1, the Lane is in receive mode

Bit	Attr	Reset Value	Description
17	RW	0x0	csiphy_ulpsactivenotclk_1 ULP state(not) active This active low signal is asserted to indicate that the Lane is in ULP state
16	RW	0x0	csiphy_ulpsactivenotclk ULP state(not) active This active low signal is asserted to indicate that the Lane is in ULP state
15	RW	0x0	csiphy_errcontentionlp1_3 LP1 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
14	RW	0x0	csiphy_errcontentionlp0_3 LP0 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
13	RW	0x0	csiphy_errcontentionlp1_2 LP1 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
12	RW	0x0	csiphy_errcontentionlp0_2 LP0 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
11	RW	0x0	csiphy_errcontentionlp1_1 LP1 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
10	RW	0x0	csiphy_errcontentionlp0_1 LP0 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
9	RW	0x0	csiphy_errcontentionlp1_0 LP1 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
8	RW	0x0	csiphy_errcontentionlp0_0 LP0 contention error This active high signal is asserted when the Lane Module detects a contention situation on a line while trying to drive the line low
7:4	RW	0x0	csiphy_rxskewcalhs High speed receive skew calibration This optional active high signal indicates that the high speed deskew burst is being received
3:0	RW	0x0	csiphy_ulpsactivenot ULP state(not) active This active low signal is asserted to indicate that the lane is in ULP state

VI_GRF_VI_MEMCFG_UHDSPRA

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x0	wtsel Timing adjustment setting for debug purpose
1:0	RW	0x1	rtsel Timing adjustment setting for debug purpose

VI_GRF_SDMMC_DET_CNT

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21:0	RW	0x000000	detect_cnt SDMMC detect pin filter counter, in the unit of pclk

VI_GRF_SDMMC_SIG_DETECT_CON

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	sig_detect_fall_en Enable SDMMC detect pin falling edge irq 1'b0: Disable 1'b1: Enable
0	RW	0x0	sig_detect_rise_en Enable SDMMC detect pin rising edge irq 1'b0: Disable 1'b1: Enable

VI_GRF_SDMMC_SIG_DETECT_STATUS

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	sig_detect_fall_irq SDMMC detect pin falling edge irq status 1'b0: Not active 1'b1: Active
0	RW	0x0	sig_detect_rise_irq SDMMC detect pin rising edge irq status 1'b0: Not active 1'b1: Active

VI_GRF_SDMMC_STATUS_CLR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
1	RW	0x0	sig_detect_fall_clr SDMMC detect pin falling edge irq clear 1'b0: Disable 1'b1: Enable
0	RW	0x0	sig_detect_rise_clr SDMMC detect pin rising edge irq clear 1'b0: Disable 1'b1: Enable

VI_GRF_SDMMC_DET_FLT_CON

Address: Operational Base + offset (0x0034)

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

VI_GRF_CIF_CON

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:3	RW	0x00	clk_delaynum_m0 Indicates the delayline number of m0.
2	RW	0x0	datapath It is used to select data from delayline or pipeline.
1	RW	0x0	clk_inv_sel_m0 It is used to select clkin or invert clkin of m0.
0	RW	0x0	m0m1_sel It is used to select io group m0 or io group m1.

4.10 VO_GRF Register Description

4.10.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VOGRF_GMAC_CON0	0x0000	W	0x00000000	GMAC controller configuration
VOGRF_GMAC_CLK_CON	0x0004	W	0x00000000	GMAC clock configuration
VOGRF_GMAC_ST	0x0008	W	0x00000000	GMAC status
VOGRF_TSADC_CON	0x000C	W	0x00000000	TSADC PHY configuration
VOGRF_OTP_CON	0x0010	W	0x00000000	OTP controller lock configuration
VOGRF_RGA_NOC	0x0014	W	0x00000000	RGA Qos configuration
VOGRF_VO_MEM	0x0018	W	0x00000001	VO memory configuration
VOGRF_SDIO_CON0	0x001C	W	0x00000004	SDIO configuration
VOGRF_SDIO_CON1	0x0020	W	0x00000000	SDIO configuration
VOGRF_SDIO_RAM_CLHGAT_DIS	0x0024	W	0x00000000	SDIO configuration
VOGRF_MACPHY_CON0	0x0028	W	0x00000000	MAC PHY configuration
VOGRF_MACPHY_CON1	0x002C	W	0x00000000	MAC PHY configuration
VOGRF_VOP_QOS	0x0030	W	0x00000000	VOP Qos configuration

Name	Offset	Size	Reset Value	Description
VOGRF_VOP_PIPE_BYPASS	0x0034	W	0x00000001	VOP IO PIPE configuration

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.10.2 Detail Registers Description

VOGRF_GMAC_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	sbd_flowctrl Sideband signal to transmit control packet or active backpressure.

VOGRF_GMAC_CLK_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	clk_gmac_50_o_sel 1'b1: Select tx clock from CRU 1'b0: Select tx clock from PHY
3:2	RW	0x0	rmii_txclk_sel Mac clock divide sel. 2'b00: 2.5Mhz 2'b01: 25Mhz Others: Reserved
1	RW	0x0	rmii_gate_en If this bit set to 1'b1, this bit will gate RMII clk.
0	RW	0x0	rmii_mode If using RMII mode, set this bit to 1'b1, otherwise set this bit to 1'b0.

VOGRF_GMAC_ST

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x0	macphy_dpx_led Duplex or Collision LED This signal represents Duples LED in full Duplex mode and Collision LED in Half Duplex mode, 1'b1: Full duplex 1'b0: Half duplex Flash: Collision

Bit	Attr	Reset Value	Description
3	RO	0x0	macphy_spd_led Speed LED 1'b1: 100Mbps 1'b0: 10Mbps
2	RO	0x0	macphy_lnk_led Link or Activity LED 1'b1: Link up 1'b0: Link down Flash: Activity
1:0	RO	0x0	mac_speed_o Mac Speed indication. 2'b00: 1000Mbps 2'b01: 2500Mbps 2'b10: 10Mbps 2'b11: 100Mbps

VOGRF_TSADC_CON

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	tsadc_tsen_en TSADC enable register. 1'b1: Enable 1'b0: Disable
7:0	RW	0x00	tsadc_ana_reg TSADC configure registers.

VOGRF_OTP_CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	otpc_s_lock GRF control OTP secure controller lock. This is controlled by VOSGRF_OTPC_CON[0]. 1'b1: Lock 1'b0: Not lock
0	RW	0x0	otpc_ns_lock GRF control OTP none secure controller lock. This is controlled by VOSGRF_OTPC_CON[0]. 1'b1: Lock 1'b0: Not lock

VOGRF_RGA_NOC

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	rga_aw_qos This is used to set RGA Aw_Qos for NOC.

VOGRF_VO_MEM

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2	RW	0x0	wtesl Configuration for wtsel for memory for RGA/VOP/GMAC.
1:0	RW	0x1	rtsel Configuration for rtsel for memory for RGA/VOP/GMAC.

VOGRF_SDIO_CON0

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	test_clkout_0_sel 1'b0: Cclk_drv 1'b1: Cclk_in
11	RW	0x0	drvsel cclk_in_drv source select: 1'b0: Just use clock after phase_shift 1'b1: Use clock after phase_shift and delay line
10:3	RW	0x00	drv_delaynum The element number in delay line for cclk_in_drv.
2:1	RW	0x2	drv_degree SDMMC driver clock phase for phase shift: 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RW	0x0	init_state Soft initial state for phase shift.

VOGRF_SDIO_CON1

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	test_clkout_1_sel 1'b0: Cclk_drv 1'b1: Cclk_in
11	RW	0x0	sample_sel cclk_in_sample source select: 1'b0: Just use clock after phase_shift 1'b1: Use clock after phase_shift and delay line
10:3	RW	0x00	sample_delaynum The element number in delay line for cclk_in_drv.
2:1	RW	0x0	sample_degree SDMMC sample clock phase for phase shift: 2'b00: 0 degree 2'b01: 90 degree 2'b10: 180 degree 2'b11: 270 degree
0	RO	0x0	reserved

VOGRF_SDIO_RAM_CLHGAT_DIS

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	ram_clkgate_disable Disable SDIO memory clock gate. 1'b1: Disable 1'b0: Enable

VOGRF_MACPHY_CON0

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	led_pol Reserved
14:10	RW	0x00	id PHY Address
9:7	RW	0x0	clksel Reference clock select, only support 3'h6.
6:5	RW	0x0	xmii_sel SMI and MII/RMII interface selection 2'b00: Internal SMI and MII 2'b01: External SMI and MII 2'b10: Internal SMI and RMII 2'b11: External SMI and RMII
4	RW	0x0	fxen Reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	mode Operation mode selection 2'b00: Normal mode 2'b01: Sim mode (not used) 2'b10: AFE test mode 2'b11: Reserved
1	RW	0x0	shutdown Analog block shutdown 1'b1: Shutdown 1'b0: Power up
0	RW	0x0	smi_sel Reserved

VOGRF_MACPHY_CON1

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:0	RW	0x0	bgs Band Gap Selection.

VOGRF_VOP_QOS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	vop_arqos_sel 1'b1: Sel grf_vop_ar_qos 1'b0: Sel VOP controller in bit[2:1] and vop_arqos_bit0 in bit[0].
8	RW	0x0	vop_awqos_sel 1'b1: Sel grf_vop_aw_qos 1'b0: Sel VOP controller in bit[2:1] and vop_awqos_bit0 in bit[0].
7	RW	0x0	vop_arqos_bit0 This is the supplement for VOP Ar_Qos in bit 0, this only used when vop_arqos_sel=1'b0.
6	RW	0x0	vop_awqos_bit0 This is the supplement for VOP Aw_Qos in bit 0, this only used when vop_awqos_sel=1'b0.
5:3	RW	0x0	grf_vop_aw_qos GRF control Aw_Qos for VOP, this only used when vop_awqos_sel=1'b1.
2:0	RW	0x0	grf_vop_ar_qos GRF control Ar_Qos for VOP, this only used when vop_arqos_sel=1'b1.

VOGRF_VOP_PIPE_BYPASS

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bit, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
1:0	RW	0x3	vop_pipe_bypass 2'b00: Use two flip-flop pipe in VOP LCD IO in vccio5. 2'b11: Bypass two flip-flop pipe in VOP LCD IO in vccio5. Other: Reserved

4.11 GPIO0_IOC Register Description

4.11.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO0_IOC_GPIO0A_IOMUX_SEL_L	0x0000	W	0x00000000	GPIO0A iomux configuration register 0
GPIO0_IOC_GPIO0A_IOMUX_SEL_H	0x0004	W	0x00000000	GPIO0A iomux configuration register 1
GPIO0_IOC_GPIO0A_DS0	0x0010	W	0x00000101	GPIO0A drive strength register 0
GPIO0_IOC_GPIO0A_DS1	0x0014	W	0x00000101	GPIO0A drive strength register 1
GPIO0_IOC_GPIO0A_DS2	0x0018	W	0x00000101	GPIO0A drive strength register 2
GPIO0_IOC_GPIO0A_DS3	0x001C	W	0x00000001	GPIO0A drive strength register 3
GPIO0_IOC_GPIO0A_IE	0x0030	W	0x0000007F	GPIO0A input enable register
GPIO0_IOC_GPIO0A_P	0x0038	W	0x00002A68	GPIO0A pull up and pull down configuration register
GPIO0_IOC_GPIO0A_SUS	0x0040	W	0x00000000	GPIO0A keeper configuration register
GPIO0_IOC_GPIO0A_SL	0x0048	W	0x00003FFF	GPIO0A skew rate register
GPIO0_IOC_GPIO0A_IE_SMT	0x0058	W	0x0000007F	GPIO0A Schmitt trigger register
GPIO0_IOC_GPIO0A_OD	0x0068	W	0x00000000	GPIO0A open drain register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.11.2 Detail Registers Description

GPIO0_IOC_GPIO0A_IOMUX_SEL_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0a3_sel 3'h0: GPIO0A3 3'h1: PMIC_SLEEP_M1 Others: Reserved
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio0a2_sel 3'h0: GPIO0A2 3'h1: PWM3_IR_M0 3'h3: TEST_CLK7_OUT Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0a1_sel 3'h0: GPIO0A1 3'h1: UART0_TX_M0 3'h2: PWM2_M0 3'h3: TEST_CLK6_OUT Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0a0_sel 3'h0: GPIO0A0 3'h1: UART0_RX_M0 3'h2: CLK_32K 3'h3: CLK_REF_OUT 3'h4: RTC_CLKO Others: Reserved

GPIO0_IOC_GPIO0A_IOMUX_SEL_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10:8	RW	0x0	gpio0a6_sel 3'h0: GPIO0A6 3'h1: I2C1_SDA_M0 3'h2: UART1_CTSN_M0 3'h3: PWM6_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0a5_sel 3'h0: GPIO0A5 3'h1: I2C1_SCL_M0 3'h2: UART1_RTSN_M0 3'h3: PWM5_M0 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio0a4_sel 3'h0: GPIO0A4 3'h1: PMIC_SLEEP_M0 3'h2: PWM1_M0 Others: Reserved

GPIO0_IOC_GPIO0A_DS0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x01	gpio0a1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio0a0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO0_IOC_GPIO0A_DS1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x01	gpio0a3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio0a2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO0_IOC_GPIO0A_DS2

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x01	gpio0a5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio0a4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO0_IOC_GPIO0A_DS3

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5:0	RW	0x01	gpio0a6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO0_IOC_GPIO0A_IE

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x1	gpio0a6_ie Receiver enable. 1'b0: Disable 1'b1: Enable
5	RW	0x1	gpio0a5_ie Receiver enable. 1'b0: Disable 1'b1: Enable
4	RW	0x1	gpio0a4_ie Receiver enable. 1'b0: Disable 1'b1: Enable
3	RW	0x1	gpio0a3_ie Receiver enable. 1'b0: Disable 1'b1: Enable
2	RW	0x1	gpio0a2_ie Receiver enable. 1'b0: Disable 1'b1: Enable
1	RW	0x1	gpio0a1_ie Receiver enable. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio0a0_ie Receiver enable. 1'b0: Disable 1'b1: Enable

GPIO0_IOC_GPIO0A_P

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x1	gpio0a6_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
12	RW	0x0	gpio0a6_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
11	RW	0x1	gpio0a5_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
10	RW	0x0	gpio0a5_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
9	RW	0x1	gpio0a4_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
8	RW	0x0	gpio0a4_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
7	RW	0x0	gpio0a3_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
6	RW	0x1	gpio0a3_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
5	RW	0x1	gpio0a2_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
4	RW	0x0	gpio0a2_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
3	RW	0x1	gpio0a1_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
2	RW	0x0	gpio0a1_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
1	RW	0x0	gpio0a0_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x0	gpio0a0_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable

GPIO0_IOC_GPIO0A_SUS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	gpio0a6_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
5	RW	0x0	gpio0a5_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
4	RW	0x0	gpio0a4_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
3	RW	0x0	gpio0a3_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
2	RW	0x0	gpio0a2_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
1	RW	0x0	gpio0a1_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x0	gpio0a0_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable

GPIO0_IOC_GPIO0A_SL

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x3	gpio0a6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio0a5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

Bit	Attr	Reset Value	Description
9:8	RW	0x3	gpio0a4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:6	RW	0x3	gpio0a3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio0a2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio0a1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio0a0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO0_IOC_GPIO0A_IE_SMT

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x1	gpio0a6_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
5	RW	0x1	gpio0a5_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
4	RW	0x1	gpio0a4_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
3	RW	0x1	gpio0a3_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
2	RW	0x1	gpio0a2_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
1	RW	0x1	gpio0a1_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x1	gpio0a0_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable

GPIO0_IOC_GPIO0A_OD

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6	RW	0x0	gpio0a6_od 1'b0: Default for regular IO 1'b1: For open drain functionality
5	RW	0x0	gpio0a5_od 1'b0: Default for regular IO 1'b1: For open drain functionality
4	RW	0x0	gpio0a4_od 1'b0: Default for regular IO 1'b1: For open drain functionality
3	RW	0x0	gpio0a3_od 1'b0: Default for regular IO 1'b1: For open drain functionality
2	RW	0x0	gpio0a2_od 1'b0: Default for regular IO 1'b1: For open drain functionality
1	RW	0x0	gpio0a1_od 1'b0: Default for regular IO 1'b1: For open drain functionality
0	RW	0x0	gpio0a0_od 1'b0: Default for regular IO 1'b1: For open drain functionality

4.12 GPIO1_IOC Register Description

Name	Offset	Size	Reset Value	Description
GPIO1_IOC_GPIO1A_IOMUX_SEL_L	0x0000	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1A_IOMUX_SEL_H	0x0004	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1B_IOMUX_SEL_L	0x0008	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1C_IOMUX_SEL_L	0x0010	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1C_IOMUX_SEL_H	0x0014	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1D_IOMUX_SEL_L	0x0018	W	0x00000000	GPIO1 iomux configuration
GPIO1_IOC_GPIO1A_DS0	0x0080	W	0x00000101	GPIO1 drive strength
GPIO1_IOC_GPIO1A_DS1	0x0084	W	0x00000103	GPIO1 drive strength

Name	Offset	Size	Reset Value	Description
GPIO1_IOC_GPIO1A_DS2	0x0088	W	0x00000001	GPIO1 drive strength
GPIO1_IOC_GPIO1B_DS0	0x0090	W	0x00000301	GPIO1 drive strength
GPIO1_IOC_GPIO1B_DS1	0x0094	W	0x00000101	GPIO1 drive strength
GPIO1_IOC_GPIO1C_DS0	0x00A0	W	0x00000707	GPIO1 drive strength
GPIO1_IOC_GPIO1C_DS1	0x00A4	W	0x00000707	GPIO1 drive strength
GPIO1_IOC_GPIO1C_DS2	0x00A8	W	0x00000707	GPIO1 drive strength
GPIO1_IOC_GPIO1C_DS3	0x00AC	W	0x00000707	GPIO1 drive strength
GPIO1_IOC_GPIO1D_DS0	0x00B0	W	0x00000707	GPIO1 drive strength
GPIO1_IOC_GPIO1D_DS1	0x00B4	W	0x00000F07	GPIO1 drive strength
GPIO1_IOC_GPIO1A_IE	0x0180	W	0x0000001F	GPIO1 input enable
GPIO1_IOC_GPIO1B_IE	0x0184	W	0x0000000F	GPIO1 input enable
GPIO1_IOC_GPIO1C_IE	0x0188	W	0x000000FF	GPIO1 input enable
GPIO1_IOC_GPIO1D_IE	0x018C	W	0x0000000F	GPIO1 input enable
GPIO1_IOC_GPIO1A_P	0x01C0	W	0x000002AA	GPIO1 pull up and pull down configuration
GPIO1_IOC_GPIO1B_P	0x01C4	W	0x000000AA	GPIO1 pull up and pull down configuration
GPIO1_IOC_GPIO1C_P	0x01C8	W	0x0000AAAA	GPIO1 pull up and pull down configuration
GPIO1_IOC_GPIO1D_P	0x01CC	W	0x000000AA	GPIO1 pull up and pull down configuration
GPIO1_IOC_GPIO1A_SUS	0x0200	W	0x00000000	GPIO1 keeper configuration
GPIO1_IOC_GPIO1B_SUS	0x0204	W	0x00000000	GPIO1 keeper configuration
GPIO1_IOC_GPIO1C_SUS	0x0208	W	0x00000000	GPIO1 keeper configuration
GPIO1_IOC_GPIO1D_SUS	0x020C	W	0x00000000	GPIO1 keeper configuration
GPIO1_IOC_GPIO1A_SL	0x0240	W	0x000003FF	GPIO1 skew rate
GPIO1_IOC_GPIO1B_SL	0x0244	W	0x000000FF	GPIO1 skew rate
GPIO1_IOC_GPIO1C_SL	0x0248	W	0x0000FFFF	GPIO1 skew rate
GPIO1_IOC_GPIO1D_SL	0x024C	W	0x000000FF	GPIO1 skew rate
GPIO1_IOC_GPIO1A_IE_SMT	0x0280	W	0x0000001F	GPIO1 Schmitt trigger
GPIO1_IOC_GPIO1B_IE_SMT	0x0284	W	0x0000000F	GPIO1 Schmitt trigger
GPIO1_IOC_GPIO1C_IE_SMT	0x0288	W	0x000000FF	GPIO1 Schmitt trigger
GPIO1_IOC_GPIO1D_IE_SMT	0x028C	W	0x0000000F	GPIO1 Schmitt trigger
GPIO1_IOC_GPIO1A_OD	0x02C0	W	0x00000000	GPIO1 open drain
GPIO1_IOC_GPIO1B_OD	0x02C4	W	0x00000000	GPIO1 open drain
GPIO1_IOC_GPIO1C_OD	0x02C8	W	0x00000000	GPIO1 open drain
GPIO1_IOC_GPIO1D_OD	0x02CC	W	0x00000000	GPIO1 open drain
GPIO1_IOC_FORCE_JTAG_UART	0x02F4	W	0x00000001	GPIO1 force jtag

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.12.1 Detail Registers Description

GPIO1_IOC_GPIO1A_IOMUX_SEL_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14:12	RW	0x0	gpio1a3_sel 3'h0: GPIO 3'h1: UART1_TX_M0 3'h2: I2C0_SCL_M0
11	RO	0x0	reserved
10:8	RW	0x0	gpio1a2_sel 3'h0: GPIO 3'h1: PWM0_M0 3'h2: AVS_ARM 3'h3: VICAP_D0_M1
7	RO	0x0	reserved
6:4	RW	0x0	gpio1a1_sel 3'h0: GPIO 3'h1: UART3_RX_M0 3'h2: I2C2_SDA_M0 3'h3: PMU_DEBUG 3'h4: PWM4_M0
3	RO	0x0	reserved
2:0	RW	0x0	gpio1a0_sel 3'h0: GPIO 3'h1: UART3_TX_M0 3'h2: I2C2_SCL_M0 3'h3: PWM7_IR_M0

GPIO1_IOC_GPIO1A_IOMUX_SEL_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0000	reserved
2:0	RW	0x0	gpio1a4_sel 3'h0: GPIO 3'h1: UART1_RX_M0 3'h2: I2C0_SDA_M0

GPIO1_IOC_GPIO1B_IOMUX_SEL_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1b3_sel 3'h0: GPIO 3'h1: A7_JTAG_TMS_M1 3'h2: UART2_RX_M1 3'h3: HPMCU_JTAG_TMS_M0 3'h4: LPMCU_JTAG_TMS_M0
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio1b2_sel 3'h0: GPIO 3'h1: A7_JTAG_TCK_M1 3'h2: UART2_TX_M1 3'h3: HPMCU_JTAG_TCK_M0 3'h4: LPMCU_JTAG_TCK_M0
7	RO	0x0	reserved
6:4	RW	0x0	gpio1b1_sel 3'h0: GPIO 3'h1: UART4_TX_M0 3'h2: PWM7_IR_M1 3'h3: SPI1_CS1N_M0 3'h4: VICAP_D1_M1
3	RO	0x0	reserved
2:0	RW	0x0	gpio1b0_sel 3'h0: GPIO 3'h1: UART4_RX_M0 3'h2: PWM3_IR_M1

GPIO1_IOC_GPIO1C_IOMUX_SEL_L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1c3_sel 3'h0: GPIO 3'h1: LCD_D4 3'h2: VICAP_D5_M1 3'h3: PWM6_M2 3'h4: I2C4_SDA_M1 3'h5: SDMMC1_CMD_M1 3'h6: SPI0_MISO_M0
11	RO	0x0	reserved
10:8	RW	0x0	gpio1c2_sel 3'h0: GPIO 3'h1: LCD_D5 3'h2: VICAP_D4_M1 3'h3: PWM5_M2 3'h4: I2C4_SCL_M1 3'h5: SDMMC1_CLK_M1 3'h6: SPI0_MOSI_M0
7	RO	0x0	reserved
6:4	RW	0x0	gpio1c1_sel 3'h0: GPIO 3'h1: LCD_D6 3'h2: VICAP_D3_M1 3'h3: PWM4_M2 3'h4: SPI0_CLK_M0 3'h5: SDMMC1_D0_M1
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio1c0_sel 3'h0: GPIO 3'h1: LCD_D7 3'h2: VICAP_D2_M1 3'h3: PWM2_M2 3'h4: SPI0_CS0N_M0 3'h5: SDMMC1_D1_M1

GPIO1_IOC_GPIO1C_IOMUX_SEL_H

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio1c7_sel 3'h0: GPIO 3'h1: LCD_D0 3'h2: VICAP_D9_M1 3'h3: PWM11_IR_M1 3'h4: UART4_CTSN_M1
11	RO	0x0	reserved
10:8	RW	0x0	gpio1c6_sel 3'h0: GPIO 3'h1: LCD_D1 3'h2: VICAP_D8_M1 3'h3: PWM10_M1 3'h4: UART4_RTSN_M1
7	RO	0x0	reserved
6:4	RW	0x0	gpio1c5_sel 3'h0: GPIO 3'h1: LCD_D2 3'h2: VICAP_D7_M1 3'h3: PWM9_M1 3'h4: UART4_TX_M1 3'h5: SDMMC1_D2_M1
3	RO	0x0	reserved
2:0	RW	0x0	gpio1c4_sel 3'h0: GPIO 3'h1: LCD_D3 3'h2: VICAP_D6_M1 3'h3: PWM8_M1 3'h4: UART4_RX_M1 3'h5: SDMMC1_D3_M1

GPIO1_IOC_GPIO1D_IOMUX_SEL_L

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio1d3_sel 3'h0: GPIO 3'h1: LCD_CLK 3'h2: VICAP_CLKOUT_M1 3'h3: I2C3_SCL_M1 3'h4: UART5_TX_M1 3'h5: PWM11_IR_M2 3'h6: Reserved 3'h7: DSMAUDIO_N
11	RO	0x0	reserved
10:8	RW	0x0	gpio1d2_sel 3'h0: GPIO 3'h1: LCD_VSYNC 3'h2: VICAP_VSYNC_M1 3'h3: I2C3_SDA_M1 3'h4: UART5_RX_M1 3'h5: SPI0_CS1N_M0 3'h6: PWM0_M1 3'h7: DSMAUDIO_P
7	RO	0x0	reserved
6:4	RW	0x0	gpio1d1_sel 3'h0: GPIO 3'h1: LCD_HSYNC 3'h2: VICAP_HSYNC_M1 3'h3: PWM10_M2 3'h4: UART5_CTSN_M1 3'h5: UART3_RX_M1
3	RO	0x0	reserved
2:0	RW	0x0	gpio1d0_sel 3'h0: GPIO 3'h1: LCD_DEN 3'h2: VICAP_CLKIN_M1 3'h3: PWM3_IR_M2 3'h4: UART5_RTSN_M1 3'h5: UART3_TX_M1

GPIO1_IOC_GPIO1A_DS0

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x01	gpio1a1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio1a0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1A_DS1

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x01	gpio1a3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x03	gpio1a2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1A_DS2

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5:0	RW	0x01	gpio1a4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1B_DS0

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	gpio1b1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio1b0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1B_DS1

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x01	gpio1b3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x01	gpio1b2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1C_DS0

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio1c1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1c0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1C_DS1

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio1c3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1c2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1C_DS2

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x07	gpio1c5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1c4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1C_DS3

Address: Operational Base + offset (0x00AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio1c7_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1c6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1D_DS0

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio1d1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1d0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1D_DS1

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x0f	gpio1d3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio1d2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO1_IOC_GPIO1A_IE

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x1	gpio1a4_ie GPIO1A4 IE control Active High input buffer enable
3	RW	0x1	gpio1a3_ie GPIO1A3 IE control Active High input buffer enable
2	RW	0x1	gpio1a2_ie GPIO1A2 IE control Active High input buffer enable
1	RW	0x1	gpio1a1_ie GPIO1A1 IE control Active High input buffer enable
0	RW	0x1	gpio1a0_ie GPIO1A0 IE control Active High input buffer enable

GPIO1_IOC_GPIO1B_IE

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio1b3_ie GPIO1B3 IE control Active High input buffer enable
2	RW	0x1	gpio1b2_ie GPIO1B2 IE control Active High input buffer enable
1	RW	0x1	gpio1b1_ie GPIO1B1 IE control Active High input buffer enable
0	RW	0x1	gpio1b0_ie GPIO1B0 IE control Active High input buffer enable

GPIO1_IOC_GPIO1C_IE

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1c7_ie GPIO1C7 IE control Active High input buffer enable
6	RW	0x1	gpio1c6_ie GPIO1C6 IE control Active High input buffer enable
5	RW	0x1	gpio1c5_ie GPIO1C5 IE control Active High input buffer enable
4	RW	0x1	gpio1c4_ie GPIO1C4 IE control Active High input buffer enable
3	RW	0x1	gpio1c3_ie GPIO1C3 IE control Active High input buffer enable
2	RW	0x1	gpio1c2_ie GPIO1C2 IE control Active High input buffer enable
1	RW	0x1	gpio1c1_ie GPIO1C1 IE control Active High input buffer enable
0	RW	0x1	gpio1c0_ie GPIO1C0 IE control Active High input buffer enable

GPIO1_IOC_GPIO1D_IE

Address: Operational Base + offset (0x018C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio1d3_ie GPIO1D4 IE control Active High input buffer enable
2	RW	0x1	gpio1d2_ie GPIO1D2 IE control Active High input buffer enable
1	RW	0x1	gpio1d1_ie GPIO1D1 IE control Active High input buffer enable
0	RW	0x1	gpio1d0_ie GPIO1D0 IE control Active High input buffer enable

GPIO1_IOC_GPIO1A_P

Address: Operational Base + offset (0x01C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x1	gpio1a4_pd Active High. Weak pull down for pad.
8	RW	0x0	gpio1a4_pu Active High. Weak pull up for pad.
7	RW	0x1	gpio1a3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio1a3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio1a2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio1a2_pu Active High. Weak pull up for pad.
3	RW	0x1	gpio1a1_pd Active High. Weak pull down for pad.
2	RW	0x0	gpio1a1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio1a0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio1a0_pu Active High. Weak pull up for pad.

GPIO1_IOC_GPIO1B_P

Address: Operational Base + offset (0x01C4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1b3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio1b3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio1b2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio1b2_pu Active High. Weak pull up for pad.
3	RW	0x1	gpio1b1_pd Active High. Weak pull down for pad.
2	RW	0x0	gpio1b1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio1b0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio1b0_pu Active High. Weak pull up for pad.

GPIO1_IOC_GPIO1C_P

Address: Operational Base + offset (0x01C8)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	gpio1c7_pd Active High. Weak pull down for pad.
14	RW	0x0	gpio1c7_pu Active High. Weak pull up for pad.
13	RW	0x1	gpio1c6_pd Active High. Weak pull down for pad.
12	RW	0x0	gpio1c6_pu Active High. Weak pull up for pad.
11	RW	0x1	gpio1c5_pd Active High. Weak pull down for pad.
10	RW	0x0	gpio1c5_pu Active High. Weak pull up for pad.
9	RW	0x1	gpio1c4_pd Active High. Weak pull down for pad.
8	RW	0x0	gpio1c4_pu Active High. Weak pull up for pad.
7	RW	0x1	gpio1c3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio1c3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio1c2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio1c2_pu Active High. Weak pull up for pad.
3	RW	0x1	gpio1c1_pd Active High. Weak pull down for pad.
2	RW	0x0	gpio1c1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio1c0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio1c0_pu Active High. Weak pull up for pad.

GPIO1_IOC_GPIO1D_P

Address: Operational Base + offset (0x01CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1d3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio1d3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio1d2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio1d2_pu Active High. Weak pull up for pad.
3	RW	0x1	gpio1d1_pd Active High. Weak pull down for pad.

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio1d1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio1d0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio1d0_pu Active High. Weak pull up for pad.

GPIO1_IOC_GPIO1A_SUS

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	gpio1a4_sus Active High. Weak pull keeper for pad.
3	RW	0x0	gpio1a3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio1a2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio1a1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio1a0_sus Active High. Weak pull keeper for pad.

GPIO1_IOC_GPIO1B_SUS

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio1b3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio1b2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio1b1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio1b0_sus Active High. Weak pull keeper for pad.

GPIO1_IOC_GPIO1C_SUS

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1c7_sus Active High. Weak pull keeper for pad.
6	RW	0x0	gpio1c6_sus Active High. Weak pull keeper for pad.

Bit	Attr	Reset Value	Description
5	RW	0x0	gpio1c5_sus Active High. Weak pull keeper for pad.
4	RW	0x0	gpio1c4_sus Active High. Weak pull keeper for pad.
3	RW	0x0	gpio1c3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio1c2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio1c1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio1c0_sus Active High. Weak pull keeper for pad.

GPIO1_IOC_GPIO1D_SUS

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio1d3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio1d2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio1d1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio1d0_sus Active High. Weak pull keeper for pad.

GPIO1_IOC_GPIO1A_SL

Address: Operational Base + offset (0x0240)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9:8	RW	0x3	gpio1a4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:6	RW	0x3	gpio1a3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio1a2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio1a1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

Bit	Attr	Reset Value	Description
1:0	RW	0x3	gpio1a0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO1_IOC_GPIO1B_SL

Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:6	RW	0x3	gpio1b3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio1b2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio1b1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio1b0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO1_IOC_GPIO1C_SL

Address: Operational Base + offset (0x0248)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17:16	RW	0x0	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	gpio1c7_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
13:12	RW	0x3	gpio1c6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio1c5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
9:8	RW	0x3	gpio1c4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

Bit	Attr	Reset Value	Description
7:6	RW	0x3	gpio1c3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio1c2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio1c1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio1c0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO1_IOC_GPIO1D_SL

Address: Operational Base + offset (0x024C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:6	RW	0x3	gpio1d3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio1d2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio1d1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio1d0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO1_IOC_GPIO1A_IE_SMT

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x1	gpio1a4_smt Active High. Receiver enable for Schmitt trigger.
3	RW	0x1	gpio1a3_smt Active High. Receiver enable for Schmitt trigger.

Bit	Attr	Reset Value	Description
2	RW	0x1	gpio1a2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio1a1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio1a0_smt Active High. Receiver enable for Schmitt trigger.

GPIO1_IOC_GPIO1B_IE_SMT

Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio1b3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio1b2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio1b1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio1b0_smt Active High. Receiver enable for Schmitt trigger.

GPIO1_IOC_GPIO1C_IE_SMT

Address: Operational Base + offset (0x0288)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio1c7_smt Active High. Receiver enable for Schmitt trigger.
6	RW	0x1	gpio1c6_smt Active High. Receiver enable for Schmitt trigger.
5	RW	0x1	gpio1c5_smt Active High. Receiver enable for Schmitt trigger.
4	RW	0x1	gpio1c4_smt Active High. Receiver enable for Schmitt trigger.
3	RW	0x1	gpio1c3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio1c2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio1c1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio1c0_smt Active High. Receiver enable for Schmitt trigger.

GPIO1_IOC_GPIO1D_IE_SMT

Address: Operational Base + offset (0x028C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio1d3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio1d2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio1d1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio1d0_smt Active High. Receiver enable for Schmitt trigger.

GPIO1_IOC_GPIO1A_OD

Address: Operational Base + offset (0x02C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RW	0x0	gpio1a4_od 1'b1: For open drain functionality 1'b0: Default for regular IO
3	RW	0x0	gpio1a3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio1a2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio1a1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RW	0x0	gpio1a0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO1_IOC_GPIO1B_OD

Address: Operational Base + offset (0x02C4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio1b3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio1b2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio1b1_od 1'b1: For open drain functionality 1'b0: Default for regular IO

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio1b0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO1_IOC_GPIO1C_OD

Address: Operational Base + offset (0x02C8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio1c7_od 1'b1: For open drain functionality 1'b0: Default for regular IO
6	RW	0x0	gpio1c6_od 1'b1: For open drain functionality 1'b0: Default for regular IO
5	RW	0x0	gpio1c5_od 1'b1: For open drain functionality 1'b0: Default for regular IO
4	RW	0x0	gpio1c4_od 1'b1: For open drain functionality 1'b0: Default for regular IO
3	RW	0x0	gpio1c3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio1c2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio1c1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RW	0x0	gpio1c0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO1_IOC_GPIO1D_OD

Address: Operational Base + offset (0x02CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio1d3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio1d2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio1d1_od 1'b1: For open drain functionality 1'b0: Default for regular IO

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio1d0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO1_IOC_FORCE_JTAG_UART

Address: Operational Base + offset (0x02F4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x1	jtag_uart_auto_switch_en This bit is used to enable uart auto-switch to a7_jtag_m1,active high.

4.13 GPIO2_IOC Register Description

4.13.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO2_IOC_GPIO2A_IOMUX_SEL_L	0x0020	W	0x00000000	VCCIO5_GPIO2A iomux configuration
GPIO2_IOC_GPIO2A_IOMUX_SEL_H	0x0024	W	0x00000000	VCCIO5_GPIO2A iomux configuration
GPIO2_IOC_GPIO2B_IOMUX_SEL_L	0x0028	W	0x00000000	VCCIO5_GPIO2B iomux configuration
GPIO2_IOC_GPIO2A_DS0	0x00C0	W	0x00000707	VCCIO5_GPIO2A drive strength
GPIO2_IOC_GPIO2A_DS1	0x00C4	W	0x0000070F	VCCIO5_GPIO2A drive strength
GPIO2_IOC_GPIO2A_DS2	0x00C8	W	0x00000007	VCCIO5_GPIO2A drive strength
GPIO2_IOC_GPIO2A_DS3	0x00CC	W	0x00000700	VCCIO5_GPIO2A drive strength
GPIO2_IOC_GPIO2B_DS0	0x00D0	W	0x00000707	VCCIO5_GPIO2B drive strength
GPIO2_IOC_GPIO2A_IE	0x0190	W	0x000000FF	VCCIO5_GPIO2A input enable
GPIO2_IOC_GPIO2B_IE	0x0194	W	0x00000003	VCCIO5_GPIO2B input enable
GPIO2_IOC_GPIO2A_P	0x01D0	W	0x0000AAAA	VCCIO5_GPIO2A pull up and pull down configuration
GPIO2_IOC_GPIO2B_P	0x01D4	W	0x0000000A	VCCIO5_GPIO2B pull up and pull down configuration
GPIO2_IOC_GPIO2A_SUS	0x0210	W	0x00000000	VCCIO5_GPIO2A keeper configuration
GPIO2_IOC_GPIO2B_SUS	0x0214	W	0x00000000	VCCIO5_GPIO2B keeper configuration
GPIO2_IOC_GPIO2A_SL	0x0250	W	0x0000FFFF	VCCIO5_GPIO2A skew rate
GPIO2_IOC_GPIO2B_SL	0x0254	W	0x0000000F	VCCIO5_GPIO2B skew rate
GPIO2_IOC_GPIO2A_IE_SMT	0x0290	W	0x000000FF	VCCIO5_GPIO2A Schmitt trigger
GPIO2_IOC_GPIO2B_IE_SMT	0x0294	W	0x00000003	VCCIO5_GPIO2B Schmitt trigger
GPIO2_IOC_GPIO2A_OD	0x02D0	W	0x00000000	VCCIO5_GPIO2A open drain
GPIO2_IOC_GPIO2B_OD	0x02D4	W	0x00000000	VCCIO5_GPIO2B open drain
GPIO2_IOC_VCCIO5_POC	0x0300	W	0x00000000	VCCIO5 voltage detect

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.13.2 Detail Registers Description

GPIO2_IOC_GPIO2A_IOMUX_SEL_L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2a3_sel 3'h0: GPIO 3'h1: SDMMC1_CMD_M0 3'h2: I2S0_SDO3_SDI1 3'h3: LCD_D11
11	RO	0x0	reserved
10:8	RW	0x0	gpio2a2_sel 3'h0: GPIO 3'h1: SDMMC1_CLK_M0 3'h2: I2S0_MCLK 3'h3: LCD_D10
7	RO	0x0	reserved
6:4	RW	0x0	gpio2a1_sel 3'h0: GPIO 3'h1: SDMMC1_D0_M0 3'h2: I2S0_LRCK 3'h3: LCD_D9 3'h4: UART1_RTSN_M1 3'h5: I2C4_SCL_M0
3	RO	0x0	reserved
2:0	RW	0x0	gpio2a0_sel 3'h0: GPIO 3'h1: SDMMC1_D1_M0 3'h2: I2S0_SCLK 3'h3: LCD_D8 3'h4: UART1_CTSN_M1 4'h5: I2C4_SDA_M0

GPIO2_IOC_GPIO2A_IOMUX_SEL_H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio2a7_sel 3'h0: GPIO 3'h1: UART0_CTSN_M1 3'h2: I2S0_SDO1_SDI3 3'h3: LCD_D15 3'h4: PWM4_M1 3'h5: I2C3_SDA_M0 3'h6: PRELIGHT_TRIG_OUT
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	gpio2a6_sel 3'h0: GPIO 3'h1: UART0_RTSN_M1 3'h2: I2S0_SDO2_SDI2 3'h3: LCD_D14 3'h4: PWM2_M1 3'h5: I2C3_SCL_M0 3'h6: FLASH_TRIG_OUT
7	RO	0x0	reserved
6:4	RW	0x0	gpio2a5_sel 3'h0: GPIO 3'h1: SDMMC1_D2_M0 3'h2: I2S0_SDI0 3'h3: LCD_D13 3'h4: UART1_RX_M1
3	RO	0x0	reserved
2:0	RW	0x0	gpio2a4_sel 3'h0: GPIO 3'h1: SDMMC1_D3_M0 3'h2: I2S0_SDO0 3'h3: LCD_D12 3'h4: UART1_TX_M1

GPIO2_IOC_GPIO2B_IOMUX_SEL_L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:4	RW	0x0	gpio2b1_sel 3'h0: GPIO 3'h1: UART0_TX_M1 3'h2: I2C1_SDA_M1 3'h3: LCD_D17 3'h4: PWM6_M1 3'h5: TEST_CLK5_OUT
3	RO	0x0	reserved
2:0	RW	0x0	gpio2b0_sel 3'h0: GPIO 3'h1: UART0_RX_M1 3'h2: I2C1_SCL_M1 3'h3: LCD_D16 3'h4: PWM5_M1 3'h5: TEST_CLK4_OUT

GPIO2_IOC_GPIO2A_DS0

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x07	gpio2a1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio2a0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO2_IOC_GPIO2A_DS1

Address: Operational Base + offset (0x00C4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio2a3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio2a2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO2_IOC_GPIO2A_DS2

Address: Operational Base + offset (0x00C8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x00	gpio2a5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio2a4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO2_IOC_GPIO2A_DS3

Address: Operational Base + offset (0x00CC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x07	gpio2a7_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x00	gpio2a6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO2_IOC_GPIO2B_DS0

Address: Operational Base + offset (0x00D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x07	gpio2b1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x07	gpio2b0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO2_IOC_GPIO2A_IE

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2a7_ie GPIO2A7 IE control Active High, input buffer enable
6	RW	0x1	gpio2a6_ie GPIO2A6 IE control Active High, input buffer enable
5	RW	0x1	gpio2a5_ie GPIO2A5 IE control Active High, input buffer enable
4	RW	0x1	gpio2a4_ie GPIO2A4 IE control Active High, input buffer enable
3	RW	0x1	gpio2a3_ie GPIO2A3 IE control Active High, input buffer enable
2	RW	0x1	gpio2a2_ie GPIO2A2 IE control Active High, input buffer enable

Bit	Attr	Reset Value	Description
1	RW	0x1	gpio2a1_ie GPIO2A1 IE control Active High, input buffer enable
0	RW	0x1	gpio2a0_ie GPIO2A0 IE control Active High, input buffer enable

GPIO2_IOC_GPIO2B_IE

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x1	gpio2b1_ie GPIO2B1 IE control Active High, input buffer enable
0	RW	0x1	gpio2b0_ie GPIO2B0 IE control Active High, input buffer enable

GPIO2_IOC_GPIO2A_P

Address: Operational Base + offset (0x01D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	gpio2a7_pd Active High. Weak pull down for pad.
14	RW	0x0	gpio2a7_pu Active High. Weak pull up for pad.
13	RW	0x1	gpio2a6_pd Active High. Weak pull down for pad.
12	RW	0x0	gpio2a6_pu Active High. Weak pull up for pad.
11	RW	0x1	gpio2a5_pd Active High. Weak pull down for pad.
10	RW	0x0	gpio2a5_pu Active High. Weak pull up for pad.
9	RW	0x1	gpio2a4_pd Active High. Weak pull down for pad.
8	RW	0x0	gpio2a4_pu Active High. Weak pull up for pad.
7	RW	0x1	gpio2a3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio2a3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio2a2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio2a2_pu Active High. Weak pull up for pad.
3	RW	0x1	gpio2a1_pd Active High. Weak pull down for pad.

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio2a1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio2a0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio2a0_pu Active High. Weak pull up for pad.

GPIO2_IOC_GPIO2B_P

Address: Operational Base + offset (0x01D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio2b1_pd Active High. Weak pull down for pad.
2	RW	0x0	gpio2b1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio2b0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio2b0_pu Active High. Weak pull up for pad.

GPIO2_IOC_GPIO2A_SUS

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2a7_sus Active High. Weak pull keeper for pad.
6	RW	0x0	gpio2a6_sus Active High. Weak pull keeper for pad.
5	RW	0x0	gpio2a5_sus Active High. Weak pull keeper for pad.
4	RW	0x0	gpio2a4_sus Active High. Weak pull keeper for pad.
3	RW	0x0	gpio2a3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio2a2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio2a1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio2a0_sus Active High. Weak pull keeper for pad.

GPIO2_IOC_GPIO2B_SUS

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	gpio2b1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio2b0_sus Active High. Weak pull keeper for pad.

GPIO2_IOC_GPIO2A_SL

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	gpio2a7_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
13:12	RW	0x3	gpio2a6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio2a5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
9:8	RW	0x3	gpio2a4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:6	RW	0x3	gpio2a3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio2a2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio2a1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio2a0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO2_IOC_GPIO2B_SL

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x3	gpio2b1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio2b0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO2_IOC_GPIO2A_IE_SMT

Address: Operational Base + offset (0x0290)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio2a7_smt Active High. Receiver enable for Schmitt trigger.
6	RW	0x1	gpio2a6_smt Active High. Receiver enable for Schmitt trigger.
5	RW	0x1	gpio2a5_smt Active High. Receiver enable for Schmitt trigger.
4	RW	0x1	gpio2a4_smt Active High. Receiver enable for Schmitt trigger.
3	RW	0x1	gpio2a3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio2a2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio2a1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio2a0_smt Active High. Receiver enable for Schmitt trigger.

GPIO2_IOC_GPIO2B_IE_SMT

Address: Operational Base + offset (0x0294)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x1	gpio2b1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio2b0_smt Active High. Receiver enable for Schmitt trigger.

GPIO2_IOC_GPIO2A_OD

Address: Operational Base + offset (0x02D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio2a7_od 1'b1: For open drain functionality 1'b0: Default for regular IO
6	RW	0x0	gpio2a6_od 1'b1: For open drain functionality 1'b0: Default for regular IO
5	RW	0x0	gpio2a5_od 1'b1: For open drain functionality 1'b0: Default for regular IO
4	RW	0x0	gpio2a4_od 1'b1: For open drain functionality 1'b0: Default for regular IO
3	RW	0x0	gpio2a3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio2a2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio2a1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RW	0x0	gpio2a0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO2_IOC_GPIO2B_OD

Address: Operational Base + offset (0x02D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	gpio2b1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RW	0x0	gpio2b0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO2_IOC_VCCIO5_POC

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RO	0x0	vccio5_voldet VCCIO5 voltage detetion. 1'b0: VDDO is 1.8v 1'b1: VDDO is 3.3v
4:0	RO	0x00	reserved

4.14 GPIO3_IOC Register Description

4.14.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO3_IOC_GPIO3A_IOMUX_SEL_L	0x0040	W	0x00000000	GPIO3A iomux configuration
GPIO3_IOC_GPIO3A_IOMUX_SEL_H	0x0044	W	0x00001B00	GPIO3A iomux configuration
GPIO3_IOC_GPIO3B_IOMUX_SEL_L	0x0048	W	0x00000000	GPIO3B iomux configuration
GPIO3_IOC_GPIO3B_IOMUX_SEL_H	0x004C	W	0x00000000	GPIO3B iomux configuration
GPIO3_IOC_GPIO3C_IOMUX_SEL_L	0x0050	W	0x00000000	GPIO3C iomux configuration
GPIO3_IOC_GPIO3C_IOMUX_SEL_H	0x0054	W	0x00000000	GPIO3C iomux configuration
GPIO3_IOC_GPIO3D_IOMUX_SEL_L	0x0058	W	0x00000000	GPIO3D iomux configuration
GPIO3_IOC_GPIO3A_DS0	0x0100	W	0x00000F00	VCCIO4_GPIO3A drive strength
GPIO3_IOC_GPIO3A_DS1	0x0104	W	0x00000F0F	VCCIO4_GPIO3A drive strength
GPIO3_IOC_GPIO3A_DS2	0x0108	W	0x00000F1F	VCCIO4_GPIO3A drive strength
GPIO3_IOC_GPIO3A_DS3	0x010C	W	0x00000F0F	VCCIO4_GPIO3A drive strength
GPIO3_IOC_GPIO3C_DS2	0x0128	W	0x00000303	VCCIO7_GPIO3C drive strength
GPIO3_IOC_GPIO3C_DS3	0x012C	W	0x00000303	VCCIO7_GPIO3C drive strength
GPIO3_IOC_GPIO3D_DS0	0x0130	W	0x00000303	VCCIO7_GPIO3D drive strength
GPIO3_IOC_GPIO3D_DS1	0x0134	W	0x00000303	VCCIO7_GPIO3D drive strength
GPIO3_IOC_GPIO3A_IE	0x01A0	W	0x000000FE	VCCIO4_GPIO3A input enable
GPIO3_IOC_GPIO3C_IE	0x01A8	W	0x000000F0	VCCIO7_GPIO3C input enable
GPIO3_IOC_GPIO3D_IE	0x01AC	W	0x0000000F	VCCIO7_GPIO3D input enable
GPIO3_IOC_GPIO3A_P	0x01E0	W	0x00005654	VCCIO4_GPIO3A pull up and pull down configuration
GPIO3_IOC_GPIO3C_P	0x01E8	W	0x0000AA00	VCCIO7_GPIO3C pull up and pull down configuration
GPIO3_IOC_GPIO3D_P	0x01EC	W	0x000000AA	VCCIO7_GPIO3D pull up and pull down configuration
GPIO3_IOC_GPIO3A_SUS	0x0220	W	0x00000000	VCCIO4_GPIO3A keeper configuration
GPIO3_IOC_GPIO3C_SUS	0x0228	W	0x00000000	VCCIO7_GPIO3C keeper configuration
GPIO3_IOC_GPIO3D_SUS	0x022C	W	0x00000000	VCCIO7_GPIO3D keeper configuration
GPIO3_IOC_GPIO3A_SL	0x0260	W	0x0000FFFC	VCCIO4_GPIO3A skew rate
GPIO3_IOC_GPIO3C_SL	0x0268	W	0x0000FF00	VCCIO7_GPIO3C skew rate
GPIO3_IOC_GPIO3D_SL	0x026C	W	0x000000FF	VCCIO7_GPIO3D skew rate
GPIO3_IOC_GPIO3A_IE_SMT	0x02A0	W	0x000000FE	VCCIO4_GPIO3A Schmitt trigger
GPIO3_IOC_GPIO3C_IE_SMT	0x02A8	W	0x000000F0	VCCIO7_GPIO3C Schmitt trigger
GPIO3_IOC_GPIO3D_IE_SMT	0x02AC	W	0x0000000F	VCCIO7_GPIO3D Schmitt trigger
GPIO3_IOC_GPIO3A_OD	0x02E0	W	0x00000000	VCCIO4_GPIO3A open drain
GPIO3_IOC_GPIO3C_OD	0x02E8	W	0x00000000	VCCIO7_GPIO3C open drain
GPIO3_IOC_GPIO3D_OD	0x02EC	W	0x00000000	VCCIO7_GPIO3D open drain

Name	Offset	Size	Reset Value	Description
GPIO3_IOC_FORCE_JTAG_SDMMC	0x02F4	W	0x00000001	sdmmc and a7 jtag auto-switch enable

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.14.2 Detail Registers Description

GPIO3_IOC_GPIO3A_IOMUX_SEL_L

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3a3_sel 3'h0: GPIO 3'h1: SDMMC0_D0 3'h2: UART2_RX_M0 3'h3: TEST_CLK1_OUT 3'h4: PWM8_M0
11	RO	0x0	reserved
10:8	RW	0x0	gpio3a2_sel 3'h0: GPIO 3'h1: SDMMC0_D1 3'h2: UART2_TX_M0 3'h3: TEST_CLK0_OUT 3'h4: PWM9_M0
7	RO	0x0	reserved
6:4	RW	0x0	gpio3a1_sel 3'h0: GPIO 3'h1: SDMMC0_DET
3:0	RO	0x0	reserved

GPIO3_IOC_GPIO3A_IOMUX_SEL_H

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
14:12	RW	0x3	gpio3a7_sel 3'h0: GPIO 3'h1: SDMMC0_D2 3'h2: UART5_RX_M0 3'h3: A7_JTAG_TCK_M0 3'h4: HPMCU_JTAG_TCK_M1
11	RO	0x0	reserved
10:8	RW	0x3	gpio3a6_sel 3'h0: GPIO 3'h1: SDMMC0_D3 3'h2: UART5_TX_M0 3'h3: A7_JTAG_TMS_M0 3'h4: HPMCU_JTAG_TMS_M1
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio3a5_sel 3'h0: GPIO 3'h1: SDMMC0_CMD 3'h2: UART5_CTSN_M0 3'h3: I2C0_SDA_M2 3'h4: LPMCU_JTAG_TMS_M1 3'h5: PWM11_IR_M0
3	RO	0x0	reserved
2:0	RW	0x0	gpio3a4_sel 3'h0: GPIO 3'h1: SDMMC0_CLK 3'h2: UART5_RTSN_M0 3'h3: I2C0_SCL_M2 3'h4: LPMCU_JTAG_TCK_M1 3'h5: PWM10_M0

GPIO3_IOC_GPIO3B_IOMUX_SEL_L

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3b3_sel 3'h0: GPIO 3'h1: VICAP_D3_M0 3'h2: MIPICSI_LVDSRX_CK1P
11	RO	0x0	reserved
10:8	RW	0x0	gpio3b2_sel 3'h0: GPIO 3'h1: VICAP_D2_M0 3'h2: MIPICSI_LVDSRX_CK1N
7	RO	0x0	reserved
6:4	RW	0x0	gpio3b1_sel 3'h0: GPIO 3'h1: VICAP_D1_M0 3'h2: MIPICSI_LVDSRX_D3P
3	RO	0x0	reserved
2:0	RW	0x0	gpio3b0_sel 3'h0: GPIO 3'h1: VICAP_D0_M0 3'h2: MIPICSI_LVDSRX_D3N

GPIO3_IOC_GPIO3B_IOMUX_SEL_H

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RW	0x0	gpio3b7_sel 3'h0: GPIO 3'h1: VICAP_D7_M0 3'h2: MIPICSI_LVDSRX_D1P
11	RO	0x0	reserved
10:8	RW	0x0	gpio3b6_sel 3'h0: GPIO 3'h1: VICAP_D6_M0 3'h2: MIPICSI_LVDSRX_D1N
7	RO	0x0	reserved
6:4	RW	0x0	gpio3b5_sel 3'h0: GPIO 3'h1: VICAP_D5_M0 3'h2: MIPICSI_LVDSRX_D2P
3	RO	0x0	reserved
2:0	RW	0x0	gpio3b4_sel 3'h0: GPIO 3'h1: VICAP_D4_M0 3'h2: MIPICSI_LVDSRX_D2N

GPIO3_IOC_GPIO3C_IOMUX_SEL_L

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3c3_sel 3'h0: GPIO 3'h1: VICAP_HSYNC_M0 3'h2: MIPICSI_LVDSRX_D0P
11	RO	0x0	reserved
10:8	RW	0x0	gpio3c2_sel 3'h0: GPIO 3'h1: VICAP_CLKIN_M0 3'h2: MIPICSI_LVDSRX_D0N
7	RO	0x0	reserved
6:4	RW	0x0	gpio3c1_sel 3'h0: GPIO 3'h1: VICAP_D9_M0 3'h2: MIPICSI_LVDSRX_CK0P
3	RO	0x0	reserved
2:0	RW	0x0	gpio3c0_sel 3'h0: GPIO 3'h1: VICAP_D8_M0 3'h2: MIPICSI_LVDSRX_CK0N

GPIO3_IOC_GPIO3C_IOMUX_SEL_H

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14:12	RW	0x0	gpio3c7_sel 3'h0: GPIO 3'h1: VICAP_D11 3'h2: UART5_TX_M2 3'h3: I2C4_SCL_M2
11	RO	0x0	reserved
10:8	RW	0x0	gpio3c6_sel 3'h0: GPIO 3'h1: VICAP_D10 3'h2: PWM7_IR_M2 3'h3: MIPICSI_REFCLK_OUT1
7	RO	0x0	reserved
6:4	RW	0x0	gpio3c5_sel 3'h0: GPIO 3'h1: VICAP_VSYNC_M0
3	RO	0x0	reserved
2:0	RW	0x0	gpio3c4_sel 3'h0: GPIO 3'h1: VICAP_CLKOUT_M0 3'h2: MIPICSI_REFCLK_OUT0

GPIO3_IOC_GPIO3D_IOMUX_SEL_L

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio3d3_sel 3'h0: GPIO 3'h1: VICAP_D15 3'h2: PWM1_M2
11	RO	0x0	reserved
10:8	RW	0x0	gpio3d2_sel 3'h0: GPIO 3'h1: VICAP_D14 3'h2: UART5_CTSN_M2 3'h3: I2C3_SDA_M2
7	RO	0x0	reserved
6:4	RW	0x0	gpio3d1_sel 3'h0: GPIO 3'h1: VICAP_D13 3'h2: UART5_RTSN_M2 3'h3: I2C3_SCL_M2
3	RO	0x0	reserved
2:0	RW	0x0	gpio3d0_sel 3'h0: GPIO 3'h1: VICAP_D12 3'h2: UART5_RX_M2 3'h3: I2C4_SDA_M2

GPIO3_IOC_GPIO3A_DS0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio3a1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:0	RO	0x00	reserved

GPIO3_IOC_GPIO3A_DS1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio3a3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio3a2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3A_DS2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio3a5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x1f	gpio3a4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3A_DS3

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x0f	gpio3a7_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio3a6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3C_DS2

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	gpio3c5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x03	gpio3c4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3C_DS3

Address: Operational Base + offset (0x012C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	gpio3c7_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x03	gpio3c6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3D_DS0

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x03	gpio3d1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x03	gpio3d0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3D_DS1

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	gpio3d3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x03	gpio3d2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO3_IOC_GPIO3A_IE

Address: Operational Base + offset (0x01A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3a7_ie Active High input buffer enable
6	RW	0x1	gpio3a6_ie Active High input buffer enable
5	RW	0x1	gpio3a5_ie Active High input buffer enable
4	RW	0x1	gpio3a4_ie Active High input buffer enable
3	RW	0x1	gpio3a3_ie Active High input buffer enable
2	RW	0x1	gpio3a2_ie Active High input buffer enable
1	RW	0x1	gpio3a1_ie Active High input buffer enable
0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_IE

Address: Operational Base + offset (0x01A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3c7_ie Active High input buffer enable
6	RW	0x1	gpio3c6_ie Active High input buffer enable
5	RW	0x1	gpio3c5_ie Active High input buffer enable
4	RW	0x1	gpio3c4_ie Active High input buffer enable
3:0	RO	0x0	reserved

GPIO3_IOC_GPIO3D_IE

Address: Operational Base + offset (0x01AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio3d3_ie Active High input buffer enable
2	RW	0x1	gpio3d2_ie Active High input buffer enable
1	RW	0x1	gpio3d1_ie Active High input buffer enable
0	RW	0x1	gpio3d0_ie Active High input buffer enable

GPIO3_IOC_GPIO3A_P

Address: Operational Base + offset (0x01E0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	gpio3a7_pd Active High. Weak pull down for pad.
14	RW	0x1	gpio3a7_pu Active High. Weak pull up for pad.
13	RW	0x0	gpio3a6_pd Active High. Weak pull down for pad.
12	RW	0x1	gpio3a6_pu Active High. Weak pull up for pad.
11	RW	0x0	gpio3a5_pd Active High. Weak pull down for pad.
10	RW	0x1	gpio3a5_pu Active High. Weak pull up for pad.
9	RW	0x1	gpio3a4_pd Active High. Weak pull down for pad.
8	RW	0x0	gpio3a4_pu Active High. Weak pull up for pad.

Bit	Attr	Reset Value	Description
7	RW	0x0	gpio3a3_pd Active High. Weak pull down for pad.
6	RW	0x1	gpio3a3_pu Active High. Weak pull up for pad.
5	RW	0x0	gpio3a2_pd Active High. Weak pull down for pad.
4	RW	0x1	gpio3a2_pu Active High. Weak pull up for pad.
3	RW	0x0	gpio3a1_pd Active High. Weak pull down for pad.
2	RW	0x1	gpio3a1_pu Active High. Weak pull up for pad.
1:0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_P

Address: Operational Base + offset (0x01E8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	gpio3c7_pd Active High. Weak pull down for pad.
14	RW	0x0	gpio3c7_pu Active High. Weak pull up for pad.
13	RW	0x1	gpio3c6_pd Active High. Weak pull down for pad.
12	RW	0x0	gpio3c6_pu Active High. Weak pull up for pad.
11	RW	0x1	gpio3c5_pd Active High. Weak pull down for pad.
10	RW	0x0	gpio3c5_pu Active High. Weak pull up for pad.
9	RW	0x1	gpio3c4_pd Active High. Weak pull down for pad.
8	RW	0x0	gpio3c4_pu Active High. Weak pull up for pad.
7:0	RO	0x00	reserved

GPIO3_IOC_GPIO3D_P

Address: Operational Base + offset (0x01EC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3d3_pd Active High. Weak pull down for pad.
6	RW	0x0	gpio3d3_pu Active High. Weak pull up for pad.
5	RW	0x1	gpio3d2_pd Active High. Weak pull down for pad.
4	RW	0x0	gpio3d2_pu Active High. Weak pull up for pad.

Bit	Attr	Reset Value	Description
3	RW	0x1	gpio3d1_pd Active High. Weak pull down for pad.
2	RW	0x0	gpio3d1_pu Active High. Weak pull up for pad.
1	RW	0x1	gpio3d0_pd Active High. Weak pull down for pad.
0	RW	0x0	gpio3d0_pu Active High. Weak pull up for pad.

GPIO3_IOC_GPIO3A_SUS

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3a7_sus Active High. Weak pull keeper for pad.
6	RW	0x0	gpio3a6_sus Active High. Weak pull keeper for pad.
5	RW	0x0	gpio3a5_sus Active High. Weak pull keeper for pad.
4	RW	0x0	gpio3a4_sus Active High. Weak pull keeper for pad.
3	RW	0x0	gpio3a3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio3a2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio3a1_sus Active High. Weak pull keeper for pad.
0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_SUS

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio3c7_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio3c6_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio3c5_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio3c4_sus Active High. Weak pull keeper for pad.

GPIO3_IOC_GPIO3D_SUS

Address: Operational Base + offset (0x022C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio3d3_sus Active High. Weak pull keeper for pad.
2	RW	0x0	gpio3d2_sus Active High. Weak pull keeper for pad.
1	RW	0x0	gpio3d1_sus Active High. Weak pull keeper for pad.
0	RW	0x0	gpio3d0_sus Active High. Weak pull keeper for pad.

GPIO3_IOC_GPIO3A_SL

Address: Operational Base + offset (0x0260)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	gpio3a7_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
13:12	RW	0x3	gpio3a6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio3a5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
9:8	RW	0x3	gpio3a4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:6	RW	0x3	gpio3a3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio3a2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio3a1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_SL

Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	gpio3c7_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
13:12	RW	0x3	gpio3c6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio3c5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
9:8	RW	0x3	gpio3c4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:0	RO	0x00	reserved

GPIO3_IOC_GPIO3D_SL

Address: Operational Base + offset (0x026C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7:6	RW	0x3	gpio3d3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio3d2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio3d1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio3d0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO3_IOC_GPIO3A_IE_SMT

Address: Operational Base + offset (0x02A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved

Bit	Attr	Reset Value	Description
7	RW	0x1	gpio3a7_smt Active High. Receiver enable for Schmitt trigger.
6	RW	0x1	gpio3a6_smt Active High. Receiver enable for Schmitt trigger.
5	RW	0x1	gpio3a5_smt Active High. Receiver enable for Schmitt trigger.
4	RW	0x1	gpio3a4_smt Active High. Receiver enable for Schmitt trigger.
3	RW	0x1	gpio3a3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio3a2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio3a1_smt Active High. Receiver enable for Schmitt trigger.
0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_IE_SMT

Address: Operational Base + offset (0x02A8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio3c7_smt Active High. Receiver enable for Schmitt trigger.
6	RW	0x1	gpio3c6_smt Active High. Receiver enable for Schmitt trigger.
5	RW	0x1	gpio3c5_smt Active High. Receiver enable for Schmitt trigger.
4	RW	0x1	gpio3c4_smt Active High. Receiver enable for Schmitt trigger.
3:0	RO	0x0	reserved

GPIO3_IOC_GPIO3D_IE_SMT

Address: Operational Base + offset (0x02AC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio3d3_smt Active High. Receiver enable for Schmitt trigger.
2	RW	0x1	gpio3d2_smt Active High. Receiver enable for Schmitt trigger.
1	RW	0x1	gpio3d1_smt Active High. Receiver enable for Schmitt trigger.
0	RW	0x1	gpio3d0_smt Active High. Receiver enable for Schmitt trigger.

GPIO3_IOC_GPIO3A_OD

Address: Operational Base + offset (0x02E0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3a7_od 1'b1: For open drain functionality 1'b0: Default for regular IO
6	RW	0x0	gpio3a6_od 1'b1: For open drain functionality 1'b0: Default for regular IO
5	RW	0x0	gpio3a5_od 1'b1: For open drain functionality 1'b0: Default for regular IO
4	RW	0x0	gpio3a4_od 1'b1: For open drain functionality 1'b0: Default for regular IO
3	RW	0x0	gpio3a3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio3a2_od 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio3a1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RO	0x0	reserved

GPIO3_IOC_GPIO3C_OD

Address: Operational Base + offset (0x02E8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio3c7_od 1'b1: For open drain functionality 1'b0: Default for regular IO
6	RW	0x0	gpio3c6_od 1'b1: For open drain functionality 1'b0: Default for regular IO
5	RW	0x0	gpio3c5_od 1'b1: For open drain functionality 1'b0: Default for regular IO
4	RW	0x0	gpio3c4_od 1'b1: For open drain functionality 1'b0: Default for regular IO
3:0	RO	0x0	reserved

GPIO3_IOC_GPIO3D_OD

Address: Operational Base + offset (0x02EC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpio2d3_od 1'b1: For open drain functionality 1'b0: Default for regular IO
2	RW	0x0	gpio3d2_0d 1'b1: For open drain functionality 1'b0: Default for regular IO
1	RW	0x0	gpio3d1_od 1'b1: For open drain functionality 1'b0: Default for regular IO
0	RW	0x0	gpio3d0_od 1'b1: For open drain functionality 1'b0: Default for regular IO

GPIO3_IOC_FORCE_JTAG_SDMMC

Address: Operational Base + offset (0x02F4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x1	jtag_sdmmc_auto_switch_en This bit is used to enable sdmmc auto-switch to a7_jtag_m0,active high.

4.15 GPIO4_IOC Register Description

4.15.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO4_IOC_GPIO4A_IOMUX_SEL_L	0x0000	W	0x00000000	GPIO4A iomux configuration register 0
GPIO4_IOC_GPIO4A_IOMUX_SEL_H	0x0004	W	0x00000000	GPIO4A iomux configuration register 1
GPIO4_IOC_GPIO4B_IOMUX_SEL_L	0x0008	W	0x00000000	GPIO4B iomux configuration register 0
GPIO4_IOC_GPIO4C_IOMUX_SEL_L	0x0010	W	0x00000000	GPIO4B iomux configuration register 0
GPIO4_IOC_GPIO4A_DS0	0x0020	W	0x00000F0F	GPIO4A drive strength register 0
GPIO4_IOC_GPIO4A_DS1	0x0024	W	0x00000F0F	GPIO4A drive strength register 1
GPIO4_IOC_GPIO4A_DS2	0x0028	W	0x00000F0F	GPIO4A drive strength register 2
GPIO4_IOC_GPIO4A_DS3	0x002C	W	0x00000F0F	GPIO4A drive strength register 3
GPIO4_IOC_GPIO4B_DS0	0x0030	W	0x00000F0F	GPIO4B drive strength register 0
GPIO4_IOC_GPIO4A_IE	0x0060	W	0x000000FF	GPIO4A input enable register
GPIO4_IOC_GPIO4B_IE	0x0064	W	0x00000003	GPIO4B input enable register
GPIO4_IOC_GPIO4A_P	0x0070	W	0x00005555	GPIO4A pull up and pull down configuration register
GPIO4_IOC_GPIO4B_P	0x0074	W	0x00000009	GPIO4B pull up and pull down configuration register

Name	Offset	Size	Reset Value	Description
GPIO4_IOC_GPIO4A_SUS	0x0080	W	0x00000000	GPIO4A keeper configuration register
GPIO4_IOC_GPIO4B_SUS	0x0084	W	0x00000000	GPIO4A keeper configuration register
GPIO4_IOC_GPIO4A_SL	0x0090	W	0x0000FFFF	GPIO4A skew rate register
GPIO4_IOC_GPIO4B_SL	0x0094	W	0x0000000F	GPIO4B skew rate register
GPIO4_IOC_GPIO4A_IE_SMT	0x00A0	W	0x000000FF	GPIO4A Schmitt trigger register
GPIO4_IOC_GPIO4B_IE_SMT	0x00A4	W	0x00000003	GPIO4B Schmitt trigger register
GPIO4_IOC_GPIO4A_OD	0x00B0	W	0x00000000	GPIO4A open drain register
GPIO4_IOC_GPIO4B_OD	0x00B4	W	0x00000000	GPIO4B open drain register
GPIO4_IOC_SARADC_IO_CON	0x00C0	W	0x00001300	GPIO4C0/GPIO4C1 control
GPIO4_IOC_VCCIO3_VOL_CON	0x0200	W	0x00000004	VCCIO3 voltage control register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

4.15.2 Detail Registers Description

GPIO4_IOC_GPIO4A_IOMUX_SEL_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio4a3_sel 3'h0: GPIO4A3 3'h1: EMMC_D1 3'h2: FSPI_D1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio4a2_sel 3'h0: GPIO4A2 3'h1: EMMC_D2 3'h2: FSPI_D2 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio4a1_sel 3'h0: GPIO4A1 3'h1: EMMC_D6 3'h2: SPI1_MOSI_M0 3'h3: UART0_TX_M2 3'h4: I2C0_SCL_M1 3'h5: TEST_CLK3_OUT Others: Reserved
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio4a0_sel 3'h0: GPIO4A0 3'h1: EMMC_D7 3'h2: SPI1_MISO_M0 3'h3: UART0_RX_M2 3'h4: I2C0_SDA_M1 3'h5: TEST_CLK2_OUT Others: Reserved

GPIO4_IOC_GPIO4A_IOMUX_SEL_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio4a7_sel 3'h0: GPIO4A7 3'h1: EMMC_D5 3'h2: SPI1_CLK_M0 3'h3: UART1_RX_M2 3'h4: I2C2_SCL_M1 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio4a6_sel 3'h0: GPIO4A6 3'h1: EMMC_D3 3'h2: FSPI_D3 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio4a5_sel 3'h0: GPIO4A5 3'h1: EMMC_D4 3'h2: SPI1_CS0n_M0 3'h3: UART1_TX_M2 3'h4: I2C2_SDA_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio4a4_sel 3'h0: GPIO4A4 3'h1: EMMC_D0 3'h2: FSPI_D0 Others: Reserved

GPIO4_IOC_GPIO4B_IOMUX_SEL_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio4b1_sel 3'h0: GPIO4B1 3'h1: EMMC_CLK 3'h2: FSPI_CLK Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio4b0_sel 3'h0: GPIO4B0 3'h1: EMMC_CMD 3'h2: FSPI_CS0n Others: Reserved

GPIO4_IOC_GPIO4C_IOMUX_SEL_L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x000	reserved
6:4	RW	0x0	gpio4c1_sel 3'h0: GPIO4C1 3'h1: ADC_IN1 3'h2: PWM1_M1 Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio4c0_sel 3'h0: GPIO4C0 3'h1: ADC_IN0 Others: Reserved

GPIO4_IOC_GPIO4A_DS0

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio4a1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio4a0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO4_IOC_GPIO4A_DS1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio4a3_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio4a2_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO4_IOC_GPIO4A_DS2

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio4a5_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio4a4_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO4_IOC_GPIO4A_DS3

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio4a7_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio4a6_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO4_IOC_GPIO4B_DS0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x0f	gpio4b1_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.
7:6	RO	0x0	reserved
5:0	RW	0x0f	gpio4b0_ds The drive strength of IO which can be programmed from 2mA to 12mA, each bit controls 2mA of driving ability. The weakest setting is 6'b000001 and the strongest setting is 6'b111111.

GPIO4_IOC_GPIO4A_IE

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio4a7_ie Receiver enable. 1'b0: Disable 1'b1: Enable
6	RW	0x1	gpio4a6_ie Receiver enable. 1'b0: Disable 1'b1: Enable
5	RW	0x1	gpio4a5_ie Receiver enable. 1'b0: Disable 1'b1: Enable
4	RW	0x1	gpio4a4_ie Receiver enable. 1'b0: Disable 1'b1: Enable
3	RW	0x1	gpio4a3_ie Receiver enable. 1'b0: Disable 1'b1: Enable
2	RW	0x1	gpio4a2_ie Receiver enable. 1'b0: Disable 1'b1: Enable
1	RW	0x1	gpio4a1_ie Receiver enable. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4a0_ie Receiver enable. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4B_IE

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x1	gpio4b1_ie Receiver enable. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4b0_ie Receiver enable. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4A_P

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	gpio4a7_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
14	RW	0x1	gpio4a7_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
13	RW	0x0	gpio4a6_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
12	RW	0x1	gpio4a6_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
11	RW	0x0	gpio4a5_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
10	RW	0x1	gpio4a5_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
9	RW	0x0	gpio4a4_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
8	RW	0x1	gpio4a4_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
7	RW	0x0	gpio4a3_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
6	RW	0x1	gpio4a3_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
5	RW	0x0	gpio4a2_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
4	RW	0x1	gpio4a2_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
3	RW	0x0	gpio4a1_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
2	RW	0x1	gpio4a1_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable
1	RW	0x0	gpio4a0_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4a0_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4B_P

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x1	gpio4b1_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
2	RW	0x0	gpio4b1_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio4b0_pd Weak pull down for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4b0_pu Weak pull up for pad. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4A_SUS

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio4a7_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
6	RW	0x0	gpio4a6_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
5	RW	0x0	gpio4a5_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
4	RW	0x0	gpio4a4_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
3	RW	0x0	gpio4a3_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
2	RW	0x0	gpio4a2_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
1	RW	0x0	gpio4a1_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x0	gpio4a0_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4B_SUS

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	gpio4a1_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable
0	RW	0x0	gpio4a0_sus Weak pull keeper for pad. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4A_SL

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	gpio4a7_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
13:12	RW	0x3	gpio4a6_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
11:10	RW	0x3	gpio4a5_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
9:8	RW	0x3	gpio4a4_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
7:6	RW	0x3	gpio4a3_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
5:4	RW	0x3	gpio4a2_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
3:2	RW	0x3	gpio4a1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio4a0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO4_IOC_GPIO4B_SL

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3:2	RW	0x3	gpio4b1_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.
1:0	RW	0x3	gpio4b0_sl Slew rate control for the driver section, while driving pad, 2'b11 is fastest and 2'b00 is slowest. Always set to 2'b11 in normal operation.

GPIO4_IOC_GPIO4A_IE_SMT

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x1	gpio4a7_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
6	RW	0x1	gpio4a6_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
5	RW	0x1	gpio4a5_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
4	RW	0x1	gpio4a4_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
3	RW	0x1	gpio4a3_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
2	RW	0x1	gpio4a2_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
1	RW	0x1	gpio4a1_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4a0_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4B_IE_SMT

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x1	gpio4b1_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable
0	RW	0x1	gpio4b0_smt Receiver enable for schmitt trigger. 1'b0: Disable 1'b1: Enable

GPIO4_IOC_GPIO4A_OD

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	gpio4a7_od 1'b0: Default for regular IO 1'b1: For open drain functionality
6	RW	0x0	gpio4a6_od 1'b0: Default for regular IO 1'b1: For open drain functionality
5	RW	0x0	gpio4a5_od 1'b0: Default for regular IO 1'b1: For open drain functionality
4	RW	0x0	gpio4a4_od 1'b0: Default for regular IO 1'b1: For open drain functionality
3	RW	0x0	gpio4a3_od 1'b0: Default for regular IO 1'b1: For open drain functionality
2	RW	0x0	gpio4a2_od 1'b0: Default for regular IO 1'b1: For open drain functionality
1	RW	0x0	gpio4a1_od 1'b0: Default for regular IO 1'b1: For open drain functionality
0	RW	0x0	gpio4a0_od 1'b0: Default for regular IO 1'b1: For open drain functionality

GPIO4_IOC_GPIO4B_OD

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	gpio4b1_od 1'b0: Default for regular IO 1'b1: For open drain functionality
0	RW	0x0	gpio4b0_od 1'b0: Default for regular IO 1'b1: For open drain functionality

GPIO4_IOC_SARADC_IO_CON

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	saradc_io_ps SARADC IO Pull selector signal, high active. PE=1'b0: High Z PE=1'b1: PS=1'b1, pull up PE=1'b1: PS=1'b0, pull down Shared by two SARADC IO.
13	RW	0x0	saradc_io_pe SARADC IO Pull selector signal, high active. PE=1'b0: High Z PE=1'b1: PS=1'b1, pull up PE=1'b1: PS=1'b0, pull down Shared by two SARADC IO.
12	RW	0x1	saradc_io_sl SARADC IO Slew rate signal, high active. Shared by two SARADC IO.
11	RW	0x0	saradc_io_ds1 SARADC IO DS1 signal, driving selector. (DS1,DS0) = 2'b00: 10.8mA (DS1,DS0) = 2'b01: 21.5mA (DS1,DS0) = 2'b10: 32.1mA (DS1,DS0) = 2'b11: 42.4mA Shared by two SARADC IO.
10	RW	0x0	saradc_io_ds0 SARADC IO DS1 signal, driving selector. (DS1,DS0) = 2'b00: 10.8mA (DS1,DS0) = 2'b01: 21.5mA (DS1,DS0) = 2'b10: 32.1mA (DS1,DS0) = 2'b11: 42.4mA Shared by two SARADC IO.
9	RW	0x1	saradc_io_st1 SARADC IO Schmitt trigger enable signal, high active. Shared by two SARADC IO.
8	RW	0x1	saradc_io_st0 SARADC IO Schmitt trigger enable signal, high active. Shared by two SARADC IO.

Bit	Attr	Reset Value	Description
7:4	RO	0x0	reserved
3	RW	0x0	saradc_io_ie1 SARADC IO1 input enable signal, high active.
2	RW	0x0	saradc_io_ie0 SARADC IO0 input enable signal, high active.
1	RW	0x0	saradc_io_he1 SARADC IO1 Hold enable signal, high active.
0	RW	0x0	saradc_io_he0 SARADC IO0 Hold enable signal, high active.

GPIO4_IOC_VCCIO3_VOL_CON

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x000	reserved
4	RO	0x0	vccio3_vd VCCIO3 voltage detetion. 1'b0: VDDO is 1.8v 1'b1: VDDO is 3.3v
3	RW	0x0	vccio3_iddq_sw Software configuration for VCCIO3 IDDQ pin. Set 1 for IDDQ testing.
2	RW	0x1	vccio3_cle_sw Software configuration for VCCIO3 CLE pin. Set to 1 for VDDO=3.3v operation.
1	RW	0x0	vccio3_v18_sw Software configuration for VCCIO3 V18 pin. Set to 1 for VDDO=1.8v operation.
0	RW	0x0	vccio3_vsel VCCOI3 voltage pin(V18/CLE/IDDQ) controller selection. 1'b0: Voltage pin controlled by hardware 1'b1: Voltage pin controlled by software

Chapter 5 Cortex-A7

5.1 Overview

The Cortex-A7 subsystem of the device is based on the ARMv7-A architecture. The Cortex-A7 has one Cortex-A7 central processing unit (CPU). Each processor has 32KB of Level 1 (L1) instruction cache, 32KB of L1 data cache, NEON and FPU. The Cortex-A7 also includes 128KB of Level 2 (L2) cache, Snoop Control Unit(SCU), Generic Interrupt Controller (GIC), and standard CoreSight components to support SMP debug and emulation.
The key features of the Cortex-A7 include:

- ARM Cortex-A7
  - Full implements the ARMv7-A architecture profile that includes the Advanced SIMD and VFP Extensions
  - 1 Cortex-A7 processor cores
  - 32KB L1 I-Cache and 32KB L1 D-Cache per CPU
  - In-order pipeline with direct and indirect branch prediction
  - Harvard L1 memory system with a Memory Management Unit (MMU)
  - SCU ensures memory coherency
  - Interrupt controller with 128 hardware interrupt inputs
- 128KB L2 cache
  - Fixed line length of 64 bytes
  - Physically indexed and tagged cache
  - 8-way set-associative cache structure
  - Pseudo-random cache replacement policy

5.2 Block Diagram

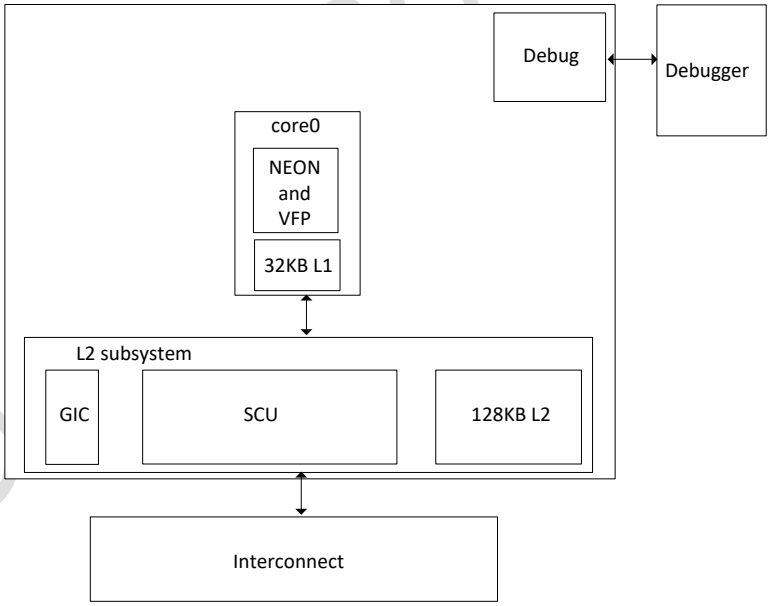


Fig. 5-1 Core Subsystem Architecture

5.3 Function Description

Please refer to the document *Cortex-A7 MPCore Technical Reference Manual* for the CPU detail description.

5.4 Register Description

Please refer to the document *Cortex-A7 MPCore Technical Reference Manual* for the CPU detail description.

Chapter 6 Embedded SRAM

6.1 Overview

There are two embedded SRAMs, SYSTEM_SRAM and PMU_SRAM.

- SYSTEM_SRAM
  - Provide 256KB access space
  - Support security and non-security access
  - Secure or non-secure space is software programmable
- PMU_SRAM
  - Provide 8KB access space
  - Support secure access only

6.2 Block Diagram

The following figure shows the block diagram of Share Memory.

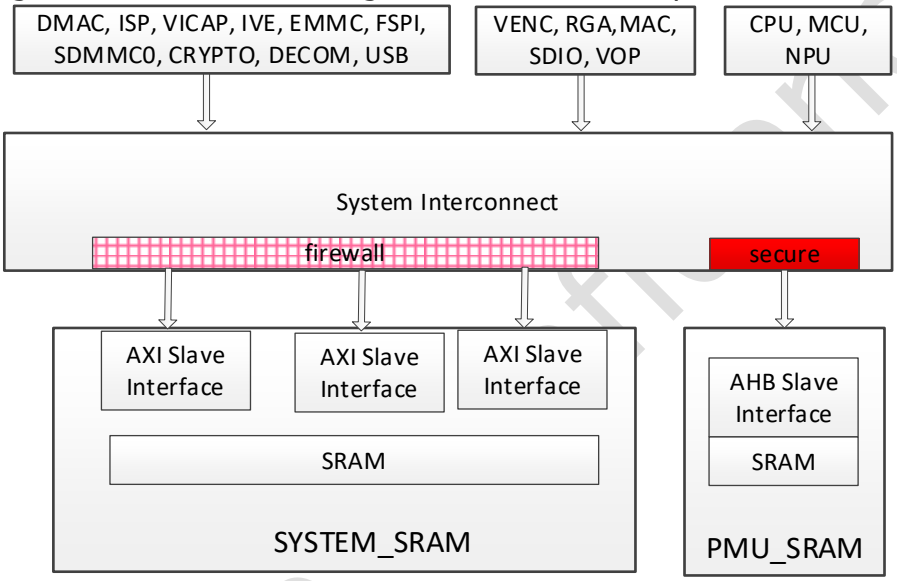


Fig. 6-1 Block Diagram of SRAM

6.3 Function Description

6.3.1 PMU SRAM Access

PMU SRAM is secure only and base address is 0xFF670000. Only CPU, MCU and CRYPTO can access PMU SRAM

6.3.2 SYSTEM SRAM Access

All the masters that can access SYSTEM SRAM is shown in the above figure. The security is decided by the firewall and can be set by SRAM firewall. There are 8 regions can configured to secure or non-secure and each master can be disabled.

Chapter 7 PMU

7.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RV1106. The RV1106 PMU is dedicated for managing the power of the whole chip.

PMU supports the following features:

- Support multi voltage domains: VD_CORE, VD_LOGIC, VD_PMU
- Support BIU idle operations: BIU_MSCH, BIU_DDR, BIU_NPU, BIU_NPU_ACLK, BIU_VI, BIU_VO, BIU_PERI, BIU_CRU, BIU_CPU, BIU_VENC_COM, BIU_VENC
- Support to send idle request to BIU
- Support global interrupt disable in low power mode
- Support low frequency clock source from PMU PVTM or RTC
- Support PMU clock switch to low frequency clock in low power mode
- Support PLLs power down/up by hardware in low power mode
- Support OSC enable/disable request in low power mode
- Support to clamp all VD_PMU input before power off VD_LOGIC in low power mode
- Support wakeup reset control in power off mode
- Support DDR self-refresh in low power mode
- Support block clock auto gating in low power mode
- Support varies configurable wakeup source for low power mode

7.2 Block Diagram

The following figure is the PMU block diagram. The PMU includes 3 following sections:

- APB Interface and Register: Provide AMBA APB interface for register read and write
- System Power State Control: Provide power management for various low power modes
- Power Gating Control: Provide power gating control for power domains

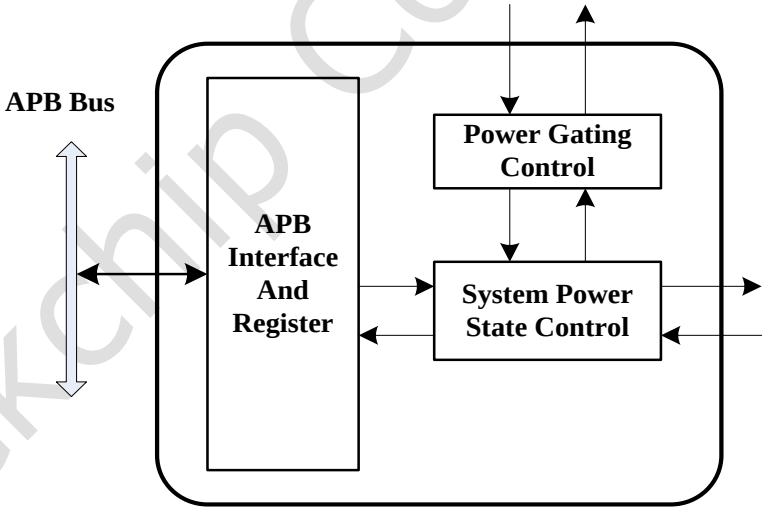


Fig. 7-1 PMU Bock Diagram

7.3 Function Description

7.3.1 Block Description

The following table lists IPs in every block or voltage domain.

Table 7-1 Block Description Summary

Voltage Domain	Block Partition	Description
VD_CORE	CORE_WRAPPER	CORE_BIU CRU_CORE CORE_GRF CORE_SGRF CORTEXA7 DAP_LITE HPMCU CORE_MAILBOX CORE_PVTM CORE_PVTPLL
VDD_PMU	PMU_WRAPPER	PMU_BIU CRU_PMU PMU_GRF PMU_SGRF PMU_SRAM PMU_PVTM PMU_WDT PMU_MAILBOX PMU LPMCU I2C1 HPTIMER DDR_FAIL_SAFE GPIO0 PMUIO_IOC NPOR PAD_RING_PMU APLL CPLL DPLL GPLL
VDD_LOGIC	CRU_WRAPPER	CRU_NIU CRU PVTPLL0/1 PVTPLL_GRF APLL_WRAPPER CPLL_WRAPPER DPLL_WRAPPER GPLL_WRAPPER
	DDR_WRAPPER	MSCH CRU_DDR CRU_SUBDDR DDR_GRF DDR_CTL DFI_CTL DFI_RS HW_LP_CTRL DDR_MONITOR SYSTEM_MEMORY DDRPHY_WRAPPER DDRPHY
	PERI_WRAPPER	PERI_BIU CRU_PERI PERI_GRF PERI_SGRF

Voltage Domain	Block Partition	Description
		DMAC SCRYPTO DECOM IVE BOOTROM SEC_TRNG_S SEC_TRNG_NS I2S SAI DSM ACODEC_WRAPPER SFC EMMC USBOTG USBPHY_WRAPPER WDT_S WDT_NS TIMER_S(TIMER_2CH) TIMER_NS(TIMER_6CH) UART0/1/2/3/4/5 I2C0/2/3/4 SPI1 SARADC_CON SARADC_WRAPPER GPIO4 VCCIO3_IOC DFT2APB TM_DECODER
	NPU_WRAPPER	NPU_BIU CRU_NPU NPU_GRF NPU_SGRF RKNN
	VENC_WRAPPER	VENC_BIU CRU_VENC VENC_GRF VENC_SGRF VENC VEPU_PP VEPU_DVBM SPI0 GPIO1 VENCIO_IOC
	VI_WRAPPER	VI_BIU CRU_VI VI_GRF VI_SGRF VICAP CSIHOST0/1 CSIPHY_WRAPPER ISP SDMMC RTC_TEST GPIO3 VIIO_IOC

Voltage Domain	Block Partition	Description
	VO_WRAPPER	VO_BIU CRU_VO VO_GRF VO_SGRF VOP RGA SDIO GMAC MACPHY_WRAPPER OTPC_S OTPC_NS OTP_WRAPPER TSADC TSADC_WRAPPER GIO2 VCCIO5_IOC
	TOP	CSIPHY USBPHY SARADC TSADC OTP MACPHY ACODEC PAD_RING_TOP
VDD_RTC	RTC_WRAPPER	RTC

7.3.2 Operation Mode

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 registers. For example, LPMCU can write PMU_BIU_IDLE_SFTCON register to determine BIU idle 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 be delayed to execute after FSM running in particular phase.
To enter low power mode, after setting some power configurations, the PMU_PWR_CON[0] bit must be set 1 to enable PMU FSM. Then CPU needs to execute a WFI command to perform ready signal if PMU_SCU_PWR_CON[2] is 0. After PMU detects CPU in WFI status, 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 RV1106 chip.

7.4 Register Description

7.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_VERSION	0x0000	W	0x030A0000	PMU version register
PMU_PWR_CON	0x0004	W	0x00000000	PMU power control register
PMU_GLB_POWER_STS	0x0008	W	0x00000000	PMU power status register
PMU_INT_MASK_CON	0x000C	W	0x00000000	Interrupt mask control register
PMU_WAKEUP_INT_CON	0x0010	W	0x00000000	Wakeup interrupt control register
PMU_WAKEUP_INT_ST	0x0014	W	0x00000000	Wakeup interrupt status register
PMU_PMIC_STABLE_CNT	0x0024	W	0x000FFFFF	PMIC stable counter register

Name	Offset	Size	Reset Value	Description
PMU_OSC_STABLE_CNT	0x0028	W	0x000FFFFF	OSC stable counter register
PMU_WAKEUP_RSTCLR_CNT	0x002C	W	0x000FFFFF	Wakeup reset clear counter register
PMU_PLL_LOCK_CNT	0x0030	W	0x000FFFFF	PLL lock counter register
PMU_WAKEUP_TIMEOUT_CNT	0x0048	W	0x00005DC0	WAKEUP timeout counter register
PMU_PWM_SWITCH_CNT	0x004C	W	0x000FFFFF	PWM switch stable counter register
PMU_SCU_PWR_CON	0x0080	W	0x00000000	Wakeup interrupt control register
PMU_SCU_STS	0x0080	W	0x00000000	SCU status register
PMU_BIU_IDLE_CON	0x00B0	W	0x00000000	BIU idle request hardware control register
PMU_BIU_IDLE_SFTCON	0x00C0	W	0x00000000	BIU idle request software control register
PMU_BIU_IDLE_ACK	0x00D0	W	0x00000000	BIU idle acknowledge status register
PMU_BIU_IDLE_ST	0x00D8	W	0x00000000	BIU idle status register
PMU_BIU_AUTO_CON	0x00E0	W	0x00000000	BIU automatic power control register
PMU_DDR_PWR_CON	0x00F0	W	0x00000000	DDR power hardware control register
PMU_DDR_PWR_SFTCON	0x00F4	W	0x00000000	DDR power software control register
PMU_DDR_POWER_STS	0x00F8	W	0x00000000	DDR power state register
PMU_DDR_STS	0x00FC	W	0x00000000	DDR status register
PMU_CRU_PWR_CON0	0x0120	W	0x00000000	Clock and reset hardware power control register 0
PMU_CRU_PWR_CON1	0x0140	W	0x00000000	Clock and reset hardware power control register 1
PMU_CRU_PWR_SFTCON	0x0124	W	0x00000000	Clock and reset software power control register
PMU_CRU_POWER_STS	0x0128	W	0x00000000	Clock and reset power state register
PMU_PLLPD_CON	0x0130	W	0x00000000	PLL power hardware control register
PMU_PLLPD_SFTCON	0x0134	W	0x00000000	PLL power software control register
PMU_INFO_TX_CON	0x0150	W	0x00000000	PMU information transmit control register
PMU_SYS_REG0	0x01C0	W	0x00000000	PMU system register 0
PMU_SYS_REG1	0x01C4	W	0x00000000	PMU system register 1
PMU_SYS_REG2	0x01C8	W	0x00000000	PMU system register 2
PMU_SYS_REG3	0x01CC	W	0x00000000	PMU system register 3
PMU_SYS_REG4	0x01D0	W	0x00000000	PMU system register 4
PMU_SYS_REG5	0x01D4	W	0x00000000	PMU system register 5
PMU_SYS_REG6	0x01D8	W	0x00000000	PMU system register 6
PMU_SYS_REG7	0x01DC	W	0x00000000	PMU system register 7

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

7.4.2 Detail Register Description

PMU_VERSION

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x030a0000	version PMU version number

PMU_PWR_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	cru_bypass Bypass clock and reset low power flow in low power procedure. If asserted, PMU_CRU_PWR_CON cannot take effect for clock and reset low power flow. If you want to execute clock and reset low power flow, you can program PMU_CRU_PWR_SFTCON through software flow. 1'b0: Disable 1'b1: Enable
6	RO	0x0	reserved
5	RW	0x0	ddr_bypass Bypass DDR low power flow in low power procedure. If asserted, PMU_DDR_PWR_CON cannot take effect for DDR low power flow. If you want to execute DDR low power flow, you can program PMU_DDR_SFTCON through software flow. 1'b0: Disable 1'b1: Enable
4	RW	0x0	bus_bypass Bypass BIU idle request in low power procedure. If asserted, PMU_BIU_IDLE_CON cannot take effect for BIU idle request. If you want to execute BIU idle request, you can program PMU_BIU_IDLE_SFTCON through software flow. 1'b0: Disable 1'b1: Enable
3:1	RO	0x0	reserved
0	R/WSC	0x0	powermode_en Low power mode enable. When controller enters low power flow, this bit is automatically cleared. 1'b0: Disable 1'b1: Enable

PMU_GLB_POWER_STS

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
3:0	RO	0x0	power_state PMU global power state. 4'h0: Normal state 4'h1: SCU low power state 4'h3: Bus low power state 4'h4: DDR low power state 4'h5: Power gating low power state 4'h6: Clock and reset low power state 4'h7: Sleep state 4'h8: Clock and reset active state 4'h9: Power gating active state 4'ha: DDR active state 4'hb: Bus active state 4'hd: SCU active state Others: Reserved

PMU_INT_MASK_CON

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	glb_int_mask Global interrupt mask during low power procedure. 1'b0: Disable 1'b1: Enable

PMU_WAKEUP_INT_CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	pmu_sft_wakeup_cfg Software wakeup enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	wakeup_timerout_en Enable PMU timeout interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
5	RW	0x0	wakeup_timer_en Enable HPTIMER interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
4	RW	0x0	wakeup_usbdev_en Enable USB device detect interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
3	RW	0x0	wakeup_sdio_en Enable SDIO interrupt as wakeup source. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
2	RW	0x0	wakeup_sdmmc_en Enable SDMMC detect interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
1	RW	0x0	wakeup_gpio_int_en Enable GPIO0 interrupt as wakeup source. 1'b0: Disable 1'b1: Enable
0	RW	0x0	wakeup_cpu_int_en Enable CPU interrupt as wakeup source. 1'b0: Disable 1'b1: Enable

PMU_WAKEUP_INT_ST

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RO	0x0	wakeup_timerout_int_st PMU timeout interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
5	RO	0x0	wakeup_timer_int_st HPTIMER interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
4	RO	0x0	wakeup_usbdev_int_st USB device detect interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
3	RO	0x0	wakeup_sdio_int_st SDIO interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
2	RO	0x0	wakeup_sdmmc_int_st SDMMC detect interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
1	RO	0x0	wakeup_gpio_int_st GPIO0 interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active
0	RO	0x0	wakeup_cpu_int_st CPU interrupt as wakeup source status. 1'b0: Inactive 1'b1: Active

PMU_PMIC_STABLE_CNT

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0xfffff	stable_cnt PMIC power stable counter for CRU from power off to wakeup

PMU_OSC_STABLE_CNT

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0xfffff	stable_cnt OSC stable counter for OSC from power off to wakeup

PMU_WAKEUP_RSTCLR_CNT

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0xfffff	wakeup_rstclr_cnt Stable counter for CRU wakeup reset clear

PMU_PLL_LOCK_CNT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0xfffff	pll_lock_cnt Lock counter for PLL from powerup to lock

PMU_WAKEUP_TIMEOUT_CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00005dc0	wakeup_timeout_cnt WAKEUP timeout counter

PMU_PWM_SWITCH_CNT

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0xfffff	stable_cnt PWM switch stable counter

PMU_SCU_PWR_CON

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	standbywfi_byass Bypass core standbywfi. 1'b0: Disable. PMU should not execute low power procedure until core enters wfi state. 1'b1: Enable, PMU will execute low power procedure not caring core wfi state.
1	RW	0x0	cpu_int_mask_ena Interrupt mask for CPU during low power procedure. 1'b0: Disable 1'b0: Enable
0	RW	0x0	scu_int_mask_ena Interrupt mask for SCU during low power procedure. 1'b0: Disable 1'b0: Enable

PMU_SCU_STS

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	standbywfil2 L2 standbywfi state. 1'b0: Inactive 1'b1: Active
0	RO	0x0	standbywfi CPU standbywfi state. 1'b0: Inactive 1'b1: Active

PMU_BIU_IDLE_CON

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	idle_req_vepu Enable sending bus idle request to BIU_VEPU by hardware. 1'b0: Disable 1'b1: Enable
9	RW	0x0	idle_req_venc_com Enable sending bus idle request to BIU_VENC_COM by hardware. 1'b0: Disable 1'b1: Enable
8	RW	0x0	idle_req_cpu Enable sending bus idle request to BIU_CPU by hardware. 1'b0: Disable 1'b1: Enable
7	RW	0x0	idle_req_cru Enable sending bus idle request to BIU_CRU by hardware. 1'b0: Disable 1'b1: Enable
6	RW	0x0	idle_req_peri Enable sending bus idle request to BIU_PERI by hardware. 1'b0: Disable 1'b1: Enable
5	RW	0x0	idle_req_vo Enable sending bus idle request to BIU_VO by hardware. 1'b0: Disable 1'b1: Enable
4	RW	0x0	idle_req_vi Enable sending bus idle request to BIU_VI by hardware. 1'b0: Disable 1'b1: Enable
3	RW	0x0	idle_req_npu_aclk Enable sending bus idle request to BIU_NPU_ACLK by hardware. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_npu Enable sending idle request to BIU_NPU by hardware. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	idle_req_ddr Enable sending bus idle request to BIU_DDR by hardware. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_msch Enable sending bus idle request to BIU_MSCH by hardware. 1'b0: Disable 1'b1: Enable

PMU_BIU_IDLE_SFTCON

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	idle_req_vepu Enable sending bus idle request to BIU_VEPU by software. 1'b0: Disable 1'b1: Enable
9	RW	0x0	idle_req_venc_com Enable sending bus idle request to BIU_VENC_COM by software. 1'b0: Disable 1'b1: Enable
8	RW	0x0	idle_req_cpu Enable sending bus idle request to BIU_CPU by software. 1'b0: Disable 1'b1: Enable
7	RW	0x0	idle_req_cru Enable sending bus idle request to BIU_CRU by software. 1'b0: Disable 1'b1: Enable
6	RW	0x0	idle_req_peri Enable sending bus idle request to BIU_PERI by software. 1'b0: Disable 1'b1: Enable
5	RW	0x0	idle_req_vo Enable sending bus idle request to BIU_VO by software. 1'b0: Disable 1'b1: Enable
4	RW	0x0	idle_req_vi Enable sending bus idle request to BIU_VI by software. 1'b0: Disable 1'b1: Enable
3	RW	0x0	idle_req_npu_aclk Enable sending bus idle request to BIU_NPU_ACLK by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	idle_req_npu Enable sending idle request to BIU_NPU by software. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	idle_req_ddr Enable sending bus idle request to BIU_DDR by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	idle_req_msch Enable sending bus idle request to BIU_MSCH by software. 1'b0: Disable 1'b1: Enable

PMU_BIU_IDLE_ACK

Address: Operational Base + offset (0x00D0)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RO	0x0	idle_ack_vepu BIU_VEPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
9	RO	0x0	idle_ack_venc_com BIU_VENC_COM bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
8	RO	0x0	idle_ack_cpu BIU_CPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
7	RO	0x0	idle_ack_cru BIU_CRU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
6	RO	0x0	idle_ack_peri BIU_PERI bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
5	RO	0x0	idle_ack_vo BIU_VO bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
4	RO	0x0	idle_ack_vi BIU_VI bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
3	RO	0x0	idle_ack_npu_aclk BIU_NPU_ACLK bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
2	RO	0x0	idle_ack_npu BIU_NPU bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge
1	RO	0x0	idle_ack_ddr BIU_DDR bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge

Bit	Attr	Reset Value	Description
0	RO	0x0	idle_ack_msch BIU_MSCH bus idle acknowledge state. 1'b0: Not acknowledge 1'b1: Acknowledge

PMU_BIU_IDLE_ST

Address: Operational Base + offset (0x00D8)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved
10	RO	0x0	idle_vepu BIU_VEPU idle state. 1'b0: Not idle 1'b1: Idle
9	RO	0x0	idle_venc_com BIU_VENC_COM idle state. 1'b0: Not idle 1'b1: Idle
8	RO	0x0	idle_cpu BIU_CPU idle state. 1'b0: Not idle 1'b1: Idle
7	RO	0x0	idle_cru BIU_CRU idle state. 1'b0: Not idle 1'b1: Idle
6	RO	0x0	idle_peri BIU_PERI idle state. 1'b0: Not idle 1'b1: Idle
5	RO	0x0	idle_vo BIU_VO idle state. 1'b0: Not idle 1'b1: Idle
4	RO	0x0	idle_vi BIU_VI idle state. 1'b0: Not idle 1'b1: Idle
3	RO	0x0	idle_npu_aclk BIU_NPU_ACLK idle state. 1'b0: Not idle 1'b1: Idle
2	RO	0x0	idle_npu BIU_NPU idle state. 1'b0: Not idle 1'b1: Idle
1	RO	0x0	idle_ddr NIU_DDR idle state. 1'b0: Not idle 1'b1: Idle
0	RO	0x0	idle_msch BIU_MSCH idle state. 1'b0: Not idle 1'b1: Idle

PMU_BIU_AUTO_CON

Address: Operational Base + offset (0x00E0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x00	reserved
10	RW	0x0	auto_idle_vepu If enable, BIU_VEPU corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
9	RW	0x0	auto_idle_venc_com If enable, BIU_VENC_COM corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
8	RW	0x0	auto_idle_cpu If enable, BIU_CPU corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
7	RW	0x0	auto_idle_cru If enable, BIU_CRU corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
6	RW	0x0	auto_idle_peri If enable, BIU_PERI corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
5	RW	0x0	auto_idle_vo If enable, BIU_VO corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
4	RW	0x0	auto_idle_vi If enable, BIU_VI corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
3	RW	0x0	auto_idle_npu_aclk If enable, BIU_NPU_ACLK corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
2	RW	0x0	auto_idle_npu If enable, BIU_NPU corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable
1	RW	0x0	auto_idle_ddr If enable, BIU_DDR corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	auto_idle_msch If enable, BIU_MSCH corresponding clock can be opened or gated automatically when idle operation. 1'b0: Disable 1'b1: Enable

PMU_DDR_PWR_CON

Address: Operational Base + offset (0x00F0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	ddrctl_a_auto_gating_ena Enable DDRCTL's AXI-clock auto clock gating. AXI-clock can be gated when in self-refresh mode. 1'b0: Disable 1'b1: Enable
6	RW	0x0	ddr_sref_a_ena Enable DDR self-refresh mode for AXI-clock domain by hardware. 1'b0: Disable 1'b1: Enable
5	RW	0x0	msch_auto_gating_ena Enable MSCH auto clock gating function performed by PMU, when DDR enter self-refresh state. 1'b0: Disable 1'b1: Enable
4	RO	0x0	reserved
3	RW	0x0	ddrctl_c_auto_gating_ena Enable DDRCTL's core-clock auto clock gating. Core-clock can be gated when in self-refresh mode. 1'b0: Disable 1'b1: Enable
2	RW	0x0	ddrio_ret_exit_ena Enable DDR IO retention de-asserted performed by hardware. 1'b0: Disable 1'b1: Enable
1	RW	0x0	ddrio_ret_ena Enable DDR IO retention function performed by hardware. 1'b0: Disable 1'b1: Enable
0	RW	0x0	ddr_sref_c_ena Enable DDR self-refresh mode for core-clock domain by hardware. 1'b0: Disable 1'b1: Enable

PMU_DDR_PWR_SFTCON

Address: Operational Base + offset (0x00F4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:5	RO	0x000	reserved
4	RW	0x0	sw_ddr_sref_a_req Enable DDR self-refresh mode for AXI-clock domain by software. 1'b0: Disable 1'b1: Enable
3	RW	0x0	ddrctl_active_wait DDR controller waits for c_active high after c_sysack high. 1'b0: Disable 1'b1: Enable
2	RW	0x0	sw_ddrio_ret_exit DDR IO retention exit request by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	sw_ddrio_ret_req DDR IO retention enter request by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	sw_ddr_sref_c_req Enable DDR self-refresh mode for core-clock domain by software. 1'b0: Disable 1'b1: Enable

PMU_DDR_POWER_STS

Address: Operational Base + offset (0x00F8)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2:0	RO	0x0	ddr_power_state DDR power state. 4'h0: Normal state 4'h1: Enter self-refresh mode for AXI-clock state 4'h2: Enter self-refresh mode core-clock state 4'h3: Enter retention mode state 4'h4: Sleep state 4'h5: Exit retention mode 4'h6: Exit self-refresh mode for core-clock state 4'h7: Exit self-refresh mode for AXI-clock state

PMU_DDR_STS

Address: Operational Base + offset (0x00FC)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x0	ddrctl_a_active DDRCTL AXI clock active. 1'b0: Inactive 1'b1: Active
3	RO	0x0	ddrctl_a_sysack DDRCTL AXI hardware low-power request acknowledge. 1'b0: Inactive 1'b1: Active
2	RO	0x0	ddrio_ret DDR IO retention state. 1'b0: Inactive 1'b1: Active

Bit	Attr	Reset Value	Description
1	RO	0x0	ddrctl_c_active DDRCTL hardware low-power clock active. 1'b0: Inactive 1'b1: Active
0	RO	0x0	ddrctl_c_sysack DDRCTL hardware low-power request acknowledge. 1'b0: Inactive 1'b1: Active

PMU_CRU_PWR_CON0

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x00	reserved
8	RW	0x0	pwm_switch_iout PWM output. 1'b0: Disable 1'b1: Enable
7	RW	0x0	pwm_gpio_ioe_ena PWM output enable. 1'b0: Disable 1'b1: Enable
6	RW	0x0	pwm_switch_ena PWM switch. 1'b0: Disable 1'b1: Enable
5	RW	0x0	power_off_ena Chip power off enable by hardware. 1'b0: Disable 1'b1: Enable
4	RW	0x0	alive_osc_ena pclk_pmu switch oscillator enable. When alive_32k_ena is asserted, this bit is ignored. 1'b0: Disable 1'b1: Enable
3	RW	0x0	input_clamp_ena VD_PMU input clamp enable by hardware. 1'b0: Disable 1'b1: Enable
2	RW	0x0	wakeup_rst_ena Wakeup reset enable. If asserted, the whole chip except IPs supporting reset hold function will be reset. 1'b0: Disable 1'b1: Enable
1	RW	0x0	osc_dis_ena Disable oscillator by hardware. 1'b0: Enable 1'b1: Disable

Bit	Attr	Reset Value	Description
0	RW	0x0	alive_32k_ena Enable pclk_pmu and clk_pmu switch to 32KHz clock by hardware. 1'b0: Disable 1'b1: Enable

PMU_CRU_PWR_CON1

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x00	reserved
7	RW	0x0	cru_clk_src_gate_ena Auto gating CRU_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
6	RW	0x0	core_clk_src_gate_ena Auto gating CORE_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
5	RW	0x0	peri_clk_src_gate_ena Auto gating PERI_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
4	RW	0x0	ddr_clk_src_gate_ena Auto gating DDR_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
3	RW	0x0	npu_clk_src_gate_ena Auto gating NPU_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
2	RW	0x0	venc_clk_src_gate_ena Auto gating VENC_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
1	RW	0x0	vo_clk_src_gate_ena Auto gating VO_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable
0	RW	0x0	vi_clk_src_gate_ena Auto gating VI_WRAPPER clock when CRU_POWER_STS is in sleep state. 1'b0: Disable 1'b1: Enable

PMU_CRU_PWR_SFTCON

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x000	reserved
5	RW	0x0	power_off_ena Power off chip by software. 1'b0: Disable 1'b1: Enable
4	RW	0x0	alive_osc_ena pclk_pmu switch oscillator enable by software. When alive_32k_ena is asserted, this bit is ignored. 1'b0: Disable 1'b1: Enable
3	RW	0x0	input_clamp_ena VD_PMU input clamp enable by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	wakeup_rst_ena Wakeup reset enable by software. Reset the whole chip, except IPs supporting reset hold function. 1'b0: Disable 1'b1: Enable
1	RW	0x0	osc_dis_ena Disable oscillator by software. 1'b0: Enable 1'b1: Disable
0	RW	0x0	alive_32k_ena Enable pclk_pmu and clk_pmu switch to 32KHz clock by software. 1'b0: Disable 1'b1: Enable

PMU_CRU_POWER_STS

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RO	0x0	cru_power_state Clock and reset power state. 4'h0: Normal state 4'h1: Clock low frequency state 4'h2: PLL power down state 4'h3: Input clamp state 4'h4: Oscillator disable state 4'h5: Sleep state 4'h6: Wakeup state 4'h7: Oscillator enable state 4'h8: Input clamp release state 4'h9: Clock high frequency state 4'ha: Wakeup reset clear state 4'hb: GPIO switch state 4'hc: PLL power up state 4'hd: PWM switch state Others: Reserved

PMU_PLLPD_CON

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpll_pd_ena GPLL power down by PMU. 1'b0: Disable 1'b1: Enable
2	RW	0x0	cpll_pd_ena CPLL power down by PMU. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dpll_pd_ena DPLL power down by PMU. 1'b0: Disable 1'b1: Enable
0	RW	0x0	apll_pd_ena APLL power down by PMU. 1'b0: Disable 1'b1: Enable

PMU_PLLPD_SFTCON

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x000	reserved
3	RW	0x0	gpll_pd_ena GPLL power down by software. 1'b0: Disable 1'b1: Enable
2	RW	0x0	cpll_pd_ena CPLL power down by software. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dpll_pd_ena DPLL power down by software. 1'b0: Disable 1'b1: Enable
0	RW	0x0	apll_pd_ena APLL power down by software. 1'b0: Disable 1'b1: Enable

PMU_INFO_TX_CON

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:8	RW	0x00	info_tx_intv_time The interval time from 1 byte sends over to a new byte sends start. The value is the cycle number counted in clk_pmu.
7:1	RO	0x00	reserved
0	RW	0x0	info_tx_en PMU debug information transmit enable. 1'b0: Disable 1'b1: Enable

PMU_SYS_REG0

Address: Operational Base + offset (0x01C0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG1

Address: Operational Base + offset (0x01C4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG2

Address: Operational Base + offset (0x01C8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG3

Address: Operational Base + offset (0x01CC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG4

Address: Operational Base + offset (0x01D0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG5

Address: Operational Base + offset (0x01D4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG6

Address: Operational Base + offset (0x01D8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

PMU_SYS_REG7

Address: Operational Base + offset (0x01DC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg PMU system register

7.5 Interface Description

Table 7-2 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
pmu_debug	O	PWM4_M0/PMU_DEBUG/I2C2_SDA_M0/UART3_RX_M0/GPIO1_A1_d	VENCIOC_GPIO1A_IOMUX_SEL_L[7:4]=3
pmic_sleep_m0	O	PWM1_M0/PMIC_SLEEP_M0/GPIO0_A4_d	PMUIOC_GPIO0A_IOMUX_SEL_H[3:0]=1
pmic_sleep_m1	O	PMIC_SLEEP_M1/GPIO0_A3_u	PMUIOC_GPIO0A_IOMUX_SEL_L15:12]=1

Notes: I=input, O=output, I/O=input/output, bidirectional

7.6 Application Notes

7.6.1 Hardware Low Power Mode

PMU can work in low power mode by hardware. After setting bit[0] of PMU_PWR_CON and CPU enter standbywfi state, PMU low power FSM will start to run. In the low power mode, PMU will manage power resources by hardware or software, such as send idle request to specified power domain, auto clock switch and clock gating, shut down PLLs, and so on. All of above are configurable by setting corresponding registers.
Also, the standbywfi state can be bypass if PMU_SCU_PWR_CON[2] is set to 1.

7.6.2 Software Low Power Mode

PMU can work in the low power mode by software. In this application, you can use LPMCU to set PMU software power control registers to execute low power procedure.

7.6.3 PMU Debug Information

The PMU internal states could be monitored through pmu_debug IO in 8-bit UART interface mode. The transmission can be enabled by setting PMU_INFO_TX_CON[0], and the transmission interval time can be set in PMU_INFO_TX_CON[15:8].

Table 7-3 PMU Debug Information Decode

Decode Value	PMU State	Description
1	Not care	-
2	Not care	-
3	Bus low power state	PMU_GLB_POWER_STS=4'h3
4	DDR enters self-refresh mode for AXI-clock state	PMU_DDR_POWER_STS=4'h1
5	DDR enters self-refresh mode for core-clock state	PMU_DDR_POWER_STS=4'h2
6	DDR enters retention mode state	PMU_DDR_POWER_STS=4'h3
7	Not care	-
8	Clock low frequency state	PMU_CRU_POWER_STS=4'h1
9	PLL power down state	PMU_CRU_POWER_STS=4'h2
10	Input clamp state	PMU_CRU_POWER_STS=4'h3
11	Oscillator disable state	PMU_CRU_POWER_STS=4'h4
12	Sleep state	PMU_GLB_POWER_STS=4'h7
13	Wake up state	PMU_CRU_POWER_STS=4'h6
14	Oscillator enable state	PMU_CRU_POWER_STS=4'h7
15	Input clamp release state	PMU_CRU_POWER_STS=4'h8
16	Clock high frequency state	PMU_CRU_POWER_STS=4'h9
17	Wake up reset clear state	PMU_CRU_POWER_STS=4'ha

Decode Value	PMU State	Description
18	GPIO switch state	PMU_CRU_POWER_STS=4'hb
19	PLL power up state	PMU_CRU_POWER_STS=4'hc
20	PWM switch state	PMU_CRU_POWER_STS=4'hd
21	Not care	-
22	DDR exits retention mode through RETON/RETOFF state	PMU_DDR_POWER_STS=4'h5
23	DDR exits self-refresh mode for core-clock state	PMU_DDR_POWER_STS=4'h6
24	DDR exits self-refresh mode for AXI-clock state	PMU_DDR_POWER_STS=4'h7
25	Bus active state	PMU_GLB_POWER_STS=4'hb
26	Not care	-
27	Not care	-

7.6.4 System Register

PMU support 8 system registers: PMU_SYS_REG0~ PMU_SYS_REG7. These registers are always on no matter what low power mode. Therefore, software can use these registers to retain some information which is useful after wakeup from any mode.

Chapter 8 MCU Subsystem

8.1 Overview

The MCU Subsystem embedded one MCU processor(RISCV Architecture), a 16KB unified I/D cache and interrupt multiplexer(INTMUX). It allowed fast & efficient data exchange. And the Level 1 Instruction/Data Cache designed for MCU can improve the memory access performance significantly.
The following features may or may not be present in the actual product. Please contact Rockchip for the actual feature configurations and the third-party licensing requirements.
The MCU features and benefits are:

- One MCU processor
- Machine privilege level
- 32 32-bit general purpose integer registers
- Instruction set is RV32I with M and C extensions
- 3 stage pipeline implementation
- 3 embedded 64bit performance counters
  - Real time clock
  - Cycle counter
  - Instructions-retired counter
- Optional debug support
  - Two hardware breakpoints
  - Support Serial Wire debug connection
- Interrupt multiplexer(INTMUX)
  - Multiplexer 16 interrupts from 128 interrupt sources using round-robin algorithm
  - Each 32 adjacent interrupts are seen as a group which share same INTMUX_INT_ENABLE_GROUPx and INTMUX_INT_FLAG_GROUPx register
- 16KB unified I/D Cache
  - Support 2-way set-associative with 32Byte cache line size
  - Support one 128bit AHB-Lite bus master interface to SoC interconnect
  - Support memory cacheable access and peripheral bypass access
  - Support Write-Through with No-Write-Allocate and Read-Allocate
  - Support Write-Back with Write-Allocate and Read-Allocate
  - Support cache maintenance operations (clean/invalidate/clean & invalidate) by address
  - Support cache invalidate/clean & invalidate operations for all cache lines
  - Support cache initialization by software configuration
  - Support RAM debug mode for tag RAM and data RAM
  - Support one interrupt source

There is a MCU subsystems in RV1106, High performance(HP) subsystems. It's main configurations are as shown below.

Table 8-1 MCU Subsystem Configurations

	HP_MCU
CACHE	16KB
IRQ_NUM	16
INTMUX	YES

8.2 Block Diagram

The Int_in[127:0] will connect with all interrupt source of RV1106, and please refer to "Chapter System Overview" for the detail interrupt number. The Int_in[15:0] will be specified in section 1.5.2. MCU Subsystem are shown below.

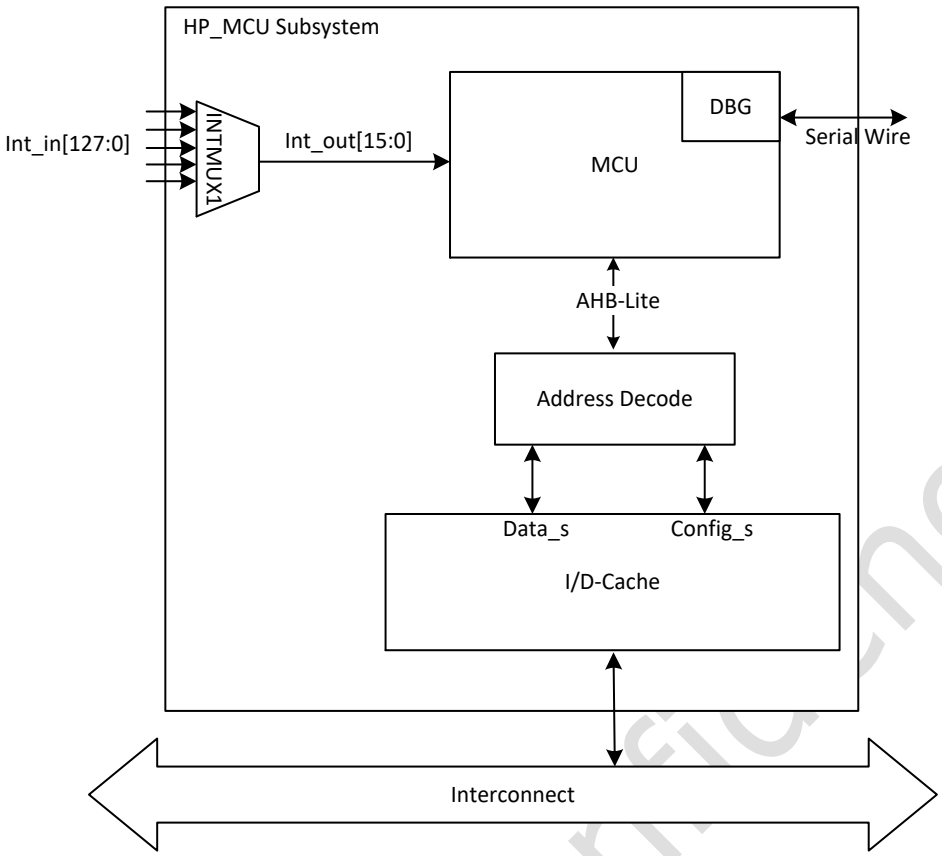


Fig. 8-1 Block Diagram of HP_MCU Subsystem

8.3 Register Description

Software should read and write these registers using 32-bits accesses.

8.3.1 Registers Summary(MCU)

The first six registers are timer related which are outside of MCU core, the base address is 0xFF1E0000, they should be accessed by load/store instructions. The rest registers are control status registers(CSR) which are in MCU core, they should be accessed by system instructions.

Name	Offset	Size	Reset Value	Description
MCU_TIMER_CTRL	0x0000	W	0x00000001	Timer Control Register
MCU_TIMER_DIV	0x0004	W	0x00000000	Timer Divider Register
MCU_MTIME	0x0008	W	0x00000000	Machine Timer Register Low 32bit
MCU_MTIMEH	0x000C	W	0x00000000	Machine Timer Register High 32bit
MCU_MTIMECMP	0x0010	W	0x00000000	Machine Timer Compare Register Low 32bit
MCU_MTIMECMPH	0x0014	W	0x00000000	Machine Timer Compare Register High 32bit
MCU_CSR_MVENDORID	0x0F11	W	0x00000000	Machine Vendor ID Register
MCU_CSR_MARCHID	0x0F12	W	0x00000008	Machine Architecture ID Register
MCU_CSR_MIMPID	0x0F13	W	0x19083000	Machine Implementation ID Register
MCU_CSR_MHARTID	0x0F14	W	0x00000000	Machine Hart ID Register
MCU_CSR_MSTATUS	0x0300	W	0x00001880	Machine Status Register
MCU_CSR_MISA	0x0301	W	0x40001104	Machine Base ISA Control Register

Name	Offset	Size	Reset Value	Description
MCU_CSR_MIE	0x0304	W	0x00000000	Machine Interrupt Enable Register
MCU_CSR_MTVEC	0x0305	W	0x002001C0	Machine Trap-Vector Base-Address Register
MCU_CSR_MSCRATCH	0x0340	W	0x00000000	Machine Scratch Register
MCU_CSR_MEPC	0x0341	W	0x00000000	Machine Exception Program Counter Register
MCU_CSR_MCAUSE	0x0342	W	0x00000000	Machine Cause Register
MCU_CSR_MTVAL	0x0343	W	0x00000000	Machine Trap Value Register
MCU_CSR_MIP	0x0344	W	0x00000000	Machine Interrupt Pending Register
MCU_CSR_MCYCLEL	0x0B00	W	0x00000000	Machine Cycle Counter Low Register
MCU_CSR_MCYCLEH	0x0B80	W	0x00000000	Machine Cycle Counter High Register
MCU_CSR_MINSTRETL	0x0B02	W	0x00000000	Machine Instructions Retired Low Register
MCU_CSR_MINSTRETH	0x0B82	W	0x00000000	Machine Instructions Retired High Register
MCU_CSR_MCOUNTEN	0x07E0	W	0x00000005	Machine Counter Enable Register
MCU_CSR_DBG_SCRATCH	0x07C8	W	0x00000000	Debug Scratch Register
MCU_CSR_IPIC_CISV	0x0BF0	W	0x00000000	Current Interrupt Vector in Service
MCU_CSR_IPIC_CICSR	0x0BF1	W	0x00000000	Current Interrupt Control Status Register
MCU_CSR_IPIC_IPR	0x0BF2	W	0x00000000	Interrupt Pending Register
MCU_CSR_IPIC_ISVR	0x0BF3	W	0x00000000	Interrupt Serviced Register
MCU_CSR_IPIC_EOI	0x0BF4	W	0x00000000	End Of Interrupt
MCU_CSR_IPIC_SOI	0x0BF5	W	0x00000000	Start Of Interrupt
MCU_CSR_IPIC_IDX	0x0BF6	W	0x00000000	Index Register
MCU_CSR_IPIC_ICSR	0x0BF7	W	0x00000000	Interrupt Control Status register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

8.3.2 Detail Registers Description

MCU_TIMER_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	CLKSRC timer clock source 1'b0: internal core clock(default) 1'b1: external real-time clock
0	RW	0x1	ENABLE timer enable bit 1'b0: timer disable 1'b1: timer enable

MCU_TIMER_DIV

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:0	RW	0x000	DIV Timer divider Timer tick occurs every DIV+1 clock ticks

MCU_MTIME

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MTIMEL Low 32bit of wall-clock real time(number of timer ticks)

MCU_MTIMEH

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MTIMEH High 32bit of wall-clock real time(number of timer ticks)

MCU_MTIMECMP

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MTIMECMPL Machine-mode timer compare register low 32bit

MCU_MTIMECMPH

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MTIMECMPH Machine-mode timer compare register high 32bit

MCU_CSR_MVENDORID

Address: Operational Base + offset (0x0F11)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	VENDOR ID The JEDEC manufacturer ID of the provider of the core

MCU_CSR_MARCHID

Address: Operational Base + offset (0x0F12)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000008	ARCH ID Encoding the base micro-architecture of the hart

MCU_CSR_MIMPID

Address: Operational Base + offset (0x0F13)

Bit	Attr	Reset Value	Description
31:0	RO	0x19083000	YEAR-MONTH-DAY-RELEASE Bit[31]-bit[24]: BCD-coded value of the year Bit[23]-bit[16]: BCD-coded value of the month Bit[16]-bit[8]: BCD-coded value of the day Bit[7]-bit[0]: 8-bit value of intraday release number

MCU_CSR_MHARTID

Address: Operational Base + offset (0x0F14)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HART ID Containing the integer ID of the hardware thread running the code.

MCU_CSR_MSTATUS

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
12:11	RO	0x3	MPP Previous privilege mode (hardwired to 11)
10:8	RO	0x0	reserved
7	RW	0x1	MPIE Previous global interrupt enable
6:4	RO	0x0	reserved
3	RW	0x0	MIE Global interrupt enable
2:0	RO	0x0	reserved

MCU_CSR_MISA

Address: Operational Base + offset (0x0301)

Bit	Attr	Reset Value	Description
31:30	RO	0x1	MXL Machine XLEN (hardwired to 01)
29:24	RO	0x00	reserved
23	RO	0x0	RVX Non-standard extensions
22:13	RO	0x000	reserved
12	RO	0x1	RVM Integer Multiply/Divide extension implemented
11:9	RO	0x0	reserved
8	RO	0x1	RVI RV32I base integer instruction set
7:5	RO	0x0	reserved
4	RO	0x0	RVE RV32E base integer instruction set
3	RO	0x0	reserved
2	RO	0x1	RVC Compressed instruction extension implemented
1:0	RO	0x0	reserved

MCU_CSR_MIE

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RW	0x0	MEIE Machine External Interrupt Enable
10:8	RO	0x0	reserved
7	RW	0x0	MTIE Machine Timer Interrupt Enable
6:4	RO	0x0	reserved
3	RW	0x0	MSIE Machine Software Interrupt Enable
2:0	RO	0x0	reserved

MCU_CSR_MTVEC

Address: Operational Base + offset (0x0305)

Bit	Attr	Reset Value	Description
31:6	RW	0x0008007	BASE Vector base address (upper 26 bits)
5:2	RO	0x0	reserved
1:0	RW	0x0	MODE Vector mode (0-direct mode, 1-vectored mode)

MCU_CSR_MSCRATCH

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MSCRATCH Hold a pointer to a machine-mode hart-local context space and swapped with a user register upon entry to an M-mode trap handler.

MCU_CSR_MEPC

Address: Operational Base + offset (0x0341)

Bit	Attr	Reset Value	Description
31:1	RW	0x00000000	MEPC Hold the virtual address of the instruction that encountered the exception.
0	RO	0x0	reserved

MCU_CSR_MCAUSE

Address: Operational Base + offset (0x0342)

Bit	Attr	Reset Value	Description
31	RW	0x0	INT Interrupt
30:4	RO	0x0000000	reserved
3:0	RW	0x0	EC Exception Code

MCU_CSR_MTVAL

Address: Operational Base + offset (0x0343)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:0	RW	0x00000000	MTVAL When a trap is taken into M-mode, mtval is written with exception-specific information to assist software in handling the trap.

MCU_CSR_MIP

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RO	0x0	MEIP Machine External Interrupt Pending
10:8	RO	0x0	reserved
7	RO	0x0	MTIP Machine Timer Interrupt Pending
6:4	RO	0x0	reserved
3	RO	0x0	MSIP Machine Software Interrupt Pending
2:0	RO	0x0	reserved

MCU_CSR_MCYCLEL

Address: Operational Base + offset (0x0B00)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MCYCLEL Represent the number of clock cycles(low 32bit) since some arbitrary point of time in the past.

MCU_CSR_MCYCLEH

Address: Operational Base + offset (0x0B80)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MCYCLEH Represent the number of clock cycles(high 32bit) since some arbitrary point of time in the past.

MCU_CSR_MINSTRETL

Address: Operational Base + offset (0x0B02)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MINSTRETL Represent the number of instructions(low 32bit) retired by the core from some arbitrary time in the past.

MCU_CSR_MINSTRETH

Address: Operational Base + offset (0x0B82)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MINSTRETH Represent the number of instructions(low 32bit) retired by the core from some arbitrary time in the past.

MCU_CSR_MCOUNTEN

Address: Operational Base + offset (0x07E0)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x1	IR Enable retired instructions counter
1	RO	0x0	reserved
0	RW	0x1	CY Enable cycle counter

MCU_CSR_DBG_SCRATCH

Address: Operational Base + offset (0x07C8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DSCRATCH Used to exchange data between the core and the debug controller.

MCU_CSR_IPIC_CISV

Address: Operational Base + offset (0x0BF0)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RO	0x00	CISV Number of the interrupt vector currently in service

MCU_CSR_IPIC_CICSR

Address: Operational Base + offset (0x0BF1)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	IE Interrupt Enable Bit 1'b0: Interrupt disabled 1'b1: Interrupt enabled

Bit	Attr	Reset Value	Description
0	W1C	0x0	IP Interrupt pending 1'b0: no interrupt 1'b1: Interrupt pending

MCU_CSR_IPIC_IPR

Address: Operational Base + offset (0x0BF2)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	W1C	0x0000	PENDING Interrupt vector0 ~ vector15 pending status, each bit represents one vector.

MCU_CSR_IPIC_ISVR

Address: Operational Base + offset (0x0BF3)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RO	0x0000	SERVICE When corresponding bit is set (1) - this interrupt vector is in service.

MCU_CSR_IPIC_EOI

Address: Operational Base + offset (0x0BF4)

Bit	Attr	Reset Value	Description
31:0	WS	0x00000000	END OF INTERRUPT Writing any value to EOI register ends the interrupt which is currently in service, read will return zero.

MCU_CSR_IPIC_SOI

Address: Operational Base + offset (0x0BF5)

Bit	Attr	Reset Value	Description
31:0	WS	0x00000000	START OF INTERRUPT Writing any value to SOI activates start of interrupt if one of the following conditions is true: 1. There is at least one pending interrupt with IE and ISR is zero (no interrupts in service) 2. There is at least one pending interrupt with IE and this interrupt has higher priority than the interrupts currently in service.

MCU_CSR_IPIC_IDX

Address: Operational Base + offset (0x0BF6)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	INDEX Interrupt vector index to access through IPIC_ICSR

MCU_CSR_IPIC_ICSR

Address: Operational Base + offset (0x0BF7)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12	RO	0x0	LN External IRQ Line Number assigned to this interrupt vector. This value is always equal to IPIC_IDX, because of the static line to vector mapping.

Bit	Attr	Reset Value	Description
11:9	RO	0x0	reserved
8	RO	0x0	PRV Privilege mode: hardwired to 11 (machine mode)
7:5	RO	0x0	reserved
4	RW	0x0	IS In Service
3	RW	0x0	INV Line Inversion 1'b0: no inversion 1'b1: line inversion
2	RW	0x0	IM Interrupt Mode 1'b0: Level interrupt 1'b1: Edge interrupt
1	RW	0x0	IE Interrupt Enable Bit 1'b0: Interrupt disabled 1'b1: Interrupt enabled
0	W1C	0x0	IP Interrupt pending 1'b0: no interrupt 1'b1: Interrupt pending

8.3.3 Registers Summary (CACHE)

Name	Offset	Size	Reset Value	Description
CACHE_CTRL	0x0000	W	0x000006cc	Cache Control Register
CACHE_MAINTAIN0	0x0004	W	0x00000000	Cache Maintain 0 Register
CACHE_MAINTAIN1	0x0008	W	0x00000000	Cache Maintain 1 Register
CACHE_STB_TIMEOUT_CTRL	0x000c	W	0x4000000c	Store Buffer Timeout Control Register
CACHE_RAM_DEBUG	0x0010	W	0x00000000	Cache RAM Debug Register
CACHE_INT_EN	0x0020	W	0x00000000	Cache Interrupt Enable Register
CACHE_INT_ST	0x0024	W	0x00000000	Cache Interrupt Status Register
CACHE_ERR_HADDR	0x0028	W	0x00000000	Cache Error Address Register
CACHE_STATUS	0x0030	W	0x00000000	Cache Status Register
CACHE_PMU_RD_NUM_CNT	0x0040	W	0x00000000	PMU Read Number Count Register
CACHE_PMU_WR_NUM_CNT	0x0044	W	0x00000000	PMU Write Number Count Register
CACHE_PMU_RAM_RD_HIT_CNT	0x0048	W	0x00000000	PMU RAM Read Hit Count Register
CACHE_PMU_HB_RD_HIT_CNT	0x004c	W	0x00000000	PMU Hot Buffer Read Hit Count Register
CACHE_PMU_STB_RD_HIT_CNT	0x0050	W	0x00000000	PMU Store Buffer Read Hit Count Register
CACHE_PMU_RD_HIT_CNT	0x0054	W	0x00000000	PMU Read Hit Count Register
CACHE_PMU_WR_HIT_CNT	0x0058	W	0x00000000	PMU Write Hit Count Register
CACHE_PMU_RD_MISS_PENALTY_CNT	0x005c	W	0x00000000	PMU Read Miss Penalty Count Register
CACHE_PMU_WR_MISS_PENALTY_CNT	0x0060	W	0x00000000	PMU Write Miss Penalty Count Register
CACHE_PMU_RD_LAT_CNT	0x0064	W	0x00000000	PMU Read Latency Count Register
CACHE_PMU_WR_LAT_CNT	0x0068	W	0x00000000	PMU Write Latency Count Register
CACHE_REVISION	0x00f0	W	0x00000100	Cache Design Revision Register

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

8.3.4 Detail Registers Description

CACHE_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	cache_pf_en Cache prefetch enable register. 1'b0: Disable 1'b1: Enable
12	RW	0x0	cache_mpu_mode Cache MPU mode enable register. When this bit is set to 1, the cacheability policy is determined by MPU of Cortex-M4. 1'b0: Disable 1'b1: Enable
11	RO	0x0	reserved
10:8	RW	0x6	stb_entry_thresh Store buffer entry threshold control register. The depth of the store buffer entry is 8. When the number of the used data entries is greater than or equal to threshold value, the write data will be written to Cache RAM.
7	RW	0x1	stb_timeout_en Store buffer timeout enable register. When this bit is set to 1, the data in the store buffer must be flushed to Cache RAM if the counter value is equal to the timeout value. 1'b0: Disable 1'b1: Enable
6	RW	0x1	cache_bypass Cache bypass mode enable register. When this bit is set to 1, data will bypass the Cache. 1'b0: Disable 1'b1: Enable
5	RW	0x0	cache_pmu_en Cache Performance Monitor Unit enable register. 1'b0: Disable 1'b1: Enable
4	RW	0x0	cache_flush Cache flush enable register. When this bit is set to 1, the dirty data is flushed to external memory and the cache line are invalidated. 1'b0: Disable 1'b1: Enable
3	RW	0x1	cache_stb_en Cache store buffer enable register. This bit must be set to 0 when Write-Through mode is selected and must be set to 1 when Write-Back mode is selected. 1'b0: Disable 1'b1: Enable
2	RW	0x1	cache_hb_en Cache hot buffer enable register. 1'b0: Disable 1'b1: Enable
1	RW	0x0	cache_wt_en Cache mode control register. 1'b0: Write-Back 1'b1: Write-Through
0	RW	0x0	cache_en Cache initialization enable register. 1'b0: Disable 1'b1: Enable

CACHE_MAINTAIN0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:5	RW	0x0000000	cache_m_addr Cache maintain start address. This address is 32Byte cache line aligned, that is, the bits[4:0] are always 0.
4:3	RO	0x0	reserved
2:1	RW	0x0	cache_m_cmd Cache maintain command register. 2'b00: Clean by address 2'b01: Invalidate by address 2'b10: Clean and Invalidate by address 2'b11: Invalidate all
0	WO	0x0	cache_m_valid Cache maintain valid register. The maintenance operation is valid only when this bit is set to 1.

CACHE_MAINTAIN1

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26:0	RW	0x0000000	cache_m_offset Cache maintain offset. This bit field indicates the end offset of cache line, that is, the value plus 1 determines the number of cache line to be maintained. The bit field of cache_m_addr is treated as the start offset of cache line, so the maintain address range is from (cache_m_addr * 32) to (cache_m_addr * 32 + 31 + cache_m_offset * 32).

CACHE_STB_TIMEOUT_CTRL

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x4000000c	stb_timeout_value Store buffer timeout value.

CACHE_RAM_DEBUG

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	ram_debug_en Cache RAM debug mode enable register. 1'b0: Disable 1'b1: Enable

CACHE_INT_EN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	err_record_en AHB master bus error record enable. 1'b0: Disable 1'b1: Enable

CACHE_INT_ST

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	ahb_error_status Error status bit for AHB master bus. 1'b0: Nothing 1'b1: Bus error

CACHE_ERR_HADDR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	status_haddr Recent record of AHB bus error address.

CACHE_STATUS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RO	0x0	cache_flush_done Cache flush done status register. 1'b0: Nothing 1'b1: Flush done
1	RO	0x0	cache_m_busy Cache maintain busy status register. 1'b0: Idle 1'b1: Busy
0	RO	0x0	cache_init_finish Cache initialization finish status register. 1'b0: Cache is uninitialized 1'b1: Cache is initialized

CACHE_PMU_RD_NUM_CNT

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_num_cnt Total count of read transfers.

CACHE_PMU_WR_NUM_CNT

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_num_cnt Total count of write transfers.

CACHE_PMU_RAM_RD_HIT_CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_ram_rd_hit_cnt Count of read hits on Cache RAM.

CACHE_PMU_HB_RD_HIT_CNT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_hb_rd_hit_cnt Count of read hits on hot buffer.

CACHE_PMU_STB_RD_HIT_CNT

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_stb_rd_hit_cnt Count of read hits on store buffer.

CACHE_PMU_RD_HIT_CNT

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_hit_cnt Total count of read hits.

CACHE_PMU_WR_HIT_CNT

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_hit_cnt Total count of write hits.

CACHE_PMU_RD_MISS_PENALTY_CNT

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_miss_penalty_cnt Total count of read miss penalty (in clock cycles).

CACHE_PMU_WR_MISS_PENALTY_CNT

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_miss_penalty_cnt Total count of write miss penalty (in clock cycles).

CACHE_PMU_RD_LAT_CNT

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_lat_cnt Total count of read latency (in clock cycles).

CACHE_PMU_WR_LAT_CNT

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_lat_cnt Total count of write latency (in clock cycles).

CACHE_REVISION

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000100	revision Cache revision number.

8.3.5 Registers Summary(INTMUX)

Name	Offset	Size	Reset Value	Description
INTMUX_INT_ENABLE_GROUP0	0x0000	W	0x00000000	Interrupt Group0 Enable Register
INTMUX_INT_ENABLE_GROUP1	0x0004	W	0x00000000	Interrupt Group1 Enable Register
INTMUX_INT_ENABLE_GROUP2	0x0008	W	0x00000000	Interrupt Group2 Enable Register

Name	Offset	Size	Reset Value	Description
INTMUX_INT_ENABLE_GROUP3	0x000C	W	0x00000000	Interrupt Group3 Enable Register
INTMUX_INT_FLAG_GROUP0	0x0080	W	0x00000000	Interrupt Group0 Flag Register
INTMUX_INT_FLAG_GROUP1	0x0084	W	0x00000000	Interrupt Group1 Flag Register
INTMUX_INT_FLAG_GROUP2	0x0088	W	0x00000000	Interrupt Group2 Flag Register
INTMUX_INT_FLAG_GROUP3	0x008C	W	0x00000000	Interrupt Group3 Flag Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

8.3.6 Detail Registers Description

INTMUX_INT_ENABLE_GROUP0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	enable Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	enable Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	enable Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_ENABLE_GROUP3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	enable Each bit represents one interrupt enable bit. 1'b0: Interrupt disabled 1'b1: Interrupt enabled

INTMUX_INT_FLAG_GROUP0

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	flag Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP1

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	flag Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP2

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	flag Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

INTMUX_INT_FLAG_GROUP3

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	flag Each bit represents one interrupt is in service. 1'b0: Interrupt in service 1'b1: Interrupt out of service

8.4 Interface Description

HP_MCU has two sets of serial debug interface. When we debug using serial wire, not only set IOMUX, but also configure the bit[1] of CPU_CTRL_CON in CORE_SGRF to enable debug for HP_MCU.

Table 8-2 MCU Serial Wire Debug Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
hpmcujtag_tckm0	I	LPMCU_JTAG_TCK_M0/HPMCU_JTAG_TCK_M0/UART2_TX_M1/A7_JTAG_TCK_M1/GPIO1_B2_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[11:8] = 3'h3
hpmcujtag_tmsm0	I/O	LPMCU_JTAG_TMS_M0/HPMCU_JTAG_TMS_M0/UART2_RX_M1/A7_JTAG_TMS_M1/GPIO1_B3_u	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[14:12] = 3'h3
hpmcujtag_tckm1	I	HPMCU_JTAG_TCK_M1/A7_JTAG_TCK_M0/UART5_RX_M0/SDMMC0_D2/GPIO3_A7_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[14:12] = 3'h4
hpmcujtag_tmsm1	I/O	HPMCU_JTAG_TMS_M1/A7_JTAG_TMS_M0/UART5_TX_M0/SDMMC0_D3/GPIO3_A6_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[11:8] = 3'h4

Notes: I=input, O=output, I/O=input/output, bidirectional

8.5 Application Notes

8.5.1 Clock and Reset Generation

Please refer to "Chapter CRU" for more detailed information.

8.5.2 Cache Initialization

The Cache is bypassed after reset and need to be initialized by the MCU. The basic initialization flow is:

1. Configure to Write-Back or Write-Through mode by setting CACHE_CTRL.cache_wt_en and start initialization by configuring CACHE_CTRL.cache_en to high while keeping the Cache in bypass mode (CACHE_CTRL.cache_bypass is high). The store buffer must and only can be enabled for Write-Back mode, so you must disable the store buffer when Write-Through mode is selected. You can also enable the hot buffer for better performance. It is recommended that configure the I-Cache to Write-Through mode and configure the D-Cache to Write-

Back mode.
2. Wait for cache initialization to be finished by polling CACHE_STATUS.cache_init_finish.
3. Disable cache bypass mode by configuring CACHE_CTRL.cache_bypass to low. Now that the Cache is enabled.
4. Simply configure CACHE_CTRL.cache_bypass to bypass or enable the Cache again after the initialization flow.

8.5.3 Cache Maintenance

The Cache Controller support the following cache maintenance operations:

- Clean by address: push the specified cache line data to external memory if it is valid and dirty.
- Invalidate by address: unconditionally clear the valid and dirty data of the specified cache lines.
- Clean and invalidate by address: clean and then invalidate the specified cache lines.
- Invalidate all: unconditionally clear the valid and dirty data for all the cache lines.
- Clean and invalidate all (Flush): clean and then invalidate all the cache lines.
  For the operation of cache clean and invalidate all, configure CACHE_CTRL.cache_flush to high first, and then wait for flush to be done by polling CACHE_STATUS.cache_flush_done. Finally, if the flush is done, you should configure CACHE_CTRL.cache_flush to low.
  For the other four maintenance operations, you should set the maintenance command and the address range of the cache lines by configuring CACHE_MAINTAIN0 and CACHE_MAINTAIN1, then start by configuring CACHE_MAINTAIN0.cache_m_valid to high. When the operation is finished, CACHE_MAINTAIN0.cache_m_valid will be automatically cleared to low. You can also poll CACHE_STATUS.cache_m_busy to judge whether the operation is done.

8.5.4 Cache RAM Debug Mode

When the Cache is enabled and not in the bypass mode, you can enter the cache RAM debug mode by configuring CACHE_RAM_DEBUG.ram_debug_en to high. In the RAM debug mode, the cache RAM can be accessed directly by the specific address, and all the other cacheable memory accesses are bypassed because the cache core is disabled. The following table shows the address map for the cache RAM debug.

Table 8-3 Address Map for the Cache RAM Debug

RAM Access Type	RAM Access Size	Address Range for Cache RAM Debug
Reserved	7KB	0xF6F26400 ~ 0xF6F27FFF
TAG RAM1	1KB	0xF6F26000 ~ 0xF6F263FF
Reserved	7KB	0xF6F24400 ~ 0xF6F25FFF
TAG RAMCU	1KB	0xF6F24000 ~ 0xF6F243FF
DATA RAM1	8KB	0xF6F22000 ~ 0xF6F23FFF
DATA RAMCU	8KB	0xF6F20000 ~ 0xF6F21FFF

As shown in the above table, there are 2 ways for each cache, and the size of one DATA RAM is 8KB. The following explanation may help you use the RAM debug method:

- Using the word-aligned address to access the TAG RAM.
- The bits[9:2] of the TAG RAM address are acted as the bits[12:5] of the corresponding memory address.
- Only the bits[20:0] of the TAG RAM data are available, and the bits[31:21] are always zero.
  - The bit[20] is the valid flag bit.
  - The bit[19] is the dirty flag bit.
  - The bits[18:0] are acted as the bits[31:13] of the corresponding memory address.
- The bits[12:0] of the DATA RAM address are acted as the bits[12:0] of the corresponding memory address.
- In order to get the cache line data at the wanted address, you should read the both of the TAG RAMs to find the matched address, and then read data from the corresponding

DATA RAM.

- If write data to the DATA RAM, 8 copies of the 32bit data on the AHB bus will be written to the 32Byte cache line.

8.5.5 Other Hints

One interrupt request source is supported for the I/D-Cache, so you can query the interrupt register in the ISR if it is enabled. When CACHE_INT_EN.err_record_en is high, the Cache will generate a bus error interrupt if access to a wrong address. You can read CACHE_INT_ST.ahb_error_status for the cache interrupt status and get the recent error address information from CACHE_ERR_HADDR.status_haddr.

8.6 Application Notes

- How to boot?

Currently, the MCU core boot address is configurable, its default value is 32'h0000_0000, and the default value of trap vector base address is 32'h0010_01c0, you can configure SCR1_BOOT_ADDR of SGRF before de-assert its reset. if you change the boot address, don't forget to configure the MCU_CSR_MTVEC register by system instructions.

Chapter 9 Timer

9.1 Overview

TIMER is a programmable timer peripheral. This component is an APB slave device. In RV1106, there are three types of TIMER according to the count type, which are normal decrement count TIMER, normal increment count TIMER and special function increment count TIMER. The normal decrement count TIMER is called NDCTIMER. The normal increment count TIMER is called NICTIMER. The special function increment count TIMER is called HPTIMER(High Precision TIMER).

5 normal decrement count TIMERs and 1 normal increment count TIMER constitute a 6-channel TIMER. In 6-channel TIMER, TIMER5 is the normal increment count TIMER, and the other channel TIMER is the normal decrement count TIMER.

2 normal increment count TIMERs form a 2-channel TIMER.

HPTIMER has calibration function and provides count value for CPU.

Timer supports the following features:

- All three kinds of counter support 32 bits APB slave
- All three kinds of counter loading count value support configurable
- All three kinds of counter support two counting modes: free-running and user-defined count
- All three kinds of counters support maskable interrupts
- HPTIMER with three timer mode: Normal, Hardware adjust, Software adjust
- Hardware adjust mode HPTIMER can be calibrated back to the exact count value without software calculation
- Software adjust mode HPTIMER can be calibrated back to the exact count value with software-aided calculation
- Common multiples of slow and fast clock cycles are configurable

9.2 Block Diagram

9.2.1 Normal count TIMER block diagram

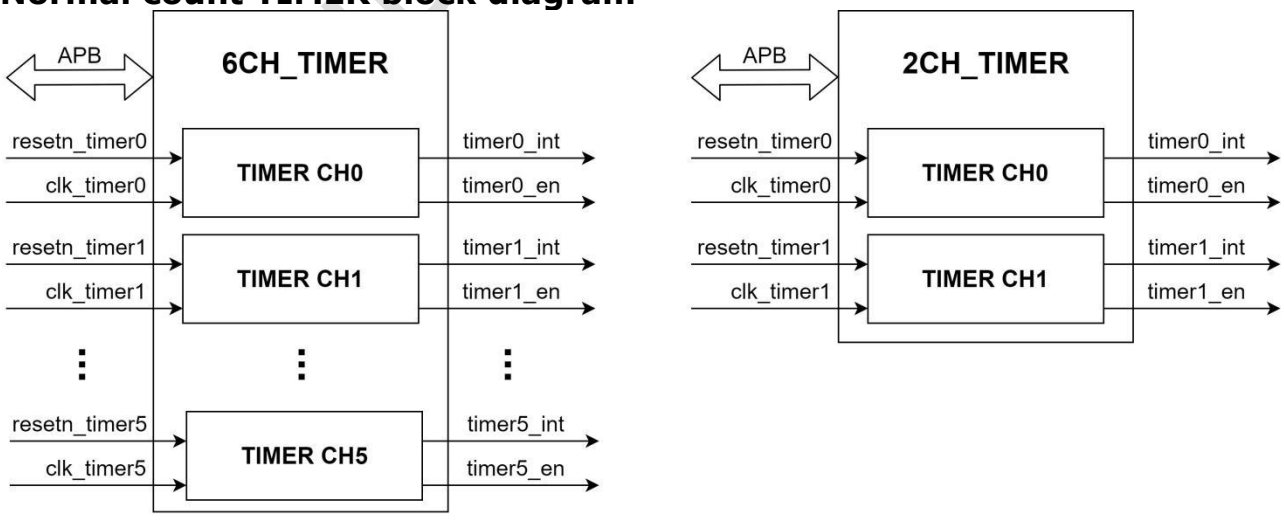


Fig. 9-1 Normal Timer Block Diagram

Fig. 9-1 show 6-channel TIMER and 2-channel TIMER. At 6CH_TIMER, the TIMER ch0~ch4 are the normal decrement count TIMER and ch5 is the normal increment count TIMER. At 2CH_TIMER, ch0 and ch1 are common normal increment count TIMER. APB is the read and write interface of timer register. reset_timern and clk_timern are reset and count clock for each channel timer. Timern_int and timern_en are interrupt and clock enable for each channel timer respectively. (timern is the signal corresponding to TIMER CHn)

9.2.2 HPTIMER block diagram

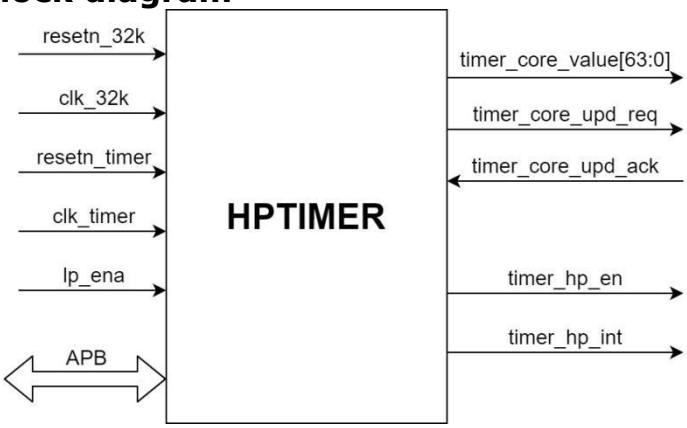


Fig. 9-2 HPTIMER Block Diagram

Fig. 9-2 shows the block diagram of HPTIMER.

HPTIMER has a set of APB interfaces for reading and writing registers.

lp_ena is the low power enable signal sent by PMU to HPTIMER, and it is also the indication signal of clk_timer frequency switching.

timer_core_value[63:0], timer_core_upd_req, timer_core_upd_ack is the interaction signal with core. timer_core_value[63:0] is the count update value, timer_core_upd_req is the update count request signal, timer_core_upd_ack is the update count response signal.

timer_hp_en is the enabling signal of clk_32k and clk_timer clocks. timer_hp_int is the interrupt output of HPTIMER.

9.3 Function Description

9.3.1 TIMER clock

The clock frequency of the counting clock source of NICTIMER and NDCTIMER is 24MHz. The clock source clock of clk_32k of HPTIMER is 32KHz, and the clock frequency of clk_timer clock source is switched between 24MHz and 32KHz.

9.3.2 Programming sequence

9.3.2.1 NDCTIMER and NICTIMER programming sequence

1. Initialize the TIMER by the TIMER_TIMERn_CONTROLREG (0≤n≤5) register:
   (1) Disable the timer by writing a "0" to the timer enable bit (bit 0). Accordingly, the timer_en output signal is de-asserted.
   (2) Program the timer mode—free-running or user-defined—by writing a "0" or "1" respectively, to the timer mode bit (bit 1).
   (3) 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 TIMER_TIMERn_LOAD_COUNT1 (0≤n≤5) and TIMER_TIMERn_LOAD_COUNT0 (0≤n≤5) register.
3. Enable the TIMER by writing a "1" to bit 0 of TIMER_TIMERn_CONTROLREG (0≤n≤5).

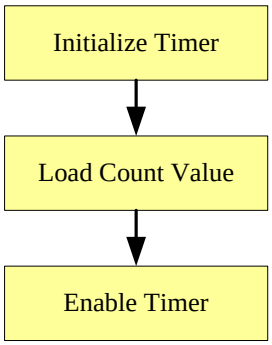


Fig. 9-3 Timer Usage Flow

9.3.2.2 HPTIMER programming sequence

9.3.2.2.1 Normal mode

1. Initialize the HPTIMER by the TIMER_HP_CTRL register:
   (1) Disable the HPTIMER by writing a '0' to the timer enable bit (bit 0). Accordingly, the timer_hp_en output signal is de-asserted.
   (2) Program the timer mode, normal mode by writing a 0x0 to the timer mode bits (bit 1 to bit 2).
   (3) Program the count mode, free-running or user-defined by writing a '0' or '1' to the count mode bit (bit 3).
2. Load the HPTIMER count value into the TIMER_HP_LOAD_COUNT1 and TIMER_HP_LOAD_ COUNT0 register.
3. Enable the HPTIMER by writing a '1' to bit 0 of TIMER_HP_CTRL.

If you want to do another count, repeat steps 1 to 3.

9.3.2.2.2 Hardware adjust mode

1. Initialize the HPTIMER by the TIMER_HP_CTRL register:
   (1) Disable the HPTIMER by writing a '0' to the timer enable bit (bit 0). Accordingly, the timer_hp_en output signal is de-asserted.
   (2) Program the timer mode, normal mode by writing a 0x1 to the timer mode bits (bit 1 to bit 2).
   (3) Program the count mode, free-running or user-defined by writing a '0' or '1' to the count mode bit (bit 3).
2. Initialize the HPTIMER by the TIMER_HP_INTR_STATUS register. Write 0x7 to TIMER_HP_ INTR_STATUS register to clear TIMER_HP_INTR_STATUS register.
3. Load the HPTIMER count value into the TIMER_HP_LOAD_COUNT1 and TIMER_HP_LOAD_ COUNT0 register.
4. Configure the HPTIMER GCD register.
   (1) Divide the least common multiple of clk_timer and clk_32k clock cycle by clk_timer clock cycle and configure it in TIMER_HP_T24_GCD register.
   (2) Divide the least common multiple of clk_timer and clk_32k clock cycle by clk_32k clock cycle and configure it in TIMER_HP_T32_GCD register.
5. Enable the HPTIMER by writing a '1' to bit 0 of TIMER_HP_CTRL.
6. Read out the INTR_STATUS register until the ini_adj_done bit segment (bit 1) is '1', which means that the HPTIMER 32K count is initially adjusted. The low power mode can only be entered after the initial adjustment is completed.
7. The chip enters and exits the low power consumption mode at any time in between.
8. Read out the INTR_STATUS register until the sync_done bit segment (bit 2) is '1', indicating that the HPTIMER count is adjusted.
9. Write 0x7 to TIMER_HP_ INTR_STATUS register to clear TIMER_HP_INTR_STATUS register.

If the chip enters and exits the low power consumption mode again, repeat steps 7 to 9.

9.3.2.2.3 Software adjust mode

1. Initialize the HPTIMER by the TIMER_HP_CTRL register:
   (1) Disable the HPTIMER by writing a '0' to the timer enable bit (bit 0). Accordingly, the timer_hp_en output signal is de-asserted.

(2) Program the timer mode, normal mode by writing a 0x2 to the timer mode bits (bit 1 to bit 2).
(3) Program the count mode, free-running or user-defined by writing a '0' or '1' to the count mode bit (bit 3).

2. Initialize the HPTIMER by the TIMER_HP_INTR_STATUS and BEGIN_END_VALID register. Write 0x7 to TIMER_HP_ INTR_STATUS register to clear TIMER_HP_INTR_STATUS register. write 0x3 to BEGIN_END_VALID register to clear BEGIN_END_VALID register.
3. Load the HPTIMER count value into the TIMER_HP_LOAD_COUNT1 and TIMER_HP_LOAD_ COUNT0 register.
4. Enable the HPTIMER by writing a '1' to bit 0 of TIMER_HP_CTRL.
5. The chip enters and exits the low power consumption mode at any time in between.
6. Read out BEGIN_END_VALID register until the value of it is 0x3, which indicates that the chip has entered and exited the low power consumption mode, then you can adjust the timer count value. Then write 0x3 to BEGIN_END_VALID register to clear the register.
7. The count value of HPTIMER compensation.
   (1) Read out TIMER_HP_T24_32BEGIN1, TIMER_ HP_T24_32BEGIN0, TIMER_HP_T32_ 24END1 and TIMER_HP_T32_24END0 register.
   (2) configure TIMER_HP_T24_DELAT_COUNT1 and TIMER_HP_T24_DELAT_COUNT0 registers to compensate for the count values according to the values of these registers read out above.
   (3) Write 0x1 to TIMER_HP_SYNC_REQ register to start compensation counting.
   (4) Read out the INTR_STATUS register until the sync_done bit segment (bit 2) is '1', indicating that the compensation count is completed.
8. Write 0x7 to TIMER_HP_ INTR_STATUS register to clear TIMER_HP_INTR_STATUS register.

If the chip enters and exits the low power consumption mode again, repeat steps 5 to 8.

9.3.3 Loading a timer count value

For the descending TIMER(that is, the value from which it counts down). The initial value for each TIMER is loaded into the TIMER using the load count register (TIMER_TIMERn_LOAD_ COUNT1 and TIMER_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 TIMER(that is, the value from which it counts up). The initial value for each timer is zero. The count register will count up to the value loaded in the register TIMER_ TIMERn_LOAD_COUNT1 and TIMER_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 TIMER_TIMERn_LOAD_COUNT1 and TIMER_ TIMERn_LOAD_COUNT0, when timer is configured into free-running mode.

9.3.4 Timer mode selection

- User-defined count mode – TIMER loads TIMER_TIMERn_LOAD_COUNT1 and TIMER_ TIMERn_LOAD_ COUNT0 registers (for descending TIMER) or zero (for incremental TIMER) as initial value. When the TIMER counts down to 0 (for descending TIMER) or counts up to the value in TIMER_TIMERn_LOAD_COUNT1 and TIMER_TIMERn_ LOAD_ COUNT0 (for incremental TIMER), 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 TIMER_TIMERn_LOAD_COUNT1 and TIMER_ TIMERn_LOAD_COUNT0(for descending TIMER) or zero(for incremental TIMER) register as initial value. TIMER will automatically reload the count register, when timer counts down to 0(for descending TIMER) or counts up to the value in TIMER_TIMERn_LOAD_ COUNT1 and TIMER_TIMERn_LOAD_COUNT0 (for incremental TIMER).

9.4 Register Description

9.4.1 Registers Summary

9.4.1.1 Normal Count TIMER Register Summary

For 2-channel normal count TIMER, the base address of channel 1 is greater than that of channel 0 by 0x20. For 6-channel normal count TIMER, the base address of channel 1 is greater than that of channel 0 by 0x20, the base address of channel 2 is greater than that of channel 1 by 0x20, and so on. The revision register offset of 2-channel / 6-channel normal count TIMER is 0xf0.

If you want to know that each TIMER is 2-channel or 6-channel, please refer to the "TIMER attributes" section at the end of "Chapter TIMER".

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_LOAD_COUNT0	0x0000 +n*0x20	W	0x00000000	Timern Load Count Register 0
TIMER_TIMERn_LOAD_COUNT1	0x0004 +n*0x20	W	0x00000000	Timern Load Count Register 1.Higher 32 bits Value to be loaded into Timer n. This is the value from which counting commences
TIMER_TIMERn_CURRENT_VALUE0	0x0008 +n*0x20	W	0x00000000	Timern Current Value Register 0
TIMER_TIMERn_CURRENT_VALUE1	0x000c +n*0x20	W	0x00000000	Timern Current Value Register 1.High 32 bits of Current Value of Timer n
TIMER_TIMERn_CONTROLREG	0x0010 +n*0x20	W	0x00000000	Timern Control Register
TIMER_TIMERn_INTSTATUS	0x0018 +n*0x20	W	0x00000000	Timern Interrupt Status Register
TIMER_2CH_REVISION	0x00F0	W	0x15650302	2-channel TIMER version
TIMER_6CH_REVISION	0x00F0	W	0x11972006	6-channel TIMER version

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

9.4.1.2 HPTIMER Register Summary

Name	Offset	Size	Reset Value	Description
TIMER_HP_REVISION	0x0000	W	0x13D10201	HPTIMER version
TIMER_HP_CTRL	0x0004	W	0x00000000	HPTIMER control
TIMER_HP_INTR_EN	0x0008	W	0x00000000	Interrupt musk
TIMER_HP_T24_GCD	0x000C	W	0x00000000	Common multiple div 24M
TIMER_HP_T32_GCD	0x0010	W	0x00000000	Common multiple div 32K
TIMER_HP_LOAD_COUNT0	0x0014	W	0x00000000	Low 32bits of load count value
TIMER_HP_LOAD_COUNT1	0x0018	W	0x00000000	High 32bits of load count value
TIMER_HP_T24_DELAT_COUNT0	0x001C	W	0x00000000	Low 32bits of 24M clock delay time count value
TIMER_HP_T24_DELAT_COUNT1	0x0020	W	0x00000000	High 32bits of 24M clock delay time count value

Name	Offset	Size	Reset Value	Description
TIMER_HP_CURR_32K_VALUE0	0x0024	W	0x00000000	Low 32bits of current cnt_32k value
TIMER_HP_CURR_32K_VALUE1	0x0028	W	0x00000000	High 32bits of current cnt_32k value
TIMER_HP_CURR_TIMER_VALUE0	0x002C	W	0x00000000	Low 32bits of current timer_cnt value
TIMER_HP_CURR_TIMER_VALUE1	0x0030	W	0x00000000	High 32bits of current timer_cnt value
TIMER_HP_T24_32BEGIN0	0x0034	W	0x00000000	Low 32 bits of low power begin
TIMER_HP_T24_32BEGIN1	0x0038	W	0x00000000	High 32 bits of low power begin
TIMER_HP_T32_24END0	0x003C	W	0x00000000	Low 32 bits of low power end
TIMER_HP_T32_24END1	0x0040	W	0x00000000	High 32 bits of low power end
TIMER_HP_BEGIN_END_VALID	0x0044	W	0x00000000	Low power count value valid
TIMER_HP_SYNC_REQ	0x0048	W	0x00000000	Synchronize request
TIMER_HP_INTR_STATUS	0x004C	W	0x00000000	Interrupt status

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

9.4.2 Detail Register Description

9.4.2.1 Normal Count TIMER Detail Register Description

TIMER_TIMERn_LOAD_COUNT0

Address: Operational Base + offset (0x0000+n*0x20)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_0 Lower 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+n*0x20)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_1 Higher 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+n*0x20)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value0 Lower 32 bits of Current Value of Timer n.

TIMER_TIMERn_CURRENT_VALUE1

Address: Operational Base + offset (0x000c+n*0x20)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value1 Higher 32 bits of Current Value of Timer n.

TIMER_TIMERn_CONTROLREG

Address: Operational Base + offset (0x0010+n*0x20)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	Reserved
2	RW	0x0	timer_int_en Timer interrupt enable 1'b0: Disable 1'b1: Enable
1	RW	0x0	timer_mode Timer mode 1'b0: Free-running mode 1'b1: User-defined count mode
0	RW	0x0	timer_en Timer enable 1'b0: Disable 1'b1: Enable

TIMER_TIMERn_INTSTATUS

Address: Operational Base + offset (0x0018+n*0x20)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	Reserved
0	RO	0x0	timern_int This register contains the interrupt status for timern.

TIMER_2CH_REVISION

Address: Operational Base + offset (0x00F0)

Bit	Attr	Reset Value	Description
31:16	RO	0x1565	svn_revision SVN revision: 16'h1565
15:10	RO	0x00	reserved
9	RW	0x1	ch1_type Channel 1 is a count up counter.
8	RW	0x1	ch0_type Channel 0 is a count up counter.
7:0	RO	0x02	ip_function IP function: 8'h2

TIMER_6CH_REVISION

Address: Operational Base + offset (0x00F0)

Bit	Attr	Reset Value	Description
31:16	RW	0x1197	svn_revision SVN revision: 16'h1197
15:14	RO	0x0	reserved
13	RW	0x1	ch5_type Channel 5 is a count up counter.
12	RW	0x0	ch4_type Channel 4 is a count down counter.
11	RW	0x0	ch3_type Channel 3 is a count down counter.
10	RW	0x0	ch2_type Channel 2 is a count down counter.
9	RW	0x0	ch1_type Channel 1 is a count down counter.

Bit	Attr	Reset Value	Description
8	RW	0x0	ch0_type Channel 0 is a count down counter.
7:0	RW	0x06	ip_function IP function: 8'h6

9.4.2.2 HPTIMER Detail Register Description

TIMER_HP_REVISION

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x19e1	svn_revision SVN revision: 16'h19e1
15:10	RO	0x00	reserved
9:8	RW	0x2	ch_type Channel 0 is a low power recovery self-correction count up counter.
7:0	RO	0x01	ip_function IP function: 8'h1.

TIMER_HP_CTRL

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	count_mode Timer count mode 1'b0: Free-running mode 1'b1: User-defined count mode
2:1	RW	0x0	timer_mode Select timer as which timer 2'b00: Normal timer 2'b01: Adjust timer with hardware adjust 2'b10: Adjust timer with software adjust
0	RW	0x0	timer_en Timer enable 1'b0: Disable 1'b1: Enable

TIMER_HP_INTR_EN

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	sync_done_intr_en Synchronization done interrupt enable 1'b0: Disable 1'b1: Enable
1	RW	0x0	ini_adj_done_intr_en Initial adjust done interrupt enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	count_reach_intr_en Timer count reach load_count interrupt enable 1'b0: Disable 1'b1: Enable

TIMER_HP_T24_GCD

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t24_gcd The least common multiple of 24M and 32K clock cycles divided by 24M clock cycles.

TIMER_HP_T32_GCD

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t32_gcd The least common multiple of 24M and 32K clock cycles divided by 32K clock cycles.

TIMER_HP_LOAD_COUNT0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count0 Low 32bits of load count value.

TIMER_HP_LOAD_COUNT1

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count1 High 32bits of load count value.

TIMER_HP_T24_DELAT_COUNT0

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t24_delay_time_count0 Low 32bits of 24M clock delay time count value.

TIMER_HP_T24_DELAT_COUNT1

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t24_delay_time_count1 High 32bits of 24M clock delay time count value.

TIMER_HP_CURR_32K_VALUE0

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_32k_value0 Low 32bits of current cnt_32k value.

TIMER_HP_CURR_32K_VALUE1

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_32k_value1 High 32bits of current cnt_32k value.

TIMER_HP_CURR_TIMER_VALUE0

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value0 Low 32bits of current timer_cnt value.

TIMER_HP_CURR_TIMER_VALUE1

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value1 High 32bits of current timer_cnt value.

TIMER_HP_T24_32BEGIN0

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	t24_32begin0 24M switches to 32K clock, with a low 32 bits of timer_cnt value.

TIMER_HP_T24_32BEGIN1

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	t24_32begin1 24M switches to 32K clock, with a high 32 bits of timer_cnt value.

TIMER_HP_T32_24END0

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	t32_24end0 32K switches to 24M clock, with a low 32 bits of timer_cnt value.

TIMER_HP_T32_24END1

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	t32_24end1 32K switches to 24M clock, with a high 32 bits of timer_cnt value.

TIMER_HP_BEGIN_END_VALID

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	W1C	0x0	t32_24end_valid T32_24END value valid 1'b0: Invalid 1'b1: Valid
0	W1C	0x0	t24_32begin_valid T24_32BEGIN value valid 1'b0: Invalid 1'b1: Valid

TIMER_HP_SYNC_REQ

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	R/WSC	0x0	sync_request Timer_cnt synchronize request.

TIMER_HP_INTR_STATUS

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	W1C	0x0	sync_done Timer_cnt synchronization done.

Bit	Attr	Reset Value	Description
1	W1C	0x0	ini_adj_done Cnt_32k initial adjust done.
0	W1C	0x0	count_reach Normal counter: timer_cnt reach load_count value.

9.5 Application Notes

9.5.1 Clock and Enable

In the chip, the timer_clk is from 24MHz XIN_OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (0≤n≤5)), the timer en 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.

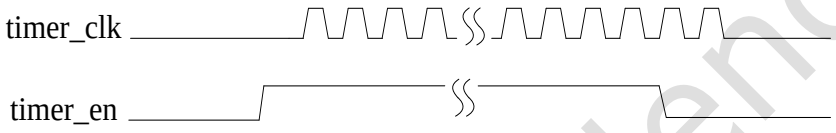


Fig. 9-4 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

9.5.2 HPTIMER application notes

- Initial adjustment, synchronous counting adjustment time is about 2ms.
- When HPTIMER is used as software / hardware adjustment count, the count reaches the TIMER_HP_LOAD_COUNT value, the interrupt status bit will be set, but the count will not stop and continue to count.
- The timer_en of the CTRL register will output as a reference signal for timer clock gating. When timer_en is '0', it will gate the 24M and 32K clocks of the input timer. The application is only allowed to re-config registers when timer_en is low.

9.5.3 TIMER attributes

Table 9-1 shows the TIMER name corresponding to each TIMER, the total number of channels and their PD. It also shows the base address of each TIMER, in which the difference of TIMER address offset between adjacent channels in each TIMER is 0x20.

Table 9-1 TIMER attribute

TIMER Name	CH NUM	PD	Base Address
TIMER_2CH_S	2	pd_peri	0xff590000
TIMER_6CH_NS	6	pd_peri	0xff580000
HP_TIMER	1	pd_pmu	0xff2f0000

Chapter 10 DMA Controller (DMAC)

10.1 Overview

RV1106 supports a Direct Memory Access (DMA) Controller. DMAC support transfers between memory and memory, peripheral and memory. DMAC is under Non-secure state after reset, and the Secure state can be changed by configurable SGRF module.
DMAC supports the following features:

- Supports 21 peripheral requests
- Up to 64 bits 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 DMAC request mapping scheme.

Table 10-1 DMAC Request Mapping Table

DMAC0		
Req number	**Source**	**Polarity**
0	SPI0_RX	High level
1	SPI0_TX	High level
2	SPI1_RX	High level
3	SPI1_TX	High level
6	UART0_RX	High level
7	UART0_TX	High level
8	UART1_RX	High level
9	UART1_TX	High level
10	UART2_RX	High level
11	UART2_TX	High level
12	UART3_RX	High level
13	UART3_TX	High level
14	UART4_RX	High level
15	UART4_TX	High level
16	UART5_RX	High level
17	UART5_TX	High level
18	PWM0	High level
19	PWM1	High level
20	PWM2	High level
21	I2S0_8CH_RX	High level
22	I2S0_8CH_TX	High level

DMAC supports incrementing-address burst and fixed-address burst. But in the case of access to SPI and UART at byte or half word size, DMAC only supports fixed-address burst and the address must be aligned to word.

10.2 Block Diagram

Following figure shows the block diagram of DMAC.

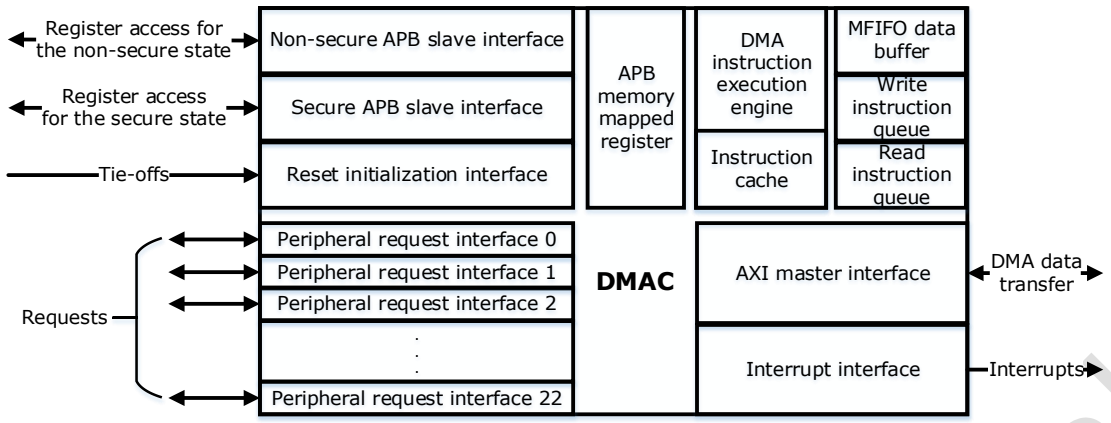


Fig. 10-1 DMAC Block Diagram

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.

10.3 Function Description

10.3.1 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.

10.3.2 Operation states

Following figure shows the operating states for the DMA manager thread and DMA channel threads.

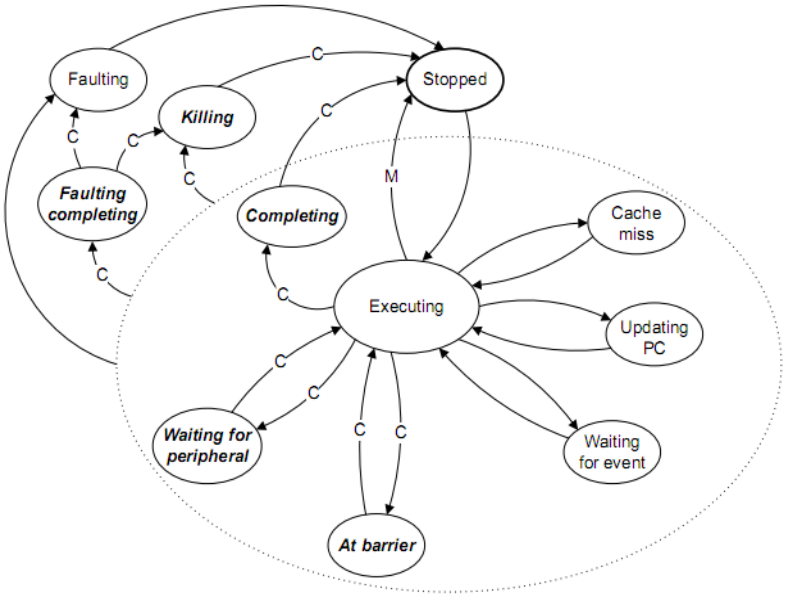


Fig. 10-2 DMAC Operation State

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.

10.4 Register Description

10.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DMAC_DSR	0x0000	W	0x00000000	DMA Manager Status Register
DMAC_DPC	0x0004	W	0x00000000	DMA Program Counter Register
DMAC_INTEN	0x0020	W	0x00000000	Interrupt Enable Register
DMAC_EVENT_RIS	0x0024	W	0x00000000	Event-Interrupt Raw Status Register
DMAC_INTMIS	0x0028	W	0x00000000	Interrupt Status Register
DMAC_INTCLR	0x002c	W	0x00000000	Interrupt Clear Register
DMAC_FSRD	0x0030	W	0x00000000	Fault Status DMA Manager Register
DMAC_FSRC	0x0034	W	0x00000000	Fault Status DMA Channel Register
DMAC_FTRD	0x0038	W	0x00000000	Fault Type DMA Manager Register
DMAC_FTR0	0x0040	W	0x00000000	Fault Type DMA Channel 0 Register
DMAC_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel 1 Register
DMAC_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel 2 Register
DMAC_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel 3 Register
DMAC_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel 4 Register
DMAC_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel 5 Register

Name	Offset	Size	Reset Value	Description
DMAC_FTR6	0x0058	W	0x00000000	Fault Type DMA Channel 6 Register
DMAC_FTR7	0x005c	W	0x00000000	Fault Type DMA Channel 7 Register
DMAC_CSR0	0x0100	W	0x00000000	Channel 0 Status Register
DMAC_CPC0	0x0104	W	0x00000000	Channel 0 Program Counter Register
DMAC_CSR1	0x0108	W	0x00000000	Channel 1 Status Register
DMAC_CPC1	0x010c	W	0x00000000	Channel 1 Program Counter Register
DMAC_CSR2	0x0110	W	0x00000000	Channel 2 Status Register
DMAC_CPC2	0x0114	W	0x00000000	Channel 2 Program Counter Register
DMAC_CSR3	0x0118	W	0x00000000	Channel 3 Status Register
DMAC_CPC3	0x011c	W	0x00000000	Channel 3 Program Counter Register
DMAC_CSR4	0x0120	W	0x00000000	Channel 4 Status Register
DMAC_CPC4	0x0124	W	0x00000000	Channel 4 Program Counter Register
DMAC_CSR5	0x0128	W	0x00000000	Channel 5 Status Register
DMAC_CPC5	0x012c	W	0x00000000	Channel 5 Program Counter Register
DMAC_CSR6	0x0130	W	0x00000000	Channel 6 Status Register
DMAC_CPC6	0x0134	W	0x00000000	Channel 6 Program Counter Register
DMAC_CSR7	0x0138	W	0x00000000	Channel 7 Status Register
DMAC_CPC7	0x013c	W	0x00000000	Channel 7 Program Counter Register
DMAC_SAR0	0x0400	W	0x00000000	Channel 0 Source Address Register
DMAC_DAR0	0x0404	W	0x00000000	Channel 0 Destination Address Register
DMAC_CCR0	0x0408	W	0x00000000	Channel 0 Channel Control Register
DMAC_LC0_0	0x040c	W	0x00000000	Channel 0 Loop Counter 0 Register
DMAC_LC1_0	0x0410	W	0x00000000	Channel 0 Loop Counter 1 Register
DMAC_SAR1	0x0420	W	0x00000000	Channel 1 Source Address Register
DMAC_DAR1	0x0424	W	0x00000000	Channel 1 Destination Address Register
DMAC_CCR1	0x0428	W	0x00000000	Channel 1 Channel Control Register
DMAC_LC0_1	0x042c	W	0x00000000	Channel 1 Loop Counter 0 Register
DMAC_LC1_1	0x0430	W	0x00000000	Channel 1 Loop Counter 1 Register
DMAC_SAR2	0x0440	W	0x00000000	Channel 2 Source Address Register
DMAC_DAR2	0x0444	W	0x00000000	Channel 2 Destination Address Register
DMAC_CCR2	0x0448	W	0x00000000	Channel 2 Channel Control Register

Name	Offset	Size	Reset Value	Description
DMAC_LC0_2	0x044c	W	0x00000000	Channel 2 Loop Counter 0 Register
DMAC_LC1_2	0x0450	W	0x00000000	Channel 2 Loop Counter 1 Register
DMAC_SAR3	0x0460	W	0x00000000	Channel 3 Source Address Register
DMAC_DAR3	0x0464	W	0x00000000	Channel 3 Destination Address Register
DMAC_CCR3	0x0468	W	0x00000000	Channel 3 Channel Control Register
DMAC_LC0_3	0x046c	W	0x00000000	Channel 3 Loop Counter 0 Register
DMAC_LC1_3	0x0470	W	0x00000000	Channel 3 Loop Counter 1 Register
DMAC_SAR4	0x0480	W	0x00000000	Channel 4 Address Register
DMAC_DAR4	0x0484	W	0x00000000	Channel 4 Destination Address Register
DMAC_CCR4	0x0488	W	0x00000000	Channel 4 Channel Control Register
DMAC_LC0_4	0x048c	W	0x00000000	Channel 4 Loop Counter 0 Register
DMAC_LC1_4	0x0490	W	0x00000000	Channel 4 Loop Counter 1 Register
DMAC_SAR5	0x04a0	W	0x00000000	Channel 5 Address Register
DMAC_DAR5	0x04a4	W	0x00000000	Channel 5 Destination Address Register
DMAC_CCR5	0x04a8	W	0x00000000	Channel 5 Channel Control Register
DMAC_LC0_5	0x04ac	W	0x00000000	Channel 5 Loop Counter 0 Register
DMAC_LC1_5	0x04b0	W	0x00000000	Channel 5 Loop Counter 1 Register
DMAC_SAR6	0x04c0	W	0x00000000	Channel 6 Source Address Register
DMAC_DAR6	0x04c4	W	0x00000000	Channel 6 Destination Address Register
DMAC_CCR6	0x04c8	W	0x00000000	Channel 6 Channel Control Register
DMAC_LC0_6	0x04cc	W	0x00000000	Channel 6 Loop Counter 0 Register
DMAC_LC1_6	0x04d0	W	0x00000000	Channel 6 Loop Counter 1 Register
DMAC_SAR7	0x04e0	W	0x00000000	Channel 7 Source Address Register
DMAC_DAR7	0x04e4	W	0x00000000	Channel 7 Destination Address Register
DMAC_CCR7	0x04e8	W	0x00000000	Channel 7 Channel Control Register
DMAC_LC0_7	0x04ec	W	0x00000000	Channel 7 Loop Counter 0 Register
DMAC_LC1_7	0x04f0	W	0x00000000	Channel 7 Loop Counter 1 Register
DMAC_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMAC_DBGCMD	0x0d04	W	0x00000000	Debug Command Register

Name	Offset	Size	Reset Value	Description
DMAC_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMAC_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMAC_CR0	0x0e00	W	0x001f3075	Configuration Register 0
DMAC_CR1	0x0e04	W	0x000000b5	Configuration Register 1
DMAC_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMAC_CR3	0x0e0c	W	0x0000ffff	Configuration Register 3
DMAC_CR4	0x0e10	W	0x000fffff	Configuration Register 4
DMAC_CRDn	0x0e14	W	0x07ff7f73	Configuration Register
DMAC_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*

10.4.2 Detail Register Description

DMAC_DSR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	dns 1'b0: DMA manager operates in the Secure state 1'b1: DMA manager operates in the Non-secure state
8:4	RO	0x00	wakeup_event 5'h0: event[0] 5'h1: event[1] 5'h2: event[2] … 5'h1f: event[31]
3:0	RO	0x0	dma_status 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'hf: Faulting Others: Reserved

DMAC_DPC

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_mgr Program counter for the DMA manager thread

DMAC_INTEN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	event_irq_select Bit [N] 1'b0: 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. 1'b1: 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.

DMAC_EVENT_RIS

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dmasev_active Bit [N] 1'b0: Event N is inactive or irq[N] is LOW 1'b1: Event N is active or irq[N] is HIGH

DMAC_INTMIS

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	irq_status Bit [N] 1'b0: Interrupt N is inactive and therefore irq[N] is LOW 1'b1: Interrupt N is active and therefore irq[N] is HIGH

DMAC_INTCLR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	irq_clr Bit [N] 1'b0: The status of irq[N] does not change 1'b1: The DMAC sets irq[N] LOW if the DMAC_INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change.

DMAC_FSRD

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	fs_mgr 1'b0: The DMA manager thread is not in the Faulting state 1'b1: The DMA manager thread is in the Faulting state

DMAC_FSRC

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	fault_status Bit [N] 1'b0: No fault is present on DMA channel N 1'b1: DMA channel N is in the Faulting or Faulting completing state

DMAC_FTRD

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: Instruction that generated an abort was read from the debug interface
29:17	RO	0x0	reserved
16	RO	0x0	instr_fetch_err Performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	mgr_evnt_err 1'b0: DMA manager has appropriate security to execute DMAWFE or DMASEV 1'b1: DMA manager thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt
4	RO	0x0	dmago_err 1'b0: DMA manager has appropriate security to execute DMAGO 1'b1: 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	operand_invalid The configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand
0	RW	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction

DMAC_FTR0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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

Bit	Attr	Reset Value	Description
18	RO	0x0	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.

Bit	Attr	Reset Value	Description
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.

Bit	Attr	Reset Value	Description
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. 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	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.

Bit	Attr	Reset Value	Description
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR6

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.

Bit	Attr	Reset Value	Description
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: 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	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_FTR7

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr Memory or from the debug interface. 1'b0: Instruction that generated an abort was read from system memory 1'b1: 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	data_read_err Thread performs a data read. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err Thread performs a data write. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err Thread performs an instruction fetch. 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: 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	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	ch_rdwr_err To perform a secure read or secure write. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: 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	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions. 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFP to wait for a secure peripheral b. DMALDP or DMASTP to notify a secure peripheral c. DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: A DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: A DMA channel thread in the Non-secure state attempted to execute either: a. DMAWFE to wait for a secure event b. DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid Valid for the configuration of the DMAC. 1'b0: Valid operand 1'b1: Invalid operand This fault is a precise abort.
0	RO	0x0	undef_instr 1'b0: Defined instruction 1'b1: Undefined instruction This fault is a precise abort.

DMAC_CSR0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status Channel 0 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC0

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 0 thread

DMAC_CSR1

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 1 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC1

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 1 thread

DMAC_CSR2

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 2 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC2

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 2 thread

DMAC_CSR3

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 3 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC3

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 3 thread

DMAC_CSR4

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 4 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC4

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 4 thread

DMAC_CSR5

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 5 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC5

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 5 thread

DMAC_CSR6

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 6 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC6

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 6 thread

DMAC_CSR7

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number Indicate the event or peripheral number that the channel is waiting for. 5'h0: DMA channel is waiting for event, or peripheral, 0 5'h1: DMA channel is waiting for event, or peripheral, 1 5'h2: DMA channel is waiting for event, or peripheral, 2 … 5'h1f: DMA channel is waiting for event, or peripheral, 31

Bit	Attr	Reset Value	Description
3:0	RO	0x0	channel_status Channel 7 status. 4'h0: Stopped 4'h1: Executing 4'h2: Cache miss 4'h3: Updating PC 4'h4: Waiting for event 4'h5: At barrier 4'h7: Waiting for peripheral 4'h8: Killing 4'h9: Completing 4'he: Faulting completing 4'hf: Faulting Others: Reserved

DMAC_CPC7

Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 7 thread

DMAC_SAR0

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 0

DMAC_DAR0

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 0

DMAC_CCR0

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_0

Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_0

Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR1

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 1

DMAC_DAR1

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 1

DMAC_CCR1

Address: Operational Base + offset (0x0428)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_1

Address: Operational Base + offset (0x042c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_1

Address: Operational Base + offset (0x0430)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR2

Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 2

DMAC_DAR2

Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 2

DMAC_CCR2

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_2

Address: Operational Base + offset (0x044c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_2

Address: Operational Base + offset (0x0450)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR3

Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 3

DMAC_DAR3

Address: Operational Base + offset (0x0464)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 3

DMAC_CCR3

Address: Operational Base + offset (0x0468)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_3

Address: Operational Base + offset (0x046c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_3

Address: Operational Base + offset (0x0470)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR4

Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 4

DMAC_DAR4

Address: Operational Base + offset (0x0484)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 4

DMAC_CCR4

Address: Operational Base + offset (0x0488)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_4

Address: Operational Base + offset (0x048c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_4

Address: Operational Base + offset (0x0490)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR5

Address: Operational Base + offset (0x04a0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 5

DMAC_DAR5

Address: Operational Base + offset (0x04a4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 5

DMAC_CCR5

Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_5

Address: Operational Base + offset (0x04ac)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_5

Address: Operational Base + offset (0x04b0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR6

Address: Operational Base + offset (0x04c0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 6

DMAC_DAR6

Address: Operational Base + offset (0x04c4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 6

DMAC_CCR6

Address: Operational Base + offset (0x04c8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_6

Address: Operational Base + offset (0x04cc)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_6

Address: Operational Base + offset (0x04d0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_SAR7

Address: Operational Base + offset (0x04e0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 7

DMAC_DAR7

Address: Operational Base + offset (0x04e4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destination data for DMA channel 7

DMAC_CCR7

Address: Operational Base + offset (0x04e8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	dst_burst_size 3'h0: Writes 1 byte per beat 3'h1: Writes 2 bytes per beat 3'h2: Writes 4 bytes per beat 3'h3: Writes 8 bytes per beat 3'h4: Writes 16 bytes per beat Others: 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	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'h0: 1 data transfer 4'h1: 2 data transfers 4'h2: 3 data transfers … 4'hf: 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	src_burst_size 3'h0: Reads 1 byte per beat 3'h1: Reads 2 bytes per beat 3'h2: Reads 4 bytes per beat 3'h3: Reads 8 bytes per beat 3'h4: Reads 16 bytes per beat Others: 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	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMAC_LC0_7

Address: Operational Base + offset (0x04ec)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 0 iterations

DMAC_LC1_7

Address: Operational Base + offset (0x04f0)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	loop_counter_iterations Loop counter 1 iterations

DMAC_DBGSTATUS

Address: Operational Base + offset (0x0d00)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	dbgstatus 1'b0: Idle 1'b1: Busy

DMAC_DBGCMD

Address: Operational Base + offset (0x0d04)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	WO	0x0	dbgcmd 2'b00: Execute the instruction that the DMAC_DBGINST [1:0] Registers contain Others: Reserved

DMAC_DBGINST0

Address: Operational Base + offset (0x0d08)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	instruction_byte1 Instruction byte 1
23:16	WO	0x00	instruction_byte0 Instruction byte 0
15:11	RO	0x0	reserved
10:8	WO	0x0	channel_number 3'b000: DMA channel 0 3'b001: DMA channel 1 3'b010: DMA channel 2 … 3'b111: DMA channel 7
7:1	RO	0x0	reserved
0	WO	0x0	debug_thread 1'b0: DMA manager thread 1'b1: DMA channel

DMAC_DBGINST1

Address: Operational Base + offset (0x0d0c)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	instruction_byte5 Instruction byte 5
23:16	WO	0x00	instruction_byte4 Instruction byte 4
15:8	WO	0x00	instruction_byte3 Instruction byte 3
7:0	WO	0x00	instruction_byte2 Instruction byte 2

DMAC_CR0

Address: Operational Base + offset (0x0e00)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:17	RO	0x0f	num_events 5'h0: 1 interrupt output, irq[0] 5'h1: 2 interrupt outputs, irq[1:0] 5'h2: 3 interrupt outputs, irq[2:0] … 5'h1f: 32 interrupt outputs, irq[31:0]
16:12	RO	0x13	num_periph_req 5'h0: 1 peripheral request interface 5'h1: 2 peripheral request interfaces 5'h2: 3 peripheral request interfaces … 5'h1f: 32 peripheral request interfaces
11:7	RO	0x0	reserved
6:4	RO	0x7	num_chnls 3'b000: 1 DMA channel 3'b001: 2 DMA channels 3'b010: 3 DMA channels … 3'b111: 8 DMA channels
3	RO	0x0	reserved
2	RO	0x1	mgr_ns_at_rst 1'b0: boot_manager_ns was LOW 1'b1: boot_manager_ns was HIGH
1	RO	0x0	boot_en 1'b0: boot_from_pc was LOW 1'b1: boot_from_pc was HIGH
0	RO	0x1	periph_req 1'b0: The DMAC does not provide a peripheral request interface 1'b1: The DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies

DMAC_CR1

Address: Operational Base + offset (0x0e04)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RO	0xb	num_i_cache_lines 4'b0000: 1 i-cache line 4'b0001: 2 i-cache lines 4'b0010: 3 i-cache lines … 4'b1111: 16 i-cache lines
3	RO	0x0	reserved
2:0	RO	0x5	i_cache_len 3'b010: 4 bytes 3'b011: 8 bytes 3'b100: 16 bytes 3'b101: 32 bytes Others: Reserved

DMAC_CR2

Address: Operational Base + offset (0x0e08)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	boot_addr Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_CR3

Address: Operational Base + offset (0x0e0c)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000ffff	ins Bit [N] 1'b0: Assigns event<N> or irq[N] to the Secure state 1'b1: Assigns event<N> or irq[N] to the Non-secure state

DMAC_CR4

Address: Operational Base + offset (0x0e10)

Bit	Attr	Reset Value	Description
31:0	RO	0x000fffff	pns Bit [N] 1'b0: Assigns peripheral request interface N to the Secure state 1'b1: Assigns peripheral request interface N to the Non-secure state

DMAC_CRDn

Address: Operational Base + offset (0x0e14)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RO	0x07f	data_buffer_dep 10'b000000000: 1 line 10'b000000001: 2 lines … 10'b111111111: 1024 lines
19:16	RO	0xf	rd_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
15	RO	0x0	reserved
14:12	RO	0x7	rd_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
11:8	RO	0xf	wr_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
7	RO	0x0	reserved
6:4	RO	0x7	wr_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RO	0x3	data_width 3'b010: 32-bit 3'b011: 64-bit 3'b100: 128-bit Others:Reserved

DMAC_WD

Address: Operational Base + offset (0x0e80)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	wd_irq_only 1'b0: The DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1'b1: The DMAC sets irq_abort HIGH

10.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

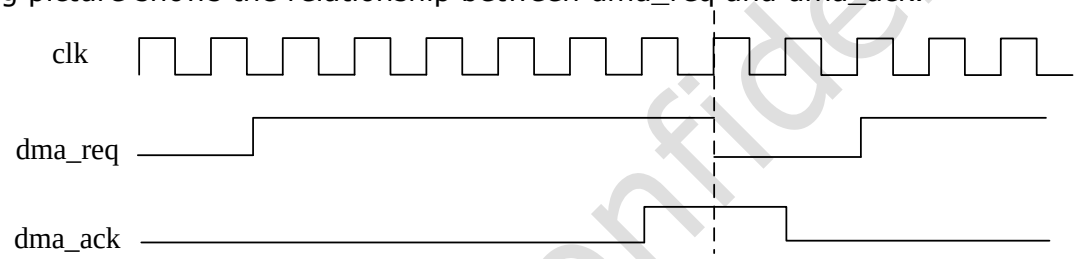


Fig. 10-3 DMAC Request and Acknowledge Timing

10.6 Application Notes

10.6.1 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 DMAC_DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DMAC_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 DMAC_DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

10.6.2 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 DMAC_FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the DMAC_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 DMAC_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 DMAC_FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the DMAC_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 DMAC_FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the DMAC_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 DMAC_CR3 Register.

DMASEV
The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the DMAC_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 DMAC_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 DMAC_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 DMAC_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 DMAC_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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the DMAC_FTRn(n=0~7) 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 DMAC_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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the DMAC_FTRn(n=0~7) 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 DMAC_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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) 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 DMAC_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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) 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 DMAC_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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the DMAC_FTRn(n=0~7) 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 DMAC_FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the DMAC_FTRn(n=0~7) Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

10.6.3 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 DMAC_CCRn(n=0~7) Register
- the DMAC_SARn(n=0~7) 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 DMAC_CCRn(n=0~7) Register
- the DMAC_DARn(n=0~7) 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 DMAC_CCRn(n=0~7) Register, to enable a DMA channel to perform an endian swap then you must set the corresponding DMAC_SARn(n=0~7) Register and the corresponding DMAC_DARn(n=0~7) 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 DMAC_CCRn(n=0~7) Register, DMAC_SARn(n=0~7) Register, and DMAC_DARn(n=0~7) 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.

10.6.4 Unaligned Transfers

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 split 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 split 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 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 10-2 Source Size in DMAC_CCRn

Source size in DMAC_CCRn	Allowed offset between DMAC_SARn and DMAC_DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

10.6.5 Interrupt Sharing between Channels

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

10.6.6 Instruction Sets

Table 10-3 DMAC 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

10.6.7 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 DMAC_SARn(n=0~7) Register or DMAC_DARn(n=0~7) 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 10-4 DMAC 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

DMAC_SARn(n=0~7) Register and sets ra to 0.

DAR

DMAC_DARn(n=0~7) 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 11 Temperature-Sensor ADC (TS-ADC)

11.1 Overview

TS-ADC Controller is used to control and get the temperature information. TS-ADC will convert for each enabled channel in loop after the initial setting and can be stopped by software. If you find that the temperature is High in a period of time, an interrupt is generated to the processor down-measures taken; if the temperature over a period of time High, the resulting TSHUT gave CRU module, let it reset the entire chip, or via GPIO give PMIC. Also, if you find that the temperature Low in a period of time, an interrupt can be generated.

TS-ADC Controller supports the following features:

- The temperature for High and Low interrupt can be configurable
- The temperature of system reset can be configurable
- The time interval of temperature detection can be configurable
- When detecting a high temperature, the time interval of temperature detection can be configurable
- High temperature debounce can be configurable
- An interrupt can be generated after TS-ADC converts all setting channel
- -40~125°C temperature range and 0.6°C temperature resolution

11.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

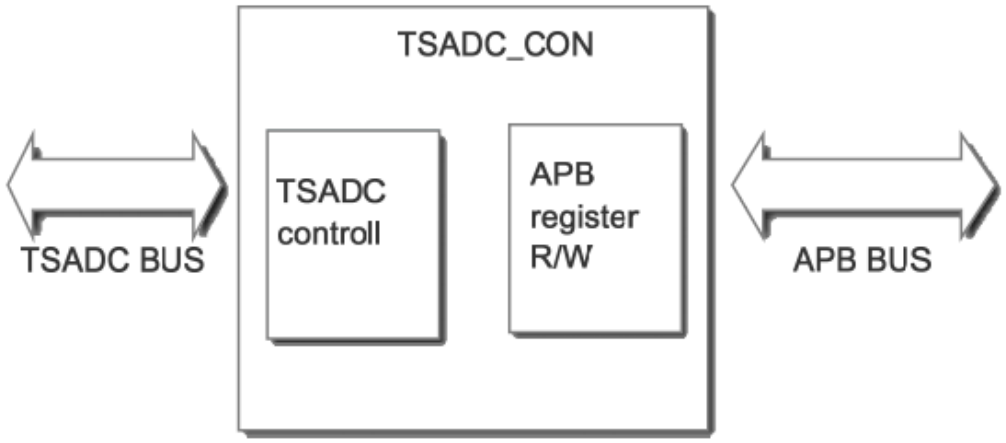


Fig. 11-1 TS-ADC Controller Block Diagram

11.3 Function Description

11.3.1 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.

11.3.2 TS-ADC Controller

This block is used to control the TS-ADC PHY to meet the conversion timing and receive the temperature information from TS-ADC PHY.

11.4 Register Description

11.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000000	User control
TSADC_AUTO_CON	0x0004	W	0x00000000	Auto control
TSADC_AUTO_STATUS	0x0008	W	0x00000000	Status used in auto mode
TSADC_AUTO_SRC	0x000C	W	0x00000000	Channel sel for tsadc in auto mode
TSADC_LT_EN	0x0010	W	0x00000000	Low temperature check logic enable
TSADC_HT_INT_EN	0x0014	W	0x00000000	High temperature interrupt enable
TSADC_CRU_EN	0x001C	W	0x00000000	Temperature violation to cru enable
TSADC_LT_INT_EN	0x0020	W	0x00000000	Low temperature interrupt enable
TSADC_HLT_INT_PD	0x0024	W	0x00000000	High and low temperature interrupt status
TSADC_EOC_HSHUT_PD	0x0028	W	0x00000000	High temperature shut and round int status
TSADC_DATA0	0x002C	W	0x00000000	Channel0 data
TSADC_COMP0_INT	0x006C	W	0x00000000	High temperature interrupt threshold for channel0
TSADC_COMP0_SHUT	0x010C	W	0x00000000	High temperature shut threshold for channel0
TSADC_HIGH_INT_DEBOUNCE	0x014C	W	0x00000003	High interrupt debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0150	W	0x00000003	High shut debounce
TSADC_AUTO_PERIOD	0x0154	W	0x00010000	Auto conversion period
TSADC_AUTO_PERIOD_HT	0x0158	W	0x00010000	Auto conversion period for high tempture
TSADC_COMP0_LOW_INT	0x015C	W	0x00000000	Low temperature threshold for channel0
TSADC_T_SETUP	0x019C	W	0x0000005F	Timing for setup
TSADC_T_PW_EN	0x0200	W	0x00000000	Timing for pw_en
TSADC_T_EN_CLK	0x0204	W	0x00000001	Timing for en_clk
TSADC_T_NON_OV	0x0208	W	0x00000002	Timong for non_ov
TSADC_T_HOLD	0x020C	W	0x00000013	Timing for hold
TSADC_Q_MAX	0x0210	W	0x00000200	Max data used for data inversion
TSADC_FLOW_CON	0x0218	W	0x00000003	Flow control
TSADC_T_PW_CLK	0x0220	W	0x00000030	Timing for pw_clk

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

11.4.2 Detail Registers Description

TSADC_USER_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable
15:9	RO	0x00	reserved
8	RO	0x0	adc_status 1'b0: ADC stop 1'b1: Conversion in progress

Bit	Attr	Reset Value	Description
7	R/W SC	0x0	start When software writes 1 to this bit, start-of-conversion will be asserted. This bit will be cleared after TSADC access finishing. Only when TSADC_USER_CON[5] = 1'b1, this bit takes effect.
6	RW	0x0	eoc_inten Enable eoc interrupt for each conversion. 1'b1: Enable 1'b0: Disable
5	RW	0x0	start_mode Start mode. 1'b0: TSADC controller will assert start_of_conversion after power-up. 1'b1: The start_of_conversion will be controlled by TSADC_USER_CON[7].
4	RW	0x0	power_control ADC power control 1'b0: power down 1'b1: power up This bit is not enabled when TSADC_AUTO_CON[0] is set to 1.
3:0	RO	0x0	reserved

TSADC_AUTO_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable
15:9	RO	0x00	reserved
8	RW	0x0	tshut_polarity This bit is used to control the output signal polarity to GPIO when the temperature is higher than threshold. 1'b0: Low active 1'b1: High active
7:3	RO	0x00	reserved
2	RW	0x0	round_int_en Int enable for round model. Used for auto model all channel set is sampled. 1'b1: Enable 1'b0: Disable
1	RW	0x0	q_sel 1'b0: Use tsadc_q as output (positive temperature coefficient) 1'b1: Use (q_max - tsadc_q) as output (negative temperature coefficient)
0	RW	0x0	auto_en Auto conversion enable 1'b0: TSADC controller works at user-defined mode 1'b1: TSADC controller works at auto mode

TSADC_AUTO_STATUS

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	ht_wram High-temperature status 1'b0: TSADC data is not higher than ht shut temperature 1'b1: TSADC data is higher than ht shut temperature
2	RW	0x0	auto_status TSADC auto mode status 1'b0: Auto mode stop 1'b1: Auto mode in progress
1	W0C	0x0	last_tshut_cru Status for cru reset latest tshut, write 1 to clear.
0	W1C	0x0	last_tshut_gpio Status for GPIO latest tshut, write 1 to clear.

TSADC_AUTO_SRC

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	auto_src Enable channel for TSADC in auto mode. Each bit can enable for one channel. In this chip onlu channel 0 is enable, so this bit should be set to 1'b1.

TSADC_LT_EN

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable
15:0	RW	0x0000	low_temperature_vio_en Low-temperature violation logic enable, only bit0 is used. 1'b1: Enable 1'b0: Disable

TSADC_HT_INT_EN

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable
15:0	RW	0x0000	high_temperature_int_en Temperature violation to GPIO enable, only bit0 is used. 1'b1: Enable 1'b0: Disable

TSADC_CRU_EN

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	cru_en Temperature violation to cru reset enable, only bit0 is used. 1'b1: Enable 1'b0: Disable

TSADC_LT_INT_EN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b1: Write access enable 1'b0: Write access disable
15:0	RW	0x0000	low_tempeture_int_en Low-temperature interrupt enable, only bit0 is used. 1'b1: Enable 1'b0: Disable

TSADC_HLT_INT_PD

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	W1C	0x0000	lt_int_status Low-temperature interrupt status for each channel.
15:0	W1C	0x0000	ht_int_status High-temperature interrupt status for each channel.

TSADC_EOC_HSHUT_PD

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	W1C	0x0	round_int_pd Auto model interrupt for each round of all set channel.
16	W1C	0x0	usr_eoc_irq_pd User model end interrupt status.
15:0	W1C	0x0000	ht_shut_pd High temperature shut down status for each channel.

TSADC_DATA0

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion.

TSADC_COMP0_INT

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high-temperature level. TSADC output is bigger than tsadc_comp, which means the temperature is high. TSADC_INT will be valid.

TSADC_COMP0_SHUT

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high-temperature level. TSADC output is bigger than tsadc_comp, which means the temperature is too high. TSHUT will be valid.

TSADC_HIGH_INT_DEBOUNCE

Address: Operational Base + offset (0x014C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when the temperature is higher than COMP_INT for "debounce" times.

TSADC_HIGHT_TSHUT_DEBOUNCE

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when the temperature is higher than COMP_SHUT for "debounce" times.

TSADC_AUTO_PERIOD

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period When auto mode is enabled, this register controls the interleave between every conversion for all channels enabled of TSADC.

TSADC_AUTO_PERIOD_HT

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every conversion for all channels enabled of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.

TSADC_COMP0_LOW_INT

Address: Operational Base + offset (0x015C)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC low-temperature level. TSADC output is lower than tsadc_comp, which means the temperature is low. TSADC_LOW_INT will be valid.

TSADC_T_SETUP

Address: Operational Base + offset (0x019C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x005f	t_setup The timing between TSADC power up and start conversion.

TSADC_T_PW_EN

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0000	t_pw_en The timing assert start conversion signal(EN_TEMP_SEN_TS).

TSADC_T_EN_CLK

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x01	t_en_clk The timing between dis-assert start conversion signal (EN_TEMP_SEN_TS) and assert CLK_SEN_TS[8].

TSADC_T_NON_OV

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x02	t_non_ov The timing between each bit assertion of CLK_SENSE_TS.

TSADC_T_HOLD

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0013	t_hold The timing between dis-assert CLK_SENSE_TS and TSADC data valid.

TSADC_Q_MAX

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12:0	RW	0x0200	q_max This register used for TSADC_AUTO_CON[1] for inversion output data.

TSADC_FLOW_CON

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	as_pd_mode When this bit set to 1'b1, TSADC PHY will come to low power mode between the two conversion.
3:2	RO	0x0	reserved
1	RW	0x1	pd_soc_mode Not used in this chip.
0	RW	0x1	eoc_mode Not used in this chip.

TSADC_T_PW_CLK

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0030	t_pw_clk The timing for the assertion of each bit of CLK_SENSE_TS.

11.5 Application Notes

11.5.1 Coversion Flow

The system works as follows

1) Temperature violation and other configurations may be set.
2) Assert tsadc_tsen_en and tsadc_ana_reg0/1/2, which could be found in VOGRF.
3) Set the bit[0] in register TSADC_AUTO_CON, then TS-ADC will work.
4) Then temperature information can be read from the APB interface.
5) If TS-ADC needs to be closed, set bit[0] in register TSADC_AUTO_CON to 1'b0.

11.5.2 Timing Diagram

When TS-ADC start to work, the timing will follow the Fig.1-2
The timing requirement is also listed as follows.

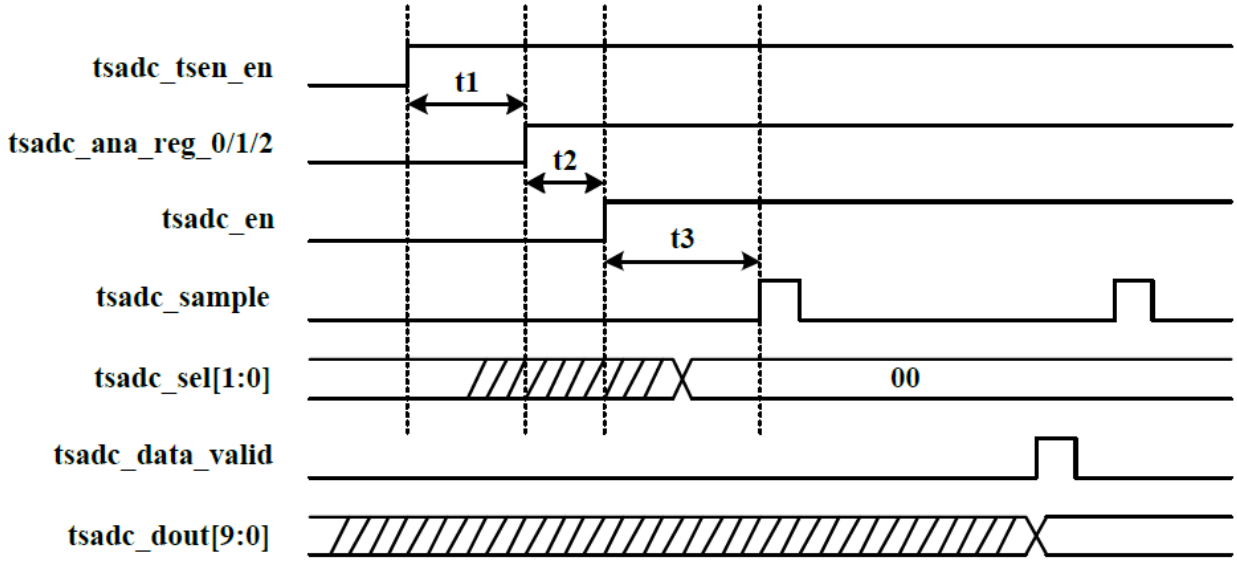


Fig. 11-2 Timing diagram for TS-ADC

Table 11-1 TS-ADC Start Timing

parameter	min
t1	10us
t2	0us
t3	90us

11.5.3 Temperature-to-Code Mapping

Table 11-2 TS-ADC Look up Table

temperature (℃)	typical output
-40	628
-35	620
-30	612
0	561
5	553
10	545
25	520
30	512
80	427
85	419
115	368
120	359

temperature (℃)	typical output
125	351

Note:
Code to Temperature mapping of the Temperature sensor is a linear curve. Any temperature, code falling between to 2 given temperatures can be linearly interpolated.
Code to Temperature mapping should be updated based on silicon results.

Chapter 12 Pulse Width Modulation (PWM)

12.1 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
- Support 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
  - Channel 3 support 32-bits power key capture mode
  - Support an input filter to remove glitch
- Support continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - PWM output polarity in inactive state and duty cycle polarity can be configured
  - 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
  - Support global lock register to control shadow register effective time
  - Support IR transmission in NEC with full repeat, NEC with simple repeat, TC9012 or SONY mode
- Pre-scaled operation to clk_pwm and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

12.2 Block Diagram

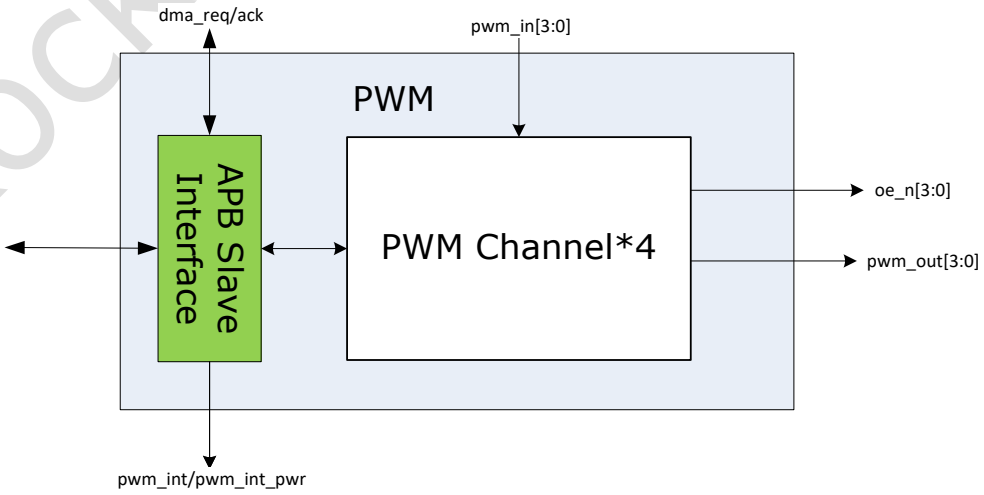


Fig. 12-1 PWM Block Diagram

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.

12.3 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.

12.3.1 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 is working 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.

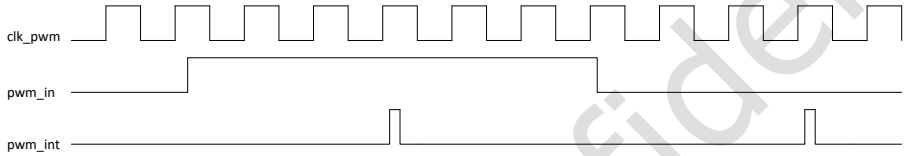


Fig. 12-2 PWM Capture Mode

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 (only channel 3) support 32-bits power key capture mode. User can configure 10 power key to match, the interrupt will be asserted when the capture value match any one.

12.3.2 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.

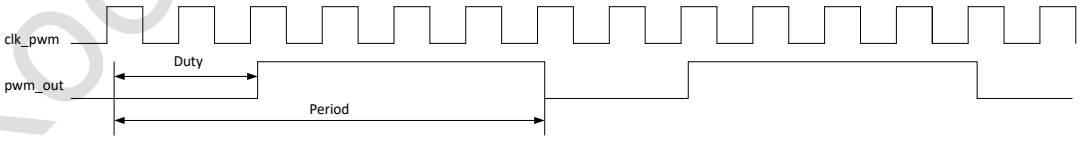


Fig. 12-3 PWM Continuous Left-aligned Output Mode

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.

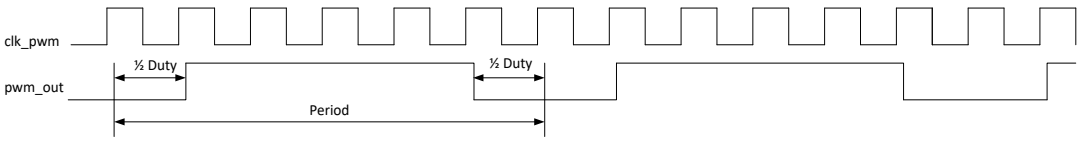


Fig. 12-4 PWM Continuous Center-aligned Output Mode

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).

12.3.3 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.

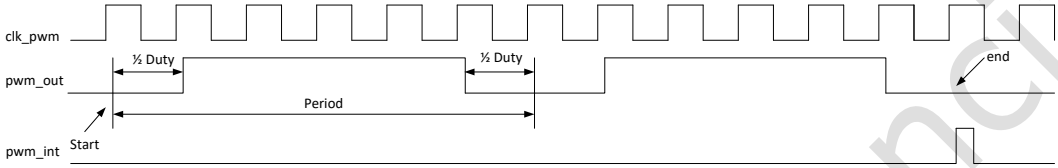


Fig. 12-5 PWM One-shot Center-aligned Output Mode

12.4 Register Description

12.4.1 Internal Address Mapping

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

12.4.2 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
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register

Name	Offset	Size	Reset Value	Description
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_VERSION_ID	0x005C	W	0x03120B34	PWM Version ID Register
PWM_FIFO	0x0060	W	0x00000000	FIFO Register
PWM_PWRMATCH_CTRL	0x0080	W	0x00000000	Power Key Match Control Register
PWM_PWRMATCH_LPRE	0x0084	W	0x238C22C4	Power Key Match Of Low Preload Register
PWM_PWRMATCH_HPRE	0x0088	W	0x11F81130	Power Key Match Of High Preload Register
PWM_PWRMATCH_LD	0x008C	W	0x029401CC	Power Key Match Of Low Data Register
PWM_PWRMATCH_HD_ZERO	0x0090	W	0x029401CC	Power Key Match Of High Data For Zero Register
PWM_PWRMATCH_HD_ONE	0x0094	W	0x06FE0636	Power Key Match Of High Data For One Register
PWM_PWRMATCH_VALUE0	0x0098	W	0x00000000	Power Key Match Value 0 Register
PWM_PWRMATCH_VALUE1	0x009C	W	0x00000000	Power Key Match Value 1 Register
PWM_PWRMATCH_VALUE2	0x00A0	W	0x00000000	Power Key Match Value 2 Register
PWM_PWRMATCH_VALUE3	0x00A4	W	0x00000000	Power Key Match Value 3 Register
PWM_PWRMATCH_VALUE4	0x00A8	W	0x00000000	Power Key Match Value 4 Register
PWM_PWRMATCH_VALUE5	0x00AC	W	0x00000000	Power Key Match Value 5 Register
PWM_PWRMATCH_VALUE6	0x00B0	W	0x00000000	Power Key Match Value 6 Register
PWM_PWRMATCH_VALUE7	0x00B4	W	0x00000000	Power Key Match Value 7 Register
PWM_PWRMATCH_VALUE8	0x00B8	W	0x00000000	Power Key Match Value 8 Register
PWM_PWRMATCH_VALUE9	0x00BC	W	0x00000000	Power Key Match Value 9 Register
PWM_PWM3_PWRCAPTURE_VALUE	0x00CC	W	0x00000000	Channel 3 Power Key Capture Value Register
PWM_FILTER_CTRL	0x00D0	W	0x00000000	Filter Control Register
PWM_PWM0_OFFSET	0x00E0	W	0x00000000	Channel0 Output Offset
PWM_PWM1_OFFSET	0x00E4	W	0x00000000	Channel1 Output Offset
PWM_PWM2_OFFSET	0x00E8	W	0x00000000	Channel2 Output Offset
PWM_PWM3_OFFSET	0x00EC	W	0x00000000	Channel3 Output Offset
PWM_IR_TRANS_CTRL	0x0100	W	0x00001F02	IR Transmission control register
PWM_IR_TRANS_PRE	0x0104	W	0x11942328	IR transmission preload period

Name	Offset	Size	Reset Value	Description
PWM_IR_TRANS_SPRE	0x0108	W	0x000008CA	IR transmission simple preload low period
PWM_IR_TRANS_LD	0x010C	W	0x00000230	IR Trans Out Data Low Period
PWM_IR_TRANS_HD	0x0110	W	0x069A0230	IR Trans Out High Period
PWM_IR_TRANS_BURST_FRAME	0x0114	W	0x2261A5E0	IR transmission frame and burst period
PWM_IR_TRANS_DATA_VALUE	0x0118	W	0x00000000	IR Trans Output Data Value

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

12.4.3 Detail Registers Description

PWM_PWM0_CNT

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT 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 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 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 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 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 This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	RW	0x0	aligned_vld_n 0: Aligned Mode Valid 1: Aligned Mode Invalid
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 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 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: 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 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol 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 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 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 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	RO	0x00000000	CNT 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 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 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 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 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 This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	RW	0x0	aligned_vld_n 0: Aligned Mode Valid 1: Aligned Mode Invalid
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 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 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: 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 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol 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 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 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 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 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 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 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 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 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
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	RW	0x0	aligned_vld_n 0: Aligned Mode Valid 1: Aligned Mode Invalid
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 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 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: 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 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol 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 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 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 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	RO	0x00000000	CNT 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 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 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 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 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 This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	RW	0x0	aligned_vld_n 0: Aligned Mode Valid 1: Aligned Mode Invalid
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 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 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: 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 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol 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 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 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 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	0x00000	reserved
11	RO	0x0	CH3_Pol 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 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 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 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	W1C	0x0	CH3_pwr_IntSts 1'b0: Channel 3 power key Interrupt not generated 1'b1: Channel 3 power key Interrupt generated
6	W1C	0x0	CH2_pwr_IntSts 1'b0: Channel 2 power key Interrupt not generated 1'b1: Channel 2 power key Interrupt generated
5	W1C	0x0	CH1_pwr_IntSts 1'b0: Channel 1 power keyInterrupt not generated 1'b1: Channel 1 power key Interrupt generated

Bit	Attr	Reset Value	Description
4	W1C	0x0	CH0_pwr_IntSts 1'b0: Channel 0 power key Interrupt not generated 1'b1: Channel 0 power key Interrupt generated
3	W1C	0x0	CH3_IntSts 1'b0: Channel 3 Interrupt not generated 1'b1: Channel 3 Interrupt generated
2	W1C	0x0	CH2_IntSts 1'b0: Channel 2 Interrupt not generated 1'b1: Channel 2 Interrupt generated
1	W1C	0x0	CH1_IntSts 1'b0: Channel 1 Interrupt not generated 1'b1: Channel 1 Interrupt generated
0	W1C	0x0	CH0_IntSts 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	0x000000	reserved
7	RW	0x0	CH3_pwr_Int_en 1'b0: Channel 3 power key Interrupt disabled 1'b1: Channel 3 power key Interrupt enabled
6	RW	0x0	CH2_pwr_Int_en 1'b0: Channel 2 power key Interrupt disabled 1'b1: Channel 2 power key Interrupt enabled
5	RW	0x0	CH1_pwr_Int_en 1'b0: Channel 1 power key Interrupt disabled 1'b1: Channel 1 power key Interrupt enabled
4	RW	0x0	CH0_pwr_Int_en 1'b0: Channel 0 power key Interrupt disabled 1'b1: Channel 0 power key Interrupt enabled
3	RW	0x0	CH3_Int_en 1'b0: Channel 3 Interrupt disabled 1'b1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en 1'b0: Channel 2 Interrupt disabled 1'b1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en 1'b0: Channel 1 Interrupt disabled 1'b1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en 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	0x00000	reserved
13:12	RW	0x0	dma_ch_sel 2'b00: Select PWM0 2'b01: Select PWM1 2'b10: Select PWM2 2'b11: Select PWM3
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	dma_ch_sel_en 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 1'b1: Enable 1'b0: Disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost full Watermark level
3	RW	0x0	watermark_int_en Watermark full interrupt
2	RW	0x0	overflow_int_en When high, an interrupt asserts when the fifo overflow.
1	RW	0x0	full_int_en When high, an interrupt asserts when the FIFO is full.
0	RW	0x0	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	0x0000000	reserved
4	RO	0x1	fifo_empty_status This bit indicates the FIFO is empty.
3	W1C	0x0	timieout_intsts Timeout interrupt
2	W1C	0x0	fifo_watermark_full_intsts This bit indicates the FIFO is Watermark Full.
1	W1C	0x0	fifo_overflow_intsts This bit indicates the FIFO is overflow.
0	W1C	0x0	fifo_full_intsts This bit indicates the FIFO is full.

PWM_FIFO_TOUTTHR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	timeout_threshold FIFO Timeout value(unit pwm clock)

PWM_VERSION_ID

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:24	RO	0x03	main_version Main version 8'h0: Base version 8'h1: Support DMA mode 8'h2: Support DMA mode and Power key mode 8'h3: Support DMA mode and Power key mode and IR transmission

Bit	Attr	Reset Value	Description
23:16	RO	0x12	minor_version Minor version
15:0	RO	0x0b34	svn_version SVN version

PWM_FIFO

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	pol This bit indicates the polarity of the lower 31-bit counter. 1'b0: Low 1'b1: High
30:0	RO	0x00000000	cycle_cnt 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	0x0000	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
14:12	RO	0x0	reserved
11	RW	0x0	CH3_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
10:8	RO	0x0	reserved
7	RW	0x0	CH3_pwrkey_polarity 1'b0: PWM in polarity is positive 1'b1: PWM in polarity is negative
6:4	RO	0x0	reserved
3	RW	0x0	CH3_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
2:0	RO	0x0	reserved

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_value0 Power key match value 0

PWM_PWRMATCH_VALUE1

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value1 Power key match value 1

PWM_PWRMATCH_VALUE2

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value2 Power key match value 2

PWM_PWRMATCH_VALUE3

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value3 Power key match value 3

PWM_PWRMATCH_VALUE4

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value4 Power key match value 4

PWM_PWRMATCH_VALUE5

Address: Operational Base + offset (0x00AC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value5 Power key match value 5

PWM_PWRMATCH_VALUE6

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value6 Power key match value 6

PWM_PWRMATCH_VALUE7

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value7 Power key match value 7

PWM_PWRMATCH_VALUE8

Address: Operational Base + offset (0x00B8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value8 Power key match value 8

PWM_PWRMATCH_VALUE9

Address: Operational Base + offset (0x00BC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value9 Power key match value 9

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
31:24	RO	0x00	reserved
23	RW	0x0	pwm3_global_lock PWM3 disable/enable lock to previous configuration 0: PWM3 disable lock to previous configuration 1: PWM3 enable lock to previous configuration
22	RW	0x0	pwm2_global_lock PWM2 disable/enable lock to previous configuration 0: PWM2 disable lock to previous configuration 1: PWM2 enable lock to previous configuration
21	RW	0x0	Pwm1_global_lock PWM1 disable/enable lock to previous configuration 0: PWM1 disable lock to previous configuration 1: PWM1 enable lock to previous configuration
20	RW	0x0	Pwm0_global_lock PWM0 disable/enable lock to previous configuration 0: PWM0 disable lock to previous configuration 1: PWM0 enable lock to previous configuration
19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RW	0x0	CH2_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled
17	RW	0x0	CH1_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled
16	RW	0x0	CH0_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled
15:13	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

PWM_PWM0_OFFSET

Address: Operational Base + offset (0x00E0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Channel Output Offset If PWM is operated at the continuous mode or one-shot mode, this value defines the offset of the output waveform. This value is based on the PWM clock. The value ranges from 0 to (period-duty). This register takes effective when output aligned mode is invalid.

PWM_PWM1_OFFSET

Address: Operational Base + offset (0x00E4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Channel Output Offset If PWM is operated at the continuous mode or one-shot mode, this value defines the offset of the output waveform. This value is based on the PWM clock. The value ranges from 0 to (period-duty). This register takes effective when output aligned mode is invalid.

PWM_PWM2_OFFSET

Address: Operational Base + offset (0x00E8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Channel Output Offset If PWM is operated at the continuous mode or one-shot mode, this value defines the offset of the output waveform. This value is based on the PWM clock. The value ranges from 0 to (period-duty). This register takes effective when output aligned mode is invalid.

PWM_PWM3_OFFSET

Address: Operational Base + offset (0x00EC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Channel Output Offset If PWM is operated at the continuous mode or one-shot mode, this value defines the offset of the output waveform. This value is based on the PWM clock. The value ranges from 0 to (period-duty). This register takes effective when output aligned mode is invalid.

PWM_IR_TRANS_CTRL

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:13	RO	0x00000	reserved
12:8	RW	0x1f	PWM IR Trans length within one frame 0x00~0x1F represet data lenth 1~32
7:4	RW	0x0	PWM IR Trans format PWM IR Trans format 0:NEC with simple repeat code 1:NEC with full repeat code 2:TC9012 3:SONY 0x4~0xf:reserved
3:2	RO	0x0	reserved
1	RO	0x1	PWM IR State Idle 0:PWM IR Trans Busy 1:PWM IR Trans Idle
0	RW	0x0	PWM IR Trans Out enable 0:disable PWM IR Trans Out Function 1:enable PWM IR Trans Out Function

PWM_IR_TRANS_PRE

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x1194	PWM IR Trans Out High Preload IR Trans Out High Preload Register
15:0	RW	0x2328	PWM IR Trans Out Low Preload IR Trans Out Low Preload Register

PWM_IR_TRANS_SPRE

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x08ca	PWM IR Trans Out Low Simple Preload PWM IR Trans Out Low Simple Preload Register

PWM_IR_TRANS_LD

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0230	PWM IR Trans Out Data Low Period IR Trans Out Data Low Period Register

PWM_IR_TRANS_HD

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RW	0x069a	IR Trans Out High Period For One IR Trans Out High Period For One Register
15:0	RW	0x0230	IR Trans Out High Period For Zero IR Trans Out High Period For Zero Register

PWM_IR_TRANS_BURST_FRAME

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RW	0x226	PWM IR Trans Out Burst Period PWM IR Trans Out Burst Period Register
19:18	RO	0x0	reserved
17:0	RW	0x1a5e0	PWM IR Trans Out Frame Period Register PWM IR Trans Out Frame Period

PWM_IR_TRANS_DATA_VALUE

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PWM IR Trans Out Value PWM IR Trans Out Value Register

12.5 Interface Description

Table 12-1 PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
PWM0m0	I/O	VICAP_D0_M1/AVS_ARM/PWM0_M0/GPIO1_A2_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[10:8]=3'h1
PWM0m1	I/O	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[10:8]=3'h6
PWM1m0	I/O	PWM1_M0/PMIC_SLEEP_M0/GPIO0_A4_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[2:0]=3'h2
PWM1m1	I/O	PWM1_M1/ADCIN1/GPIO4_C1_z	VCCIO2_IOC_GPIO4C_IOMUX_S

Module Pin	Direction	Pad Name	IOMUX Setting
			EL_L[6:4]=3'h2
PWM1m2	I/O	PWM1_M2/VICAP_D15/GPIO3_D3_d	VCCIO7_IOC_GPIO3D_IOMUX_SEL_L[14:12]=3'h2
PWM2m0	I/O	TEST_CLK6_OUT/PWM2_M0/UART0_TX_M0/GPIO0_A1_d	PMU_IOC_GPIO0A_IOMUX_SEL_L[6:4]=3'h2
PWM2m1	I/O	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[10:8]=3'h4
PWM2m2	I/O	SDMMC1_D1_M1/SPI0_CS0n_M0/PWM2_M2/VICAP_D2_M1/LCD_D7/GPIO1_C0_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[2:0]=3'h3
PWM3m0	I/O	TEST_CLK7_OUT//PWM3_IR_M0/GPIO0_A2_d	PMU_IOC_GPIO0A_IOMUX_SEL_L[10:8]=3'h1
PWM3m1	I/O	PWM3_IR_M1/UART4_RX_M0/GPIO1_B0_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[2:0]=3'h2
PWM3m2	I/O	UART3_TX_M1/UART5_RTSN_M1/PWM3_IR_M2/VICAP_CLKIN_M1/LCD_DEN/GPIO1_D0_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[2:0]=3'h3
PWM4m0	I/O	PWM4_M0/PMU_DEBUG/I2C2_SDA_M0/UART3_RX_M0/GPIO1_A1_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[6:4]=3'h4
PWM4m1	I/O	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[14:12]=3'h4
PWM4m2	I/O	SDMMC1_D0_M1/SPI0_CLK_M0/PWM4_M2/VICAP_D3_M1/LCD_D6/GPIO1_C1_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[6:4]=3'h3
PWM5m0	I/O	PWM5_M0/UART1_RTSN_M0/I2C1_SCL_M0/GPIO0_A5_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[6:4]=3'h3
PWM5m1	I/O	TEST_CLK4_OUT/PWM5_M1/LCD_D16/I2C1_SDA_M1/UART0_RX_M1/GPIO2_B0_d	VCCIO5_IOC_GPIO2B_IOMUX_SEL_L[2:0]=3'h4
PWM5m2	I/O	SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VCCIO6_IOC_GP

Module Pin	Direction	Pad Name	IOMUX Setting
			IO1C_IOMUX_SEL_L[10:8]=3'h3
PWM6m0	I/O	PWM6_M0/UART1_CTSN_M0/I2C1_SDA_M0/GPIO0_A6_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[10:8]=3'h3
PWM6m1	I/O	TEST_CLK5_OUT/PWM6_M1/LCD_D17/I2C1_SCL_M1/UART0_TX_M1/GPIO2_B1_d	VCCIO5_IOC_GPIO2B_IOMUX_SEL_L[6:4]=3'h4
PWM6m2	I/O	SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[14:12]=3'h3
PWM7m0	I/O	PWM7_IR_M0/I2C2_SCL_M0/UART3_TX_M0/GPIO1_A0_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[2:0]=3'h3
PWM7m1	I/O	VICAP_D1_M1/SPI1_CS1n_M0/PWM7_IR_M1/UART4_TX_M0/GPIO1_B1_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[6:4]=3'h2
PWM7m2	I/O	MIPI_REFCLK_OUT1/PWM7_IR_M2/VICAP_D10/GPIO3_C6_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_H[10:8]=3'h2
PWM8m0	I/O	PWM8_M0/TEST_CLK1_OUT/UART2_RX_M0/SDMMC0_D0/GPIO3_A3_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[14:12]=3'h4
PWM8m1	I/O	SDMMC1_D3_M1/UART4_RX_M1/PWM8_M1/VICAP_D6_M1/LCD_D3/GPIO1_C4_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[2:0]=3'h3
PWM9m0	I/O	PWM9_M0/TEST_CLK0_OUT/UART2_TX_M0/SDMMC0_D1/GPIO3_A2_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[10:8]=3'h4
PWM9m1	I/O	SDMMC1_D2_M1/UART4_TX_M1/PWM9_M1/VICAP_D7_M1/LCD_D2/GPIO1_C5_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[6:4]=3'h3
PWM10m0	I/O	PWM10_M0/LPMCU_JTAG_TCK_M1/I2C0_SCL_M2/UART5_RTSN_M0/SDMMC0_CLK/GPIO3_A4_d	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[2:0]=3'h5
PWM10m1	I/O	UART4_RTSN_M1/PWM10_M1/VICAP_D8_M1/LCD_D1/GPIO1_C6_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[10:8]=3'h3

Module Pin	Direction	Pad Name	IOMUX Setting
PWM10_m2	I/O	UART3_RX_M1/UART5_CTSN_M1/PWM10_M2/VICAP_HSYNC_M1/LCD_HSYNC/GPIO1_D1_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[6:4]=3'h3
PWM11_m0	I/O	PWM11_IR_M0/LPMCU_JTAG_TMS_M1/I2C0_SDA_M2/UART5_CTSN_M0/SDMMC0_CMD/GPIO3_A5_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[6:4]=3'h5
PWM11_m1	I/O	UART4_CTSN_M1/PWM11_IR_M1/VICAP_D9_M1/LCD_D0/GPIO1_C7_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[14:12]=3'h3
PWM11_m2	I/O	DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[14:12]=3'h5

Notes: I=input, O=output, I/O=input/output, bidirectional

12.6 Application Notes

12.6.1 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.

12.6.2 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_en and 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 transfer 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.

12.6.3 PWM Power key Capture Mode Standard Usage Flow

1. Set PWM_PWM3_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_PWM3_CTRL.prescale and PWM_PWM3_CTRL.scale, and select the clock needed by setting PWM_PWM3_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.CH3_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.CH3_pwrkey_polarity for the polarity of power key signal, the default value is 0. Enable the PWM_PWRMATCH_CTRL.CH3_pwrkey_enable.
8. Set PWM_FILTER_CTRL. filter_number, then Enable the PWM_FILTER_CTRL. CH3_input_filter_enable(Optional).
9. Enable the channel by writing '1' to PWM_PWM3_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_PWM3_PWRCAPTURE_VALUE to know the power key capture value.
11. Write '0' to PWM_PWM3_CTRL.pwm_en to disable the channel.

12.6.4 PWM One-shot Mode/Continuous 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 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.

12.6.5 Low-power Usage Flow

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.

12.6.6 Multi-channel synchronous start Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Set PWM_FILTER_CTRL.PWMx_global_lock '1' to lock previous configuration.
3. 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.
4. 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.
5. Set the PWM_PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode.
6. Configure the channel to work in the one-shot mode or the continuous mode.
7. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation if the channel is desired to work in the one-shot mode.
8. Set PWM_PWMx_CTRL.pwm_en to '1' to start pwm.
9. Set PWM_FILTER_CTRL.PWMx_global_lock '0' to unlock previous configuration and the

latest configuration will be take effective immediately.
10. 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.

12.6.7 Multi-channel synchronous stop or frequency change Usage Flow

1. Set PWM_FILTER_CTRL.PWMx_global_lock '1' to lock previous configuration.
2. Set PWM_PWMx_CTRL.pwm_en to '0' to stop pwm or set PWM_PWMx_PERIOD_HPC and PWM_PWMx_DUTY_LPC to change frequency.
3. Set PWM_FILTER_CTRL.PWMx_global_lock '0' to unlock previous configuration and the latest configuration will be take effective immediately.

12.6.8 IR transmission Usage Flow

1. Set PWM_PWM2_CTRL.pwm_en to '0' to disable the PWM channel.
2. Set PWM_IR_TRANS_CTRL.transout_en to '1' enable PWM IR Trans Out Function.
3. Set PWM_IR_TRANS_CTRL.format, PWM_IR_TRANS_CTRL.length, PWM_IR_TRANS_PRE, PWM_IR_TRANS_DATA_VALUE .et.
4. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWM2_CTRL.prescale and PWM_PWM2_CTRL.scale, and select the clock needed by setting PWM_PWM2_CTRL.clk_sel. The clock should be 1 MHz after division.
5. Choose the output mode by setting PWM_PWM2_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWM_PWM2_CTRL.duty_pol and PWM_PWM2_CTRL.inactive_pol.
6. Set the PWM_PWM2_CTRL.rpt if the channel is desired to work in the one-shot mode.
7. Enable the channel by writing '1' to PWM_PWM3_CTRL.pwm_en bit to start the channel.
8. If you want to disable the output waveform, write '0' PWM_PWM2_CTRL.pwm_en, and the channel will be disable at the end of one frame. Please refer to PWM_IR_TRANS_CTRL.idle to see if the channel is true stop.

12.6.9 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', others 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 13 UART

13.1 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
- All contain two 64 Bytes FIFOs for data receive and transmit
- All 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

13.2 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

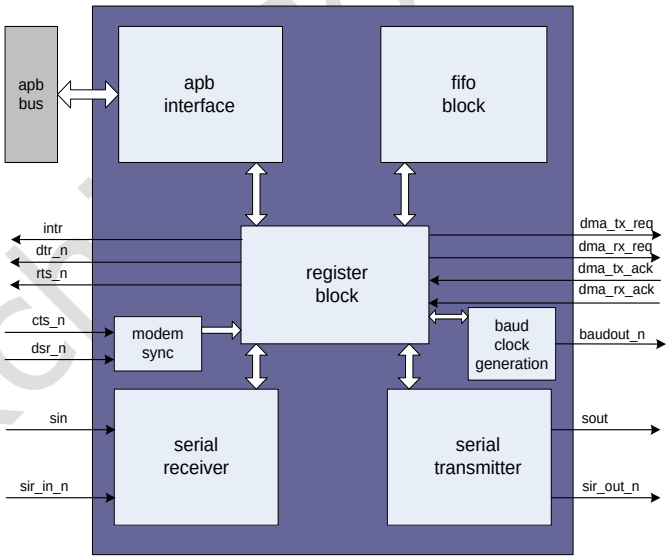


Fig. 13-1 UART Architecture

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 clock along 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.

Serial Receiver
Converts the serial data character (as specified by the control register) received to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

13.3 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.

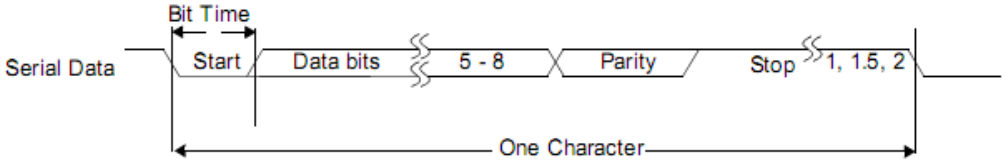


Fig. 13-2 UART Serial protocol

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.

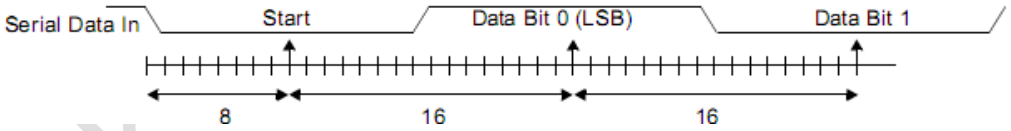


Fig. 13-3 UART baud rate

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 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.

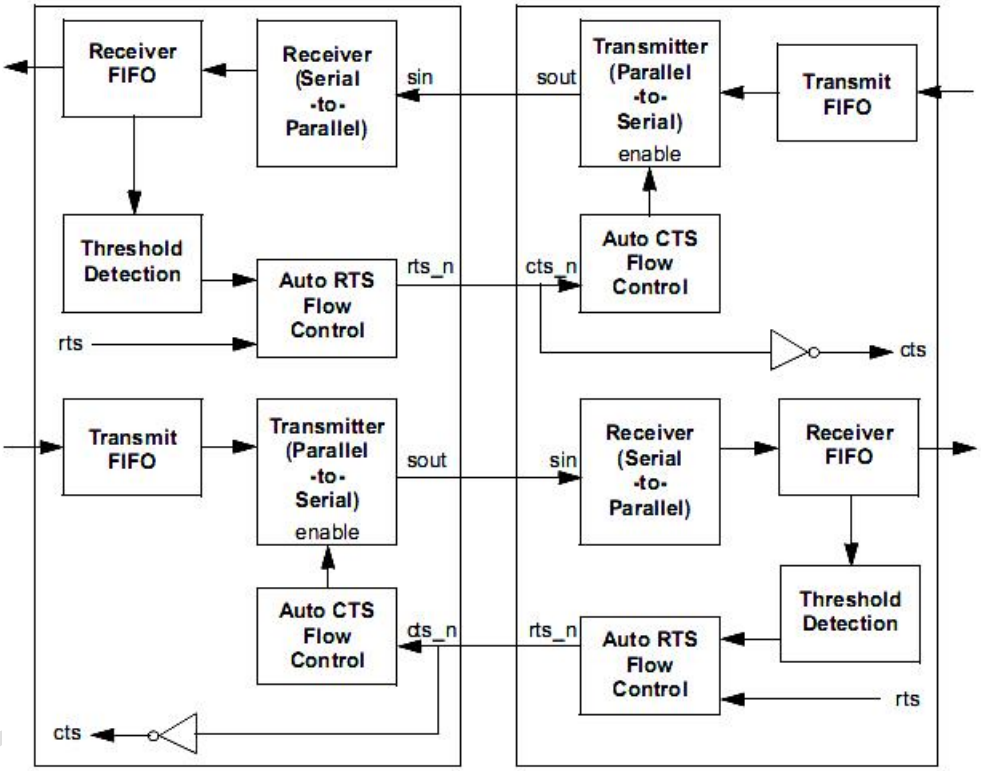


Fig. 13-4 UART Auto flow control block diagram

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)

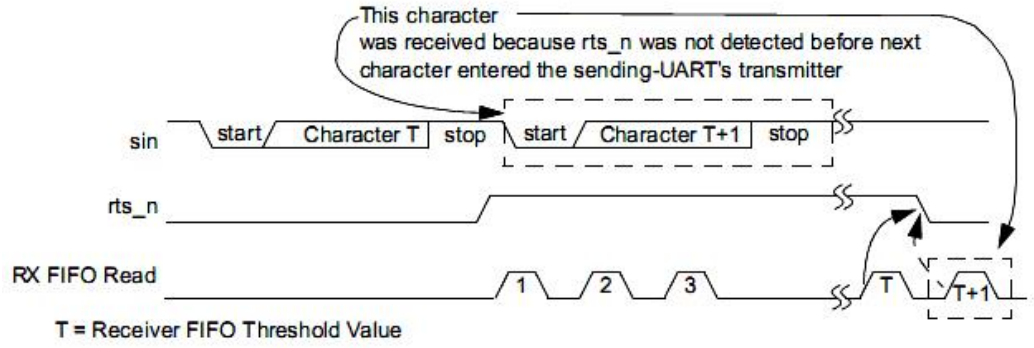


Fig. 13-5 UART AUTO RTS TIMING

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)

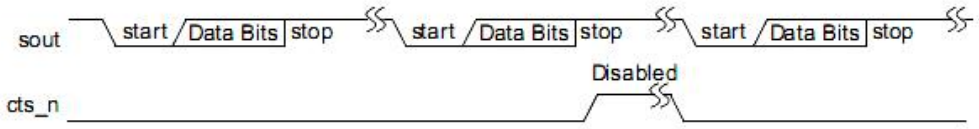


Fig. 13-6 UART AUTO CTS TIMING

13.4 Register Description

13.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch Low
UART_THR	0x0000	W	0x00000000	Transmit Buffer Register
UART_DLH	0x0004	W	0x00000000	Divisor Latch High
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_FCR	0x0008	W	0x00000000	FIFO Enable
UART_IIR	0x0008	W	0x00000001	Interrupt Identity 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	0x0030	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

Name	Offset	Size	Reset Value	Description
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	0x00043FF2	Component Parameter Register
UART_UCV	0x00F8	W	0x3430322A	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, **DW**-Double WORD (64 bits) access*

13.4.2 Detail Registers Description

UART_RBR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	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 will lost and an over-run error occurs.

UART_DLL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
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 (pclk if one clock design, sclk if two clock design (CLOCK_MODE == Enabled)) 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 disable and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

UART_THR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	WO	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, 64 characters of data may be written to the THR before the FIFO is full.. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLH

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	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. 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 (pclk if one clock design, sclk if two clock design CLOCK_MODE == Enabled)) 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 disable 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_IER

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. 1'b0: Disabled 1'b1: 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. 1'b0: Disabled 1'b1: 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. 1'b0: Disabled 1'b1: Enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 1'b0: Disabled 1'b1: Enabled
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. 1'b0: Disabled 1'b1: Enabled

UART_FCR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	WO	0x0	rcvr_trigger 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. For details on DMA support, refer to "DMA Support". The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. Writes have no effect when THRE_MODE_USER == Disabled. 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. For details on DMA support, refer to " DMA Support". The following trigger levels are supported: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dma_mode DMA Mode. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == No). For details on DMA support, refer to DMA Support. 1'b0: Mode 0 1'b1: Mode 1
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 (DMA_EXTRA == YES). 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 (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
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_IIR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 2'b00: Disabled 2'b11: 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: 4'b0000: Modem status 4'b0001: No interrupt pending 4'b0010: THR empty 4'b0100: Received data available 4'b0110: Receiver line status 4'b0111: Busy detect 4'b1100: Character timeout The interrupt priorities are split into four levels that are detailed in Table X. Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

UART_LCR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	div_lat_access Divisor Latch Access Bit.Writable 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 SIR_MODE == Enabled and active (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	RW	0x0	stick_parity If UART_16550_COMPATIBLE = NO, then writable only when UART is not busy (USR[0] is 0); otherwise always writable and always readable. This bit is used to force parity value. When PEN, EPS and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled.
4	RW	0x0	even_parity_sel Even Parity Select. Writable 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. Writable 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. 1'b0: Parity disabled 1'b1: Parity enabled
2	RW	0x0	stop_bits_num Number of stop bits. Writable 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, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	data_length_sel Data Length Select. Writable 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: 2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits

UART_MCR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	sir_mode_en SIR Mode Enable. Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/disable the IrDA SIR Mode features as described in "IrDA 1.0 SIR Protocol". 1'b0: IrDA SIR Mode disabled 1'b1: IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in "Auto Flow Control". 1'b0: Auto Flow Control Mode disabled 1'b1: 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. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
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: 1'b0: Out2_n de-asserted (logic 1) 1'b1: Out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.

Bit	Attr	Reset Value	Description
2	RW	0x0	out1 OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 1'b0: Out1_n de-asserted (logic 1) 1'b1: Out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.
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. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
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: 1'b0: dtr_n de-asserted (logic 1) 1'b1: dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive

UART_LSR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND 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. 1'b0: No error in RX FIFO 1'b1: Error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
6	RO	0x1	trans_empty Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or 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_USER == Disabled or 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 THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (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. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.
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. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 1'b0: No framing error 1'b1: Framing error Reading the LSR clears the FE bit.
2	RO	0x0	parity_error 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. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 1'b0: No parity error 1'b1: Parity error Reading the LSR clears the PE bit.

Bit	Attr	Reset Value	Description
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. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 1'b0: No overrun error 1'b1: Overrun error Reading the LSR clears the OE bit.
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. 1'b0: No data ready 1'b1: Data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

UART_MSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	data_carrior_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 1'b0: dcd_n input is de-asserted (logic 1) 1'b1: dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2).
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 1'b0: ri_n input is de-asserted (logic 1) 1'b1: ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1).
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the UART. 1'b0: dsr_n input is de-asserted (logic 1) 1'b1: dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR).

Bit	Attr	Reset Value	Description
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the UART. 1'b0: cts_n input is de-asserted (logic 1) 1'b1: cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
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. 1'b0: No change on dcd_n since last read of MSR 1'b1: Change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
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'b0: No change on ri_n since last read of MSR 1'b1: Change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.
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. 1'b0: No change on dsr_n since last read of MSR 1'b1: Change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.
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. 1'b0: No change on ctsdsr_n since last read of MSR 1'b1: Change on ctsdsr_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

UART_SCR

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	temp_store_space Scratchpad register.This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

UART_SRBR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations (0x30-0x6c) 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 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 overrun 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 are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	WO	0x00	shadow_thr This is a shadow register for the THR and has been allocated sixteen 32-bit locations(0x30-0x6c) so as to accommodate burst accesses from the master. This register contains 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] set to zero) 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] set to one) 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_FAR

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	fifo_access_test_en Writes have no effect when FIFO_ACCESS == No, always readable. This register is used 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 implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 1'b0: FIFO access mode disabled 1'b1: FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty.

UART_TFR

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
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. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

UART_RFW

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	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). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.
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). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.
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 implemented and 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 are not implemented or 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	0x0000000	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 1'b0: Receive FIFO not full 1'b1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 1'b0: Receive FIFO is empty 1'b1: 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 bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 1'b0: Transmit FIFO is not empty 1'b1: 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 bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 1'b0: Transmit FIFO is full 1'b1: 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 DW_apb_uart is idle or inactive. 1'b0: UART is idle or inactive 1'b1: UART is busy (actively transferring data) Note that it is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the UART has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the UART. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock.

UART_TFL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RO	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
31:6	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
5:0	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	0x00000000	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. 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 (DMA_EXTRA == YES). 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 is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO 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 (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
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	0x00000000	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. 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. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input.

UART_SBCR

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	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. 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 SIR_MODE == Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. 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 (DMA_EXTRA == NO). 1'b0: Mode 0 1'b1: Mode 1

UART_SFE

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset.

UART_SRT

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store thepreviously written value to the FCR in memory and having to mask thisvalue so that only the RCVR trigger bit gets updated.This is used to select the trigger level in the receiver FIFO at which theReceived Data Available Interrupt is generated. It also determines when thedma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full

UART_STET

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	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]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated.This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full

UART_HTX

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	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. 1'b0: Halt TX disabled 1'b1: Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

UART_DMASA

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	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. For example, if the DMA disables the channel, then the UART should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

UART_CPR

Address: Operational Base + offset (0x00F4)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	0x04	fifo_mode 8'h4 means FIFO mode is 64.
15:14	RO	0x0	reserved
13	RO	0x1	dma_extra 1'b1 means DMA_EXTRA enabled.
12	RO	0x1	uart_add_encoded_params 1'b1 means UART_ADD_ENCODED_PARAMS enabled.
11	RO	0x1	shadow 1'b1 means SHADOW mode enabled.
10	RO	0x1	fifo_stat 1'b1 means FIFO_STAT enabled.
9	RO	0x1	fifo_access 1'b1 means FIFO ACCESS enabled.
8	RO	0x1	new_feat 1'b1 means Additional features enabled.
7	RO	0x1	sir_lp_mode 1'b1 means SIR_LP mode enabled.
6	RO	0x1	sir_mode 1'b1 means SIR mode enabled.
5	RO	0x1	thre_mode 1'b1 means THRE mode enabled.
4	RO	0x1	afce_mode 1'b1 means AFCE mode enabled.
3:2	RO	0x0	reserved
1:0	RO	0x2	apb_data_width 2'b10 means APB data width is 32bit.

UART_UCV

Address: Operational Base + offset (0x00F8)

Bit	Attr	Reset Value	Description
31:0	RO	0x3430322a	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.

13.5 Interface Description

Table 13-1 UART0 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	RTC_CLKO/REF_CLK_OUT/CLKIO_32K/UART0_RX_M0/GPIO0_A0_z	PMU_IOC_GPIO0A_IOMUX_SEL_L[2:0]=3'h1
uart_tx	O	TEST_CLK6_OUT/PWM2_M0/UART0_TX_M0/GPIO0_A1_d	PMU_IOC_GPIO0A_IOMUX_SEL_L[6:4]=3'h1
IOMUX1			

uart_rx	I	TEST_CLK4_OUT/PWM5_M1/LCD_D16/I2C1_SDA_M1/UART0_RX_M1/GPIO2_B0_d	VO_IOC_GPIO2B_IOMUX_SEL_L[2:0]=3'h1
uart_tx	O	TEST_CLK5_OUT/PWM6_M1/LCD_D17/I2C1_SCL_M1/UART0_TX_M1/GPIO2_B1_d	VO_IOC_GPIO2B_IOMUX_SEL_L[6:4]=3'h1
uart_cts_n	I	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VO_IOC_GPIO2A_IOMUX_SEL_H[14:12]=3'h1
uart_rts_n	O	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VO_IOC_GPIO2A_IOMUX_SEL_H[10:8]=3'h1
IOMUX2			
uart_rx	I	TEST_CLK2_OUT/I2C0_SDA_M1/UART0_RX_M2/SPI1_MISO_M0/EMMC_D7/GPIO4_A0_u	PERI_IOC_GPIO4A_IOMUX_SEL_L[2:0]=3'h3
uart_tx	O	TEST_CLK3_OUT/I2C0_SCL_M1/UART0_TX_M2/SPI1_MOSI_M0/EMMC_D6/GPIO4_A1_u	PERI_IOC_GPIO4A_IOMUX_SEL_L[6:4]=3'h3

Table 13-2 UART1 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	I2C0_SDA_M0/UART1_RX_M0/GPIO1_A4_d	VENC_IOC_GPIO1A_IOMUX_SEL_H[2:0]=3'h1
uart_tx	O	I2C0_SCL_M0/UART1_TX_M0/GPIO1_A3_d	VENC_IOC_GPIO1A_IOMUX_SEL_L[14:12]=3'h1
uart_cts_n	I	PWM6_M0/UART1_CTSN_M0/I2C1_SDA_M0/GPIO0_A6_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[10:8]=3'h2
uart_rts_n	O	PWM5_M0/UART1_RTSN_M0/I2C1_SCL_M0/GPIO0_A5_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[6:4]=3'h2
IOMUX1			
uart_rx	I	UART1_RX_M1/LCD_D13/I2S0_SDI0/SDMMC1_D2_M0/GPIO2_A5_d	VO_IOC_GPIO2A_IOMUX_SEL_H[6:4]=3'h4
uart_tx	O	UART1_TX_M1/LCD_D12/I2S0_SDO0/SDMMC1_D3_M0/GPIO2_A4_d	VO_IOC_GPIO2A_IOMUX_SEL_H[2:0]=3'h4
uart_cts_n	I	I2C4_SDA_M0/UART1_CTSN_M1/LCD_D8/I2S0_SCLK/SDMMC1_D1_M0/GPIO2_A0_d	VO_IOC_GPIO2A_IOMUX_SEL_L[2:0]=3'h4
uart_rts_n	O	I2C4_SCL_M0/UART1_RTSN_M1/LCD_D9/I2S0_LRCK/SDMMC1_D0_M0/GPIO2_A1_d	VO_IOC_GPIO2A_IOMUX_SEL_L[6:4]=3'h4
IOMUX2			
uart_rx	I	I2C2_SCL_M1/UART1_RX_M2/SPI1_CLK_M0/EMMC_D5/GPIO4_A7_u	PERI_IOC_GPIO4A_IOMUX_SEL_H[14:12]=3'h3
uart_tx	O	I2C2_SDA_M1/UART1_TX_M2/SPI1_CS0n_M0/EMMC_D4/GPIO4_A5_u	PERI_IOC_GPIO4A_IOMUX_SEL_H[6:4]=3'h3

Table 13-3 UART2 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	PWM8_M0/TEST_CLK1_OUT/UART2_RX_M0/SDMMC0_D0/GPIO3_A3_u	VI_IOC_GPIO3A_IOMUX_SEL_L[14:12]=3'h2
uart_tx	O	PWM9_M0/TEST_CLK0_OUT/UART2_TX_M0/SDMMC0_D1/GPIO3_A2_u	VI_IOC_GPIO3A_IOMUX_SEL_L[10:8]=3'h2

IOMUX1			
uart_rx	I	LPMCU_JTAG_TMS_M0/HPMCU_JTAG_TMS_M0/UART2_RX_M1/A7_JTAG_TMS_M1/GPIO1_B3_u	VENC_IOC_GPIO1B_IOMUX_SEL_L[14:12]=3'h2
uart_tx	O	LPMCU_JTAG_TCK_M0/HPMCU_JTAG_TCK_M0/UART2_TX_M1/A7_JTAG_TCK_M1/GPIO1_B2_d	VENC_IOC_GPIO1B_IOMUX_SEL_L[10:8]=3'h2

Table 13-4 UART3 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	PWM4_M0/PMU_DEBUG/I2C2_SDA_M0/UART3_RX_M0/GPIO1_A1_d	VENC_IOC_GPIO1A_IOMUX_SEL_L[6:4]=3'h1
uart_tx	O	PWM7_IR_M0/I2C2_SCL_M0/UART3_TX_M0/GPIO1_A0_d	VENC_IOC_GPIO1A_IOMUX_SEL_L[2:0]=3'h1
IOMUX1			
uart_rx	I	UART3_RX_M1/UART5_CTSN_M1/PWM10_M2/VICAP_HSYNC_M1/LCD_HSYNC/GPIO1_D1_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[6:4]=3'h5
uart_tx	O	UART3_TX_M1/UART5_RTSN_M1/PWM3_IR_M2/VICAP_CLKIN_M1/LCD_DEN/GPIO1_D0_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[2:0]=3'h5

Table 13-5 UART4 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	PWM3_IR_M1/UART4_RX_M0/GPIO1_B0_d	VENC_IOC_GPIO1B_IOMUX_SEL_L[2:0]=3'h1
uart_tx	O	VICAP_D1_M1/SPI1_CS1n_M0/PWM7_IR_M1/UART4_TX_M0/GPIO1_B1_d	VENC_IOC_GPIO1B_IOMUX_SEL_L[6:4]=3'h1
IOMUX1			
uart_rx	I	SDMMC1_D3_M1/UART4_RX_M1/PWM8_M1/VICAP_D6_M1/LCD_D3/GPIO1_C4_d	VENC_IOC_GPIO1C_IOMUX_SEL_H[2:0]=3'h4
uart_tx	O	SDMMC1_D2_M1/UART4_TX_M1/PWM9_M1/VICAP_D7_M1/LCD_D2/GPIO1_C5_d	VENC_IOC_GPIO1C_IOMUX_SEL_H[6:4]=3'h4
uart_cts_n	I	UART4_CTSN_M1/PWM11_IR_M1/VICAP_D9_M1/LCD_D0/GPIO1_C7_d	VENC_IOC_GPIO1C_IOMUX_SEL_H[14:12]=3'h4
uart_rts_n	O	UART4_RTSN_M1/PWM10_M1/VICAP_D8_M1/LCD_D1/GPIO1_C6_d	VENC_IOC_GPIO1C_IOMUX_SEL_H[10:8]=3'h4

Table 13-6 UART5 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
uart_rx	I	HPMCU_JTAG_TCK_M1/A7_JTAG_TCK_M0/UART5_RX_M0/SDMMC0_D2/GPIO3_A7_u	VI_IOC_GPIO3A_IOMUX_SEL_H[14:12]=3'h2
uart_tx	O	HPMCU_JTAG_TMS_M1/A7_JTAG_TMS_M0/UART5_TX_M0/SDMMC0_D3/GPIO3_A6_u	VI_IOC_GPIO3A_IOMUX_SEL_H[10:8]=3'h2
uart_cts_n	I	PWM11_IR_M0/LPMCU_JTAG_TMS_M1/I2C0	VI_IOC_GPIO3A_IOMU

		_SDA_M2/UART5_CTSN_M0/SDMMC0_CMD/GPIO3_A5_u	X_SEL_H[6:4]=3'h2
uart_rts_n	O	PWM10_M0/LPMCU_JTAG_TCK_M1/I2C0_SCL_M2/UART5_RTSN_M0/SDMMC0_CLK/GPIO3_A4_d	VI_IOC_GPIO3A_IOMUX_SEL_H[2:0]=3'h2
IOMUX1			
uart_rx	I	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[10:8]=3'h4
uart_tx	O	DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[14:12]=3'h4
uart_cts_n	I	UART3_RX_M1/UART5_CTSN_M1/PWM10_M2/VICAP_HSYNC_M1/LCD_HSYNC/GPIO1_D1_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[6:4]=3'h4
uart_rts_n	O	UART3_TX_M1/UART5_RTSN_M1/PWM3_IR_M2/VICAP_CLKIN_M1/LCD_DEN/GPIO1_D0_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[2:0]=3'h4
IOMUX2			
uart_rx	I	I2C4_SDA_M2/UART5_RX_M2/VICAP_D12/GPIO3_D0_d	VI_IOC_GPIO3D_IOMUX_SEL_L[2:0]=3'h2
uart_tx	O	I2C4_SCL_M2/UART5_TX_M2/VICAP_D11/GPIO3_C7_d	VI_IOC_GPIO3C_IOMUX_SEL_H[14:12]=3'h2
uart_cts_n	I	I2C3_SDA_M2/UART5_CTSN_M2/VICAP_D14/GPIO3_D2_d	VI_IOC_GPIO3D_IOMUX_SEL_L[10:8]=3'h2
uart_rts_n	O	I2C3_SCL_M2/UART5_RTSN_M2/VICAP_D13/GPIO3_D1_d	VI_IOC_GPIO3D_IOMUX_SEL_L[6:4]=3'h2

13.6 Application Notes

13.6.1 No FIFO Mode Transfer Flow

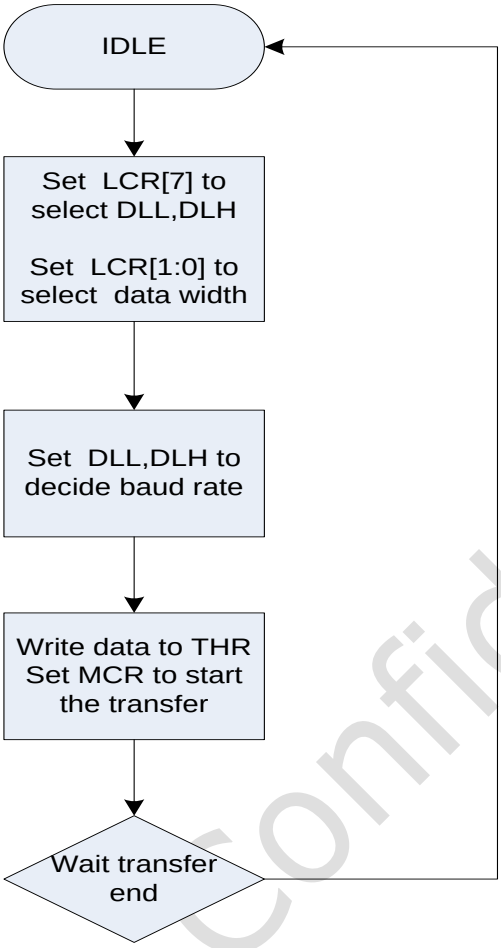


Fig. 13-7 UART none fifo mode

13.6.2 FIFO Mode Transfer Flow

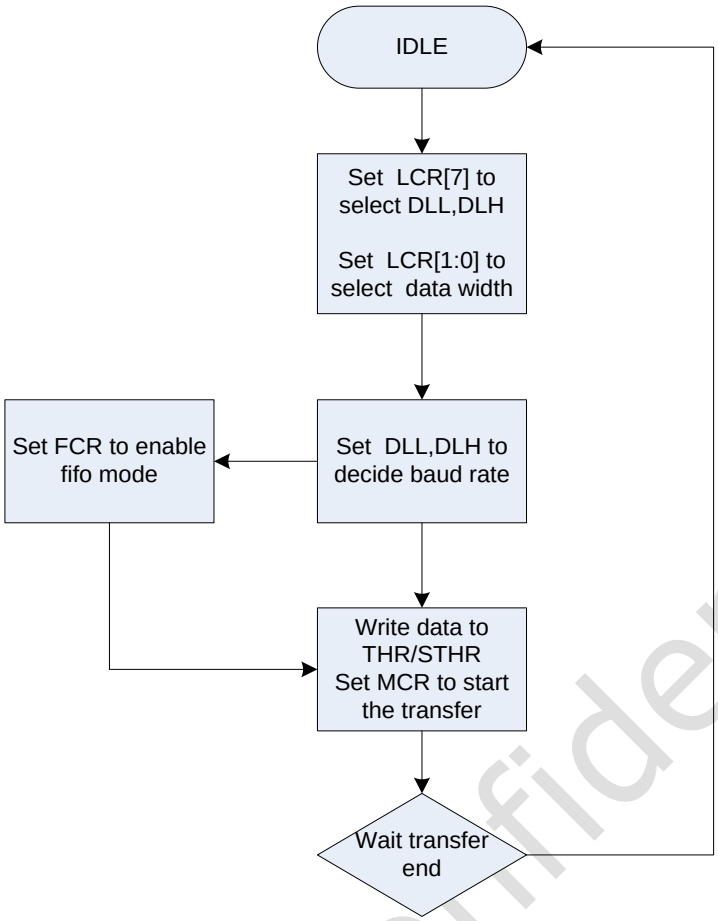


Fig. 13-8 UART fifo mode

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.

13.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation. UARTs source clocks can be selected from different PLL outputs. UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again, or be provided by XIN24M.

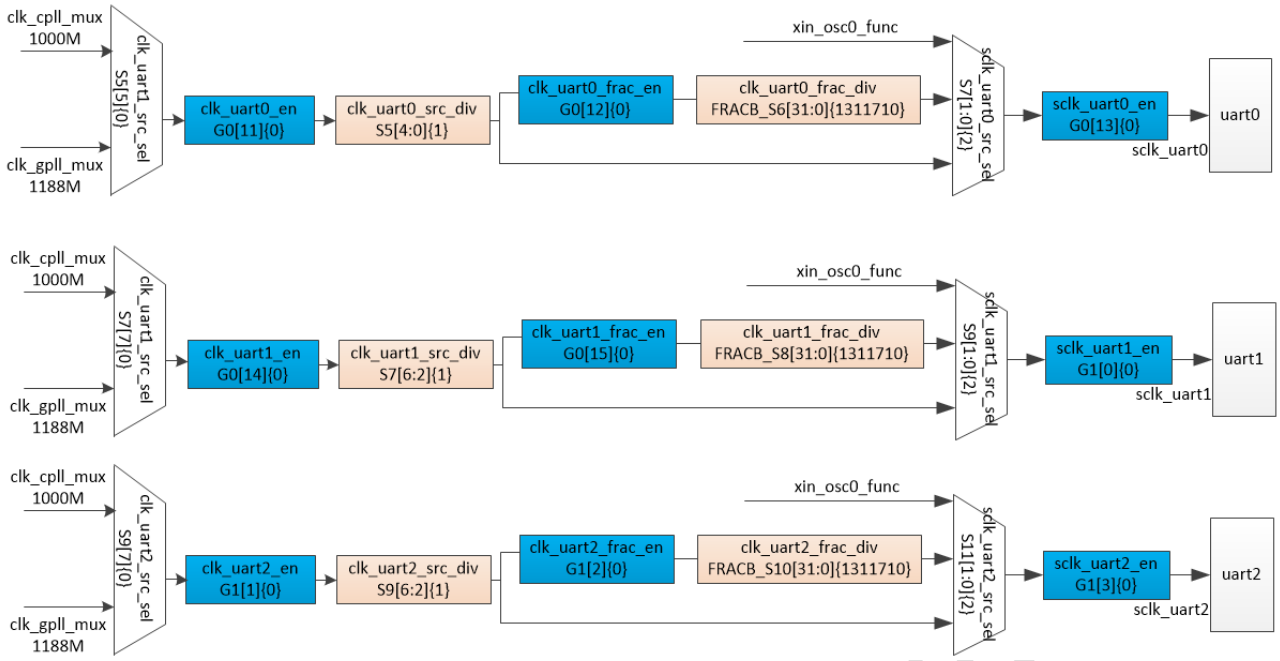


Fig. 13-9 UART0-UART2 clock generation

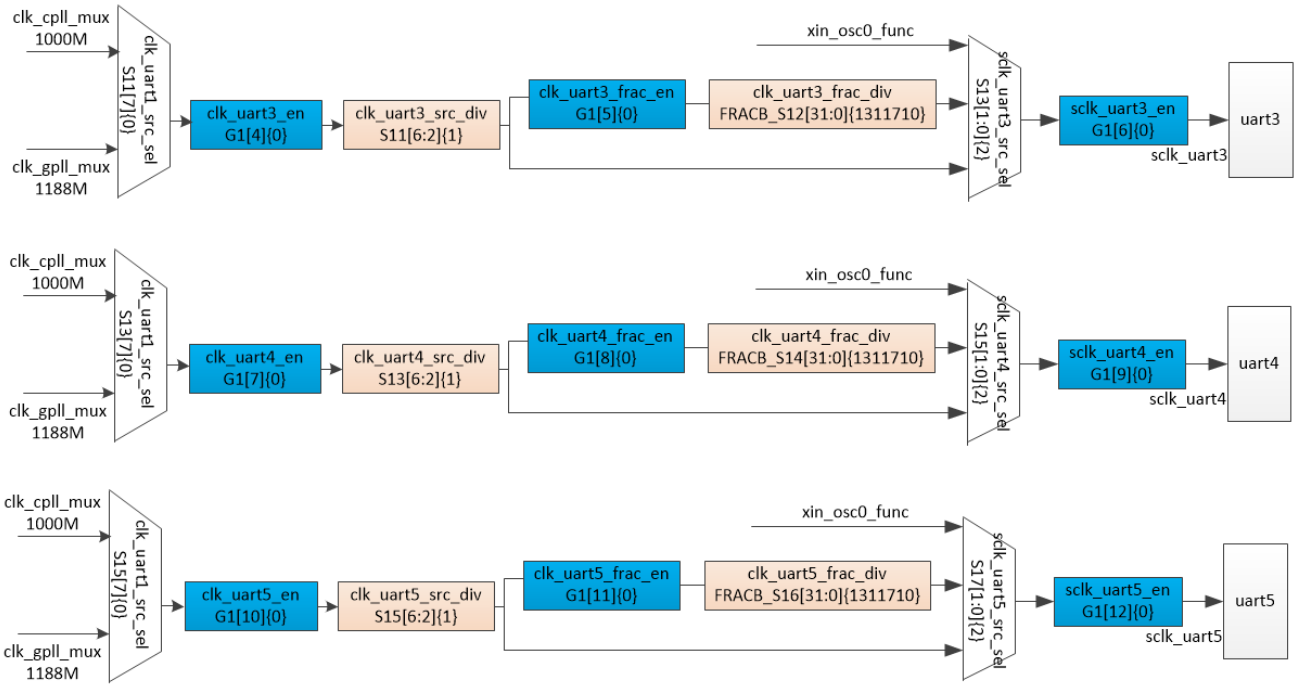


Fig. 13-10 UART3-UART5 clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 13-7 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 8.
460.8 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 2.
921.6 Kbps	Configure PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose PLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1.
3 Mbps	Choose PLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1.
4 Mbps	Configure PLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1.

Chapter 14 GPIO

14.1 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 can also read back the data on external pads using memory-mapped registers.
GPIO supports the following features:

- 32 bits APB data bus width
- Up to 32 independently configurable signals
- Software control registers with write mask for each bit of each signal
- Configurable debounce logic with a slow clock to debounce interrupts
- Configurable interrupt mode
- Two virtual OS with independent control registers can be supported
- In two virtual OS models, each OS has an independent interrupt
- Not in two virtual OS models, two interrupts with priority can be set

14.2 Block Diagram

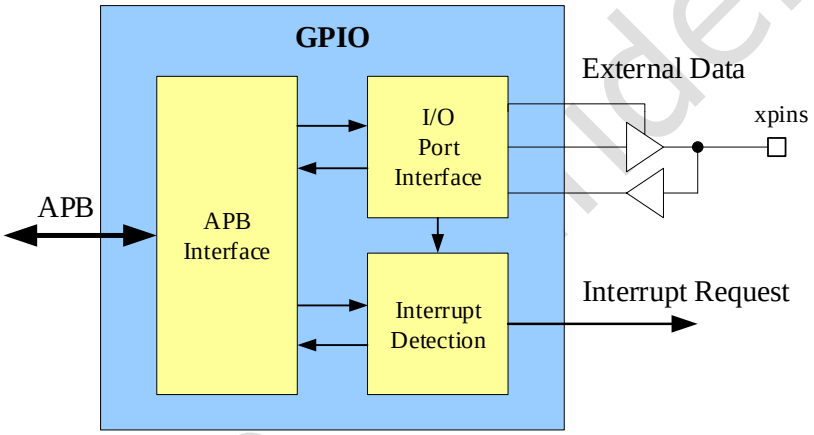


Fig. 14-1 GPIO Block Diagram

GPIO is comprised of:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- I/O Port Interface

External data interface to or from I/O pads.

- Interrupt Detection

Interrupt interface to or from interrupt controller.

14.3 Function Description

14.3.1 Data Control

Under software control, the data and direction control for the signal are sourced from the port data registers (GPIO_SWPORT_DR_L/GPIO_SWPORT_DR_H) and direction control registers (GPIO_SWPORT_DDR_L/GPIO_SWPORT_DRR_H).
The direction of the external I/O pad is controlled by the value of the port data direction registers. The data written to these memory-mapped registers gets mapped onto an output signal (gpio_port_ddr) of the GPIO peripheral. This output signal controls the direction of an external I/O pad. The default data direction is Input.
The data written to the port data registers drives the output buffer (gpio_port_dr) of the I/O pad.
External data are input on the external data signal (gpio_ext_port). Reading the external signal register (GPIO_EXT_PORT) shows the value of this signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software

interface.

14.3.2 Interrupts

I/O port 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_INT_MASK_L/GPIO_INT_MASK_H registers. The interrupt status can be read before masking (GPIO_INT_RAWSTATUS) and after masking (GPIO_INT_STATUS).

For edge-sensitive interrupts, the Interrupt Service Routine (ISR) can clear the interrupt by writing a 1 to the corresponding bit of the GPIO_PORT_EOI_L/GPIO_PORT_EOI_H registers. This write operation also clears the interrupt status and raw status registers. Writing to the interrupt clear registers has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the interrupt raw status register until the interrupt source disappears, or it can write to the interrupt mask 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.

The interrupts are combined into an active-high interrupt output signal. 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 I/O port is configured for interrupts, the data direction must be set to Input. If the data direction is reprogrammed to Output, then any pending edge-sensitive interrupts are not lost. However, no new interrupts are generated, and level-sensitive interrupts are lost.

Interrupt signals are internally synchronized to a system clock pclk_intr, which is connected to the APB bus clock pclk. Therefore, the pclk needs to be running for interrupt detection.

14.3.3 Debounce Operation

The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When an input interrupt signal is debounced using a slow debounce clock (external input clock dbclk or internally divided clock dbclk_div), the signal must be active for a minimum of two cycles of the debounce clock to guarantee that it is 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 rising edge, it is not registered.

The debounce function can be controlled by programming the debounce enable registers (GPIO_DEBOUNCE_L/GPIO_DEBOUNCE_H), debounce clock divide enable registers (GPIO_DBCLK_DIV_EN_L/GPIO_DBCLK_DIV_EN_H) and debounce clock divide control register (GPIO_DBCLK_DIV_CON).

14.3.4 Two OS Operation and Tow Interrupt

To select this model, a virtual enable register should be set(GPIO_VIRTUAL_EN). Then all the configure can be allotted to two OS（for example OS_A and OS_B). OS_A will use the original address offset, OS_B will use the offset + 0x1000. The 32 bit I/O port should also be allotted to the OS by setting reg group registers(GPIO_REG_GROUP_L and GPIO_REG_GROUP_H). Once reg group is set, the I/O operation can only be used by the setting address offset. Each OS has its own interrupt for I/O port, this is depended on reg

group.
If the virtual enable register is disabled, reg group can also be used to allotted I/O interrupt. These two independent interrupts may be used for priority interrupt settings.

14.4 Register Description

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORT_DR_L	0x0000	W	0x00000000	Port Data Register (Low)
GPIO_SWPORT_DR_H	0x0004	W	0x00000000	Port Data Register (High)
GPIO_SWPORT_DDR_L	0x0008	W	0x00000000	Port Data Direction Register (Low)
GPIO_SWPORT_DDR_H	0x000C	W	0x00000000	Port Data Direction Register (High)
GPIO_INT_EN_L	0x0010	W	0x00000000	Interrupt Enable Register (Low)
GPIO_INT_EN_H	0x0014	W	0x00000000	Interrupt Enable Register (High)
GPIO_INT_MASK_L	0x0018	W	0x00000000	Interrupt Mask Register (Low)
GPIO_INT_MASK_H	0x001C	W	0x00000000	Interrupt Mask Register (High)
GPIO_INT_TYPE_L	0x0020	W	0x00000000	Interrupt Level Register (Low)
GPIO_INT_TYPE_H	0x0024	W	0x00000000	Interrupt Level Register (High)
GPIO_INT_POLARITY_L	0x0028	W	0x00000000	Interrupt Polarity Register (Low)
GPIO_INT_POLARITY_H	0x002C	W	0x00000000	Interrupt Polarity Register (High)
GPIO_INT_BOTHEDGE_L	0x0030	W	0x00000000	Interrupt Both Edge Type Register (Low)
GPIO_INT_BOTHEDGE_H	0x0034	W	0x00000000	Interrupt Both Edge Type Register (High)
GPIO_DEBOUNCE_L	0x0038	W	0x00000000	Debounce Enable Register (Low)
GPIO_DEBOUNCE_H	0x003C	W	0x00000000	Debounce Enable Register (High)
GPIO_DBCLK_DIV_EN_L	0x0040	W	0x00000000	DBCLK Divide Enable Register (Low)
GPIO_DBCLK_DIV_EN_H	0x0044	W	0x00000000	DBCLK Divide Enable Register (High)
GPIO_DBCLK_DIV_CON	0x0048	W	0x00000001	DBCLK Divide Control Register
GPIO_INT_STATUS	0x0050	W	0x00000000	Interrupt Status Register
GPIO_INT_RAWSTATUS	0x0058	W	0x00000000	Interrupt Raw Status Register
GPIO_PORT_EOI_L	0x0060	W	0x00000000	Interrupt Clear Register (Low)
GPIO_PORT_EOI_H	0x0064	W	0x00000000	Interrupt Clear Register (High)
GPIO_EXT_PORT	0x0070	W	0x00000000	External Port Data Register
GPIO_VER_ID	0x0078	W	0x0101157C	Version ID Register
GPIO_GPIO_REG_GROUP_L	0x0100	W	0x00000000	GPIO Group Control
GPIO_GPIO_REG_GROUP_H	0x0104	W	0x0000FFFF	GPIO Group Control
GPIO_GPIO_VIRTUAL_EN	0x0108	W	0x00000000	GPIO Virtual Enable

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

14.4.2 Detail Registers Description

GPIO_SWPORT_DR_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	swport_dr_low Output data for the lower 16 bits of I/O Port, each bit is individual. 1'b0: Low 1'b1: High Values written to this register are output on the I/O signals for the lower 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DR_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_dr_high Output data for the upper 16 bits of I/O Port, each bit is individual. 1'b0: Low 1'b1: High Values written to this register are output on the I/O signals for the upper 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DDR_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_ddr_low Data direction for the lower 16 bits of I/O Port, each bit is individual. 1'b0: Input 1'b1: Output Values written to this register independently control the direction of the corresponding data bit in the lower 16 bits of I/O Port.

GPIO_SWPORT_DDR_H

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	swport_ddr_high Data direction for the upper 16 bits of I/O Port, each bit is individual. 1'b0: Input 1'b1: Output Values written to this register independently control the direction of the corresponding data bit in the upper 16 bits of I/O Port.

GPIO_INT_EN_L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_en_low Allows each bit of the lower 16 bits of I/O Port to be configured for interrupts. 1'b0: Interrupt is disabled 1'b1: Interrupt is enabled Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt source; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output.

GPIO_INT_EN_H

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_en_high Allows each bit of the upper 16 bits of I/O Port to be configured for interrupts. 1'b0: Interrupt is disabled 1'b1: Interrupt is enabled Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt source; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output.

GPIO_INT_MASK_L

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_mask_low Controls whether an interrupt on the lower 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 1'b0: Interrupt is unmasked 1'b1: Interrupt is masked 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.

GPIO_INT_MASK_H

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_mask_high Controls whether an interrupt on the upper 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 1'b0: Interrupt is unmasked 1'b1: Interrupt is masked 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.

GPIO_INT_TYPE_L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_type_low Controls the type of interrupt that can occur on the lower 16 bits of I/O Port. 1'b0: Level-sensitive 1'b1: Edge-sensitive Whenever a 1 is written to a bit of this register, it configures the interrupt type to be edge-sensitive; otherwise, it is level-sensitive.

GPIO_INT_TYPE_H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_type_high Controls the type of interrupt that can occur on the upper 16 bits of I/O Port. 1'b0: Level-sensitive 1'b1: Edge-sensitive Whenever a 1 is written to a bit of this register, it configures the interrupt type to be edge-sensitive; otherwise, it is level-sensitive.

GPIO_INT_POLARITY_L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_polarity_low Controls the polarity of edge or level sensitivity that can occur on the lower 16 bits of I/O Port. 1'b0: Active-low 1'b1: Active-high Whenever a 1 is written to a bit of this register, it configures the interrupt type to rising-edge or active-high sensitive; otherwise, it is falling-edge or active-low sensitive.

GPIO_INT_POLARITY_H

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_polarity_high Controls the polarity of edge or level sensitivity that can occur on the upper 16 bits of I/O Port. 1'b0: Active-low 1'b1: Active-high Whenever a 1 is written to a bit of this register, it configures the interrupt type to rising-edge or active-high sensitive; otherwise, it is falling-edge or active-low sensitive.

GPIO_INT_BOTHEDGE_L

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	int_bothedge_low Controls the edge type of interrupt that can occur on the lower 16 bits of I/O Port. 1'b0: Disable both-edge detection 1'b1: Enable both-edge detection 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 I/O Port. The values programmed in the registers int_type_low and int_polarity_low 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 int_type_low and int_polarity_low registers.

GPIO_INT_BOTHEDGE_H

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_bothedge_high Controls the edge type of interrupt that can occur on the upper 16 bits of I/O Port. 1'b0: Disable both-edge detection 1'b1: Enable both-edge detection 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 I/O Port. The values programmed in the registers int_type_high and int_polarity_high 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 int_type_high and int_polarity_high registers.

GPIO_DEBOUNCE_L

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	debounce_low Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 1'b0: Disable debounce 1'b1: Enable debounce 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.

GPIO_DEBOUNCE_H

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	debounce_high Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 1'b0: Disable debounce 1'b1: Enable debounce 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.

GPIO_DBCLK_DIV_EN_L

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	dbclk_div_en_low Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the lower 16 bits of I/O Port. 1'b0: Disable divider for debounce clock 1'b1: Enable divider for debounce clock Whenever a 1 is written to a bit of this register, the clock divided from dbclk is used as debounce clock; otherwise, the original dbclk is used. The clock divider factor depends on the register dbclk_div_con. The values programmed in this register for this particular bit are not considered when the corresponding bit of the register debounce_low is set to 0.

GPIO_DBCLK_DIV_EN_H

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	dbclk_div_en_high Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the upper 16 bits of I/O Port. 1'b0: Disable divider for debounce clock 1'b1: Enable divider for debounce clock Whenever a 1 is written to a bit of this register, the clock divided from dbclk is used as debounce clock; otherwise, the original dbclk is used. The clock divider factor depends on the register dbclk_div_con. The values programmed in this register for this particular bit are not considered when the corresponding bit of the register debounce_high is set to 0.

GPIO_DBCLK_DIV_CON

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:0	RW	0x000001	dbclk_div_con dbclk_div = dbclk / (dbclk_div_con + 1)

GPIO_INT_STATUS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	int_status Interrupt status of I/O Port.

GPIO_INT_RAWSTATUS

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	int_rawstatus Interrupt raw status of I/O Port (pre masking bits).

GPIO_PORT_EOI_L

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	R/WSC	0x0000	port_eoi_low Controls the clearing of edge type interrupts from the lower 16 bits of I/O Port. 1'b0: Nothing 1'b1: Clear edge-sensitive interrupt When a 1 is written into a corresponding bit of this register, the interrupt is cleared and the bit is self cleared at once. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts.

GPIO_PORT_EOI_H

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	R/WSC	0x0000	port_eoi_high Controls the clearing of edge type interrupts from the upper 16 bits of I/O Port. 1'b0: Nothing 1'b1: Clear edge-sensitive interrupt When a 1 is written into a corresponding bit of this register, the interrupt is cleared and the bit is self cleared at once. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts.

GPIO_EXT_PORT

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ext_port This register always reflects the value of the signals on the external I/O Port.

GPIO_VER_ID

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:0	RO	0x0101157c	ver_id Version ID.

GPIO_GPIO_REG_GROUP_L

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	gpio_reg_group_low This register controls the low 16 bit GPIO and each bit corresponds to each GPIO. When virtual_en=1'b1: 1'b1: GPIO control by OS_A with offset 0x0000 1'b0: GPIO control by OS_B with offset 0x1000 When virtual_en=1'b0: 1'b1: GPIO interrupt connect to gpio_int_flag 1'b0: GPIO interrupt connect to gpio_int_flag_exp

GPIO_GPIO_REG_GROUP_H

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0xffff	gpio_reg_group_high This register controls the high 16 bit GPIO and each bit corresponds to each GPIO. When virtual_en=1'b1: 1'b1: GPIO control by OS_A with offset 0x0000 1'b0: GPIO control by OS_B with offset 0x1000 When virtual_en=1'b0: 1'b1: GPIO interrupt connect to gpio_int_flag 1'b0: GPIO interrupt connect to gpio_int_flag_exp

GPIO_GPIO_VIRTUAL_EN

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	gpio_virtual_en 1'b1: Enable virtual, two OS supported 1'b0: Disable virtual.

14.5 Interface Description

Table 14-1 GPIO Interface Description

Module Pin	Pad Name	IOMUX Setting
GPIO0 Interface		
gpio0_port[0]	RTC_CLKO/REF_CLK_OUT/CLKIO_32K/UART0_RX_M0/GPIO0_A0_z	PMU_IOC_GPIO0A_IOMUX_SEL_L[2:0]=3'h0
gpio0_port[1]	TEST_CLK6_OUT/PWM2_M0/UART0_TX_M0/GPIO0_A1_d	PMU_IOC_GPIO0A_IOMUX_SEL_L[6:4]=3'h0
gpio0_port[2]	TEST_CLK7_OUT//PWM3_IR_M0/GPIO0_A2_d	PMU_IOC_GPIO0A_IOMUX_SEL_L[10:8]=3'h0
gpio0_port[3]	PMIC_SLEEP_M1/GPIO0_A3_u	PMU_IOC_GPIO0A_IOMUX_SEL_L[14:12]=3'h0
gpio0_port[4]	PWM1_M0/PMIC_SLEEP_M0/GPIO0_A4_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[2:0]=3'h0
gpio0_port[5]	PWM5_M0/UART1_RTSN_M0/I2C1_S	PMU_IOC_GPIO0A_IOMUX_SEL

Module Pin	Pad Name	IOMUX Setting
	CL_M0/GPIO0_A5_d	_H[6:4]=3'h0
gpio0_port[6]	PWM6_M0/UART1_CTSN_M0/I2C1_SDA_M0/GPIO0_A6_d	PMU_IOC_GPIO0A_IOMUX_SEL_H[10:8]=3'h0
GPIO1 Interface		
gpio1_port[0]	PWM7_IR_M0/I2C2_SCL_M0/UART3_TX_M0/GPIO1_A0_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[2:0]=3'h0
gpio1_port[1]	PWM4_M0/PMU_DEBUG/I2C2_SDA_M0/UART3_RX_M0/GPIO1_A1_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[6:4]=3'h0
gpio1_port[2]	VICAP_D0_M1/AVS_ARM/PWM0_M0/GPIO1_A2_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[10:8]=3'h0
gpio1_port[3]	I2C0_SCL_M0/UART1_TX_M0/GPIO1_A3_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_L[14:12]=3'h0
gpio1_port[4]	I2C0_SDA_M0/UART1_RX_M0/GPIO1_A4_d	VCCIO1_IOC_GPIO1A_IOMUX_SEL_H[2:0]=3'h0
gpio1_port[8]	PWM3_IR_M1/UART4_RX_M0/GPIO1_B0_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[2:0]=3'h0
gpio1_port[9]	VICAP_D1_M1/SPI1_CS1n_M0/PWM7_IR_M1/UART4_TX_M0/GPIO1_B1_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[6:4]=3'h0
gpio1_port[10]	LPMCU_JTAG_TCK_M0/HPMCU_JTAG_TCK_M0/UART2_TX_M1/A7_JTAG_TCK_M1/GPIO1_B2_d	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[10:8]=3'h0
gpio1_port[11]	LPMCU_JTAG_TMS_M0/HPMCU_JTAG_TMS_M0/UART2_RX_M1/A7_JTAG_TMS_M1/GPIO1_B3_u	VCCIO1_IOC_GPIO1B_IOMUX_SEL_L[14:12]=3'h0
gpio1_port[16]	SDMMC1_D1_M1/SPI0_CS0n_M0/PWM2_M2/VICAP_D2_M1/LCD_D7/GPIO1_C0_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[2:0]=3'h0
gpio1_port[17]	SDMMC1_D0_M1/SPI0_CLK_M0/PWM4_M2/VICAP_D3_M1/LCD_D6/GPIO1_C1_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[6:4]=3'h0
gpio1_port[18]	SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[10:8]=3'h0
gpio1_port[19]	SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_L[14:12]=3'h0
gpio1_port[20]	SDMMC1_D3_M1/UART4_RX_M1/PWM8_M1/VICAP_D6_M1/LCD_D3/GPIO1_C4_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[2:0]=3'h0
gpio1_port[21]	SDMMC1_D2_M1/UART4_TX_M1/PWM9_M1/VICAP_D7_M1/LCD_D2/GPIO1_C5_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[6:4]=3'h0
gpio1_port[22]	UART4_RTSN_M1/PWM10_M1/VICAP_D8_M1/LCD_D1/GPIO1_C6_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[10:8]=3'h0
gpio1_port[23]	UART4_CTSN_M1/PWM11_IR_M1/VICAP_D9_M1/LCD_D0/GPIO1_C7_d	VCCIO6_IOC_GPIO1C_IOMUX_SEL_H[14:12]=3'h0
gpio1_port[24]	UART3_TX_M1/UART5_RTSN_M1/PWM3_IR_M2/VICAP_CLKIN_M1/LCD_DEN/GPIO1_D0_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[2:0]=3'h0
gpio1_port[25]	UART3_RX_M1/UART5_CTSN_M1/PWM10_M2/VICAP_HSYNC_M1/LCD_HSYNC/GPIO1_D1_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[6:4]=3'h0
gpio1_port[26]	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[10:8]=3'h0

Module Pin	Pad Name	IOMUX Setting
	_D2_d	
gpio1_port[27]	DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VCCIO6_IOC_GPIO1D_IOMUX_SEL_L[14:12]=3'h0
GPIO2 Interface		
gpio2_port[0]	I2C4_SDA_M0/UART1_CTSN_M1/LCD_D8/I2S0_SCLK/SDMMC1_D1_M0/GPIO2_A0_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_L[2:0]=3'h0
gpio2_port[1]	I2C4_SCL_M0/UART1_RTSN_M1/LCD_D9/I2S0_LRCK/SDMMC1_D0_M0/GPIO2_A1_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_L[6:4]=3'h0
gpio2_port[2]	LCD_D10/I2S0_MCLK/SDMMC1_CLK_M0/GPIO2_A2_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_L[10:8]=3'h0
gpio2_port[3]	LCD_D11/I2S0_SDO3_SDI1/SDMMC1_CMD_M0/GPIO2_A3_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_L[14:12]=3'h0
gpio2_port[4]	UART1_TX_M1/LCD_D12/I2S0_SDO0/SDMMC1_D3_M0/GPIO2_A4_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[2:0]=3'h0
gpio2_port[5]	UART1_RX_M1/LCD_D13/I2S0_SDI0/SDMMC1_D2_M0/GPIO2_A5_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[6:4]=3'h0
gpio2_port[6]	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[10:8]=3'h0
gpio2_port[7]	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VCCIO5_IOC_GPIO2A_IOMUX_SEL_H[14:12]=3'h0
gpio2_port[8]	TEST_CLK4_OUT/PWM5_M1/LCD_D16/I2C1_SDA_M1/UART0_RX_M1/GPIO2_B0_d	VCCIO5_IOC_GPIO2B_IOMUX_SEL_L[2:0]=3'h0
gpio2_port[9]	TEST_CLK5_OUT/PWM6_M1/LCD_D17/I2C1_SCL_M1/UART0_TX_M1/GPIO2_B1_d	VCCIO5_IOC_GPIO2B_IOMUX_SEL_L[6:4]=3'h0
GPIO3 Interface		
gpio3_port[1]	SDMMC0_DET/GPIO3_A1_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[6:4]=3'h0
gpio3_port[2]	PWM9_M0/TEST_CLK0_OUT/UART2_TX_M0/SDMMC0_D1/GPIO3_A2_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[10:8]=3'h0
gpio3_port[3]	PWM8_M0/TEST_CLK1_OUT/UART2_RX_M0/SDMMC0_D0/GPIO3_A3_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_L[14:12]=3'h0
gpio3_port[4]	PWM10_M0/LPMCU_JTAG_TCK_M1/I2C0_SCL_M2/UART5_RTSN_M0/SDMMC0_CLK/GPIO3_A4_d	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[2:0]=3'h0
gpio3_port[5]	PWM11_IR_M0/LPMCU_JTAG_TMS_M1/I2C0_SDA_M2/UART5_CTSN_M0/SDMMC0_CMD/GPIO3_A5_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[6:4]=3'h0
gpio3_port[6]	HPMCU_JTAG_TMS_M1/A7_JTAG_TMS_M0/UART5_TX_M0/SDMMC0_D3/GPIO3_A6_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[10:8]=3'h0
gpio3_port[7]	HPMCU_JTAG_TCK_M1/A7_JTAG_TCK_M0/UART5_RX_M0/SDMMC0_D2/GPIO3_A7_u	VCCIO4_IOC_GPIO3A_IOMUX_SEL_H[14:12]=3'h0
gpio3_port[8]	MIPI/LVDS_D3N/VICAP_D0_M0/GPIO3_B0_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_L[2:0]=3'h0
gpio3_port[9]	MIPI/LVDS_D3P/VICAP_D1_M0/GPIO3_B1_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_L[6:4]=3'h0
gpio3_port[10]	MIPI/LVDS_CK1N/VICAP_D2_M0/GPI	VCCIO7_IOC_GPIO3B_IOMUX_

Module Pin	Pad Name	IOMUX Setting
	O3_B2_d	SEL_L[10:8]=3'h0
gpio3_port[11]	MIPI/LVDS_CK1P/VICAP_D3_M0/GPIO3_B3_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_L[14:12]=3'h0
gpio3_port[12]	MIPI/LVDS_D2N/VICAP_D4_M0/GPIO3_B4_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_H[2:0]=3'h0
gpio3_port[13]	MIPI/LVDS_D2P/VICAP_D5_M0/GPIO3_B5_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_H[6:4]=3'h0
gpio3_port[14]	MIPI/LVDS_D1N/VICAP_D6_M0/GPIO3_B6_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_H[10:8]=3'h0
gpio3_port[15]	MIPI/LVDS_D1P/VICAP_D7_M0/GPIO3_B7_d	VCCIO7_IOC_GPIO3B_IOMUX_SEL_H[14:12]=3'h0
gpio3_port[16]	MIPI/LVDS_CK0N/VICAP_D8_M0/GPIO3_C0_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_L[2:0]=3'h0
gpio3_port[17]	MIPI/LVDS_CK0P/VICAP_D9_M0/GPIO3_C1_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_L[6:4]=3'h0
gpio3_port[18]	MIPI/LVDS_D0N/VICAP_CLKIN_M0/GPIO3_C2_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_L[10:8]=3'h0
gpio3_port[19]	MIPI/LVDS_D0P/VICAP_HSYNC_M0/GPIO3_C3_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_L[14:12]=3'h0
gpio3_port[20]	MIPI_REFCLK_OUT0/VICAP_CLKOUT_M0/GPIO3_C4_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_H[2:0]=3'h0
gpio3_port[21]	VICAP_VSYNC_M0/GPIO3_C5_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_H[6:4]=3'h0
gpio3_port[22]	MIPI_REFCLK_OUT1/PWM7_IR_M2/VICAP_D10/GPIO3_C6_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_H[10:8]=3'h0
gpio3_port[23]	I2C4_SCL_M2/UART5_TX_M2/VICAP_D11/GPIO3_C7_d	VCCIO7_IOC_GPIO3C_IOMUX_SEL_H[14:12]=3'h0
gpio3_port[24]	I2C4_SDA_M2/UART5_RX_M2/VICAP_D12/GPIO3_D0_d	VCCIO7_IOC_GPIO3D_IOMUX_SEL_L[2:0]=3'h0
gpio3_port[25]	I2C3_SCL_M2/UART5_RTSN_M2/VICAP_D13/GPIO3_D1_d	VCCIO7_IOC_GPIO3D_IOMUX_SEL_L[6:4]=3'h0
gpio3_port[26]	I2C3_SDA_M2/UART5_CTSN_M2/VICAP_D14/GPIO3_D2_d	VCCIO7_IOC_GPIO3D_IOMUX_SEL_L[10:8]=3'h0
gpio3_port[27]	PWM1_M2/VICAP_D15/GPIO3_D3_d	VCCIO7_IOC_GPIO3D_IOMUX_SEL_L[14:12]=3'h0
GPIO4 Interface		
gpio4_port[0]	TEST_CLK2_OUT/I2C0_SDA_M1/UART0_RX_M2/SPI1_MISO_M0/EMMC_D7/GPIO4_A0_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[2:0]=3'h0
gpio4_port[1]	TEST_CLK3_OUT/I2C0_SCL_M1/UART0_TX_M2/SPI1_MOSI_M0/EMMC_D6/GPIO4_A1_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[6:4]=3'h0
gpio4_port[2]	FSPI_D2/EMMC_D2/GPIO4_A2_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[10:8]=3'h0
gpio4_port[3]	FSPI_D1/EMMC_D1/GPIO4_A3_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[14:12]=3'h0
gpio4_port[4]	FSPI_D0/EMMC_D0/GPIO4_A4_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[2:0]=3'h0
gpio4_port[5]	I2C2_SDA_M1/UART1_TX_M2/SPI1_CS0n_M0/EMMC_D4/GPIO4_A5_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[6:4]=3'h0
gpio4_port[6]	FSPI_D3/EMMC_D3/GPIO4_A6_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[10:8]=3'h0
gpio4_port[7]	I2C2_SCL_M1/UART1_RX_M2/SPI1_CLK_M0/EMMC_D5/GPIO4_A7_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[14:12]=3'h0
gpio4_port[8]	FSPI_CS0n/EMMC_CMD/GPIO4_B0_u	VCCIO3_IOC_GPIO4B_IOMUX_

Module Pin	Pad Name	IOMUX Setting
		SEL_L[2:0]=3'h0
gpio4_port[9]	FSPI_CLK/EMMC_CLK/GPIO4_B1_d	VCCIO3_IOC_GPIO4B_IOMUX_SEL_L[6:4]=3'h0
gpio4_port[16]	ADCIN0/GPIO4_C0_z	VCCIO2_IOC_GPIO4C_IOMUX_SEL_L[2:0]=3'h0
gpio4_port[17]	PWM1_M1/ADCIN1/GPIO4_C1_z	VCCIO2_IOC_GPIO4C_IOMUX_SEL_L[6:4]=3'h0

Notes: Unused Module Pin is tied to zero!

14.6 Application Notes

- 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 detection should be completed prior to enabling the interrupts to prevent spurious glitches on the interrupt output signal to the interrupt controller.

Chapter 15 Inter-Integrated Circuit (I2C)

15.1 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 5 independent I2C: I2C0~4, with 13 pairs IOs.
- 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
- Filter out glitch on SCL and SDA

15.2 Block Diagram

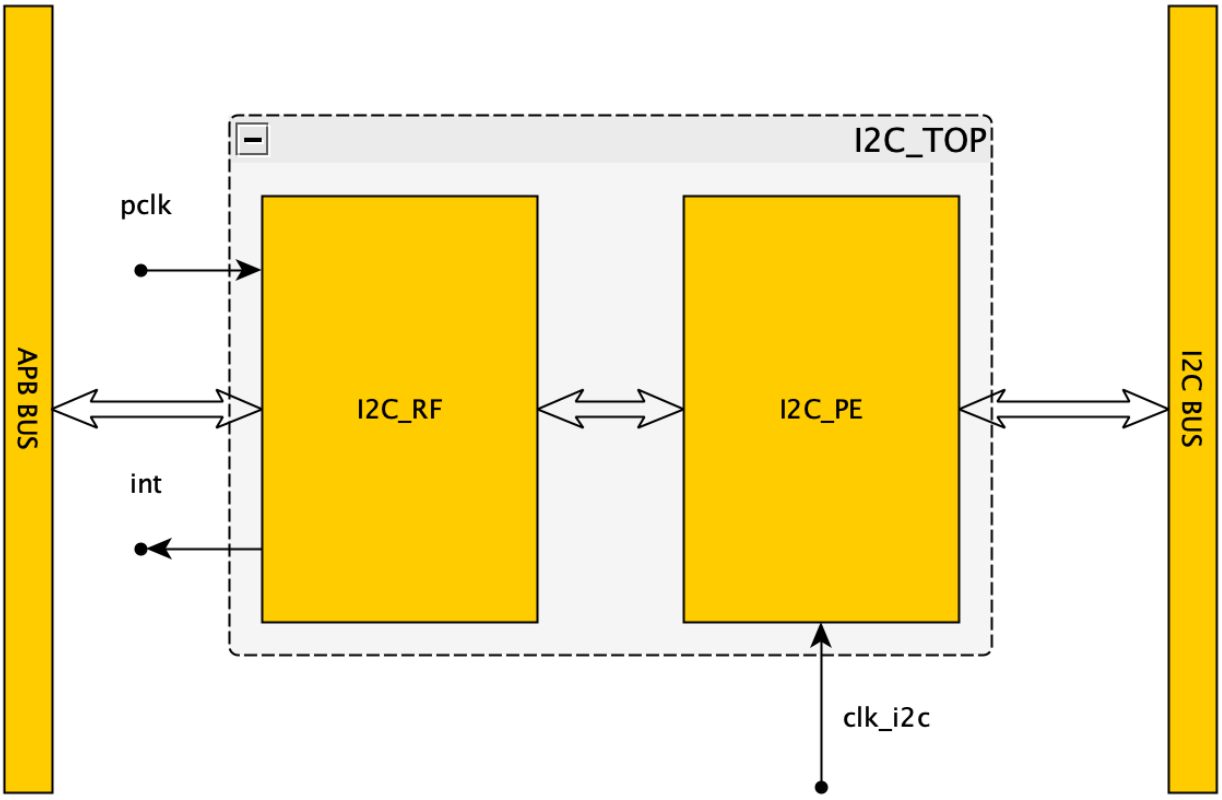


Fig. 15-1 I2C architecture

15.2.1 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 I2C_RF component operates synchronously with the pclk.

15.2.2 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 clk_i2c.

15.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

15.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.
The I2C controller supports only Master function. It supports 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.

15.3.1 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 to configure controller and uses clk_i2c as the working clock. The correct register setting is subject to the system requirement.

15.3.2 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); clk_i2c = 100MHz~200MHz.
  SCL = clk_i2c/ 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.

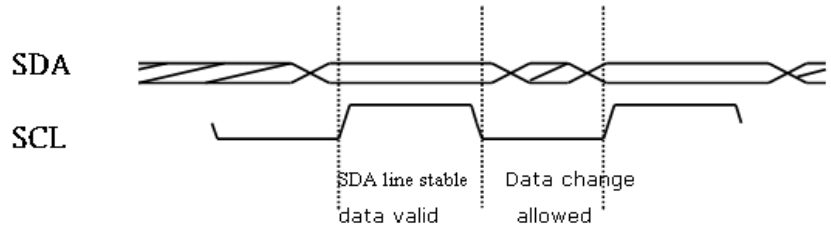


Fig. 15-2 I2C DATA Validity

- 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.

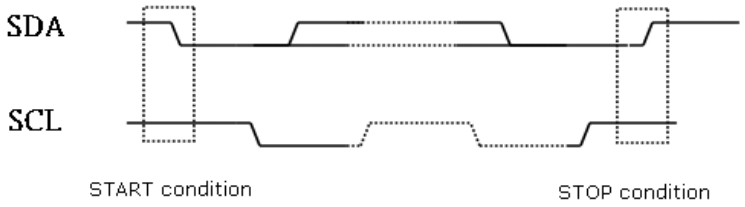


Fig. 15-3 I2C Start and stop conditions

- 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".

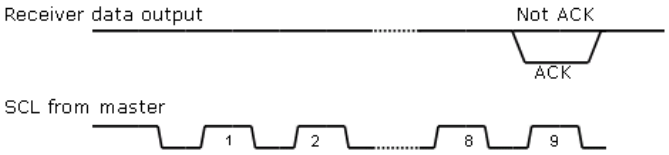


Fig. 15-4 I2C Acknowledge

  - 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.

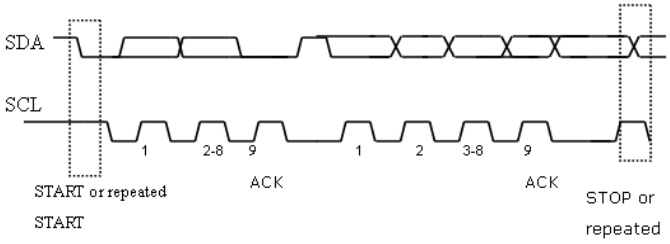


Fig. 15-5 I2C byte transfer

15.4 Register Description

15.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00050000	Control register

Name	Offset	Size	Reset Value	Description
RKI2C_CLKDIV	0x0004	W	0x00000001	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 received for 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
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	0x00000000	Status debug register
RKI2C_DBGCTRL	0x0224	W	0x00000000	Debug config register
RKI2C_CON1	0x0228	W	0x00000000	Control register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

15.4.2 Detail Registers Description

RKI2C_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0005	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.

Bit	Attr	Reset Value	Description
11:8	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.
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: Disable 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	0x00	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	0x00	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	0x0000000	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	0x0000000	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	0x000000	reserved

Bit	Attr	Reset Value	Description
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	0x000000	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.

Bit	Attr	Reset Value	Description
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	0x0000000	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	0x000000	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	0x00000000	reserved
1	RO	0x0	scl_st SCL status. 1'b0: SCL status low. 1'b0: SCL status high.
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	0x00000	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	h0_check_scl 1'b0: Check if scl been pull down by slave at the whole SCL_HIGH. 1'b1: 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 1'b0: Hold scl as low when recieved nack. 1'b1: Release scl as high when recieved nack.
12	RW	0x0	flt_en SCL edage glitch filter enable. 1'b0: Disable 1'b1: Enable
11:8	RW	0x0	slv_hold_scl_th Slave hold scl threshold = slv_hold_scl_db * 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.

RKI2C_CON1

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	auto_stop_nak Auto stop when i2c master recieved nak from slave. Work when CON1[0]=1'b1. 1'b0: Do not auto stop when recieved nak. 1'b1: Auto stop when recieved nak.
1	RW	0x0	auto_stop_tx_end Auto stop when i2c master tx end. Work when CON1[0]=1'b1. 1'b0: Do not auto stop when tx end. 1'b1: Auto stop when tx end.
0	RW	0x0	auto_stop Auto stop when i2c master recieved nak from slave or tx end. 1'b0: Do not auto stop when recieved nak or tx end. 1'b1: Auto stop when recieved nak or tx end.

15.5 Interface Description

Table 15-1 I2C Interface Description

Module pin	Direction	Pin name	IOMUX
I2C0_SCL_M0	I/O	I2C0_SCL_M0/UART1_TX_M0/GPIO1_A3_d	VENCIOC_GPIO1A_IOMUX_SEL_L[15:12] = 0x2
I2C0_SDA_M0	I/O	I2C0_SDA_M0/UART1_RX_M0/GPIO1_A4_d	VENCIOC_GPIO1A_IOMUX_SEL_H[3:0] = 0x2
I2C0_SCL_M1	I/O	TEST_CLK3_OUT/I2C0_SCL_M1/UART0_TX_M2/SPI1_MOSI_M0/EMMC_D6/GPIO4_A1_u	PERIIOC_GPIO4A_IOMUX_SEL_L[7:4] = 0x4
I2C0_SDA_M1	I/O	TEST_CLK2_OUT/I2C0_SDA_M1/UART0_RX_M2/SPI1_MISO_M0/EMMC_D7/GPIO4_A0_u	PERIIOC_GPIO4A_IOMUX_SEL_L[3:0] = 0x4
I2C0_SCL_M2	I/O	PWM10_M0/LPMCU_JTAG_TCK_M1/I2C0_SCL_M2/UART5_RTSN_M0/SDMMC0_CLK/GPIO3_A4_d	VIIOC_GPIO3A_IOMUX_SEL_H[3:0] = 0x3
I2C0_SDA_M2	I/O	PWM11_IR_M0/LPMCU_JTAG_TMS_M1/I2C0_SDA_M2/UART5_CTSN_M0/SDMMC0_CMD/GPIO3_A5_u	VIIOC_GPIO3A_IOMUX_SEL_H[7:4] = 0x3
I2C1_SCL_M0	I/O	PWM5_M0/UART1_RTSN_M0/I2C1_SCL_M0/GPIO0_A5_d	PMUIOC_GPIO0A_IOMUX_SEL_H[7:4] = 0x1
I2C1_SDA_M0	I/O	PWM6_M0/UART1_CTSN_M0/I2C1_SDA_M0/GPIO0_A6_d	PMUIOC_GPIO0A_IOMUX_SEL_H[11:8] = 0x1
I2C1_SCL_M1	I/O	TEST_CLK5_OUT/PWM6_M1/LCD_D17/I2C1_SCL_M1/UART0_TX_M1/GPIO2_B1_d	VOIOC_GPIO2B_IOMUX_SEL_L[3:0] = 0x2
I2C1_SDA_M1	I/O	TEST_CLK4_OUT/PWM5_M1/LCD_D16/I2C1_SDA_M1/UART0_RX_M1/GPIO2_B0_d	VOIOC_GPIO2B_IOMUX_SEL_L[7:4] = 0x2
I2C2_SCL_M0	I/O	PWM7_IR_M0/I2C2_SCL_M0/UART3_TX_M0/GPIO1_A0_d	VENCIOC_GPIO1A_IOMUX_SEL_L[3:0] = 0x2
I2C2_SDA_M0	I/O	PWM4_M0/PMU_DEBUG/I2C2_SDA_M0/UART3_RX_M0/GPIO1_A1_d	VENCIOC_GPIO1A_IOMUX_SEL_L[7:4] = 0x2
I2C2_SCL_M1	I/O	I2C2_SCL_M1/UART1_RX_M2/SPI1_CLK_M0/EMMC_D5/GPIO4_A7_u	PERIIOC_GPIO4A_IOMUX_SEL_H[15:12] = 0x4
I2C2_SDA_M1	I/O	I2C2_SDA_M1/UART1_TX_M2/SPI1_CS0n_M0/EMMC_D4/GPIO4_A5_u	PERIIOC_GPIO4A_IOMUX_SEL_H[7:4] = 0x4
I2C3_SCL_M0	I/O	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VOIOC_GPIO2A_IOMUX_SEL_H[11:8] = 0x5

Module pin	Direction	Pin name	IOMUX
I2C3_SDA_M0	I/O	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VOIOC_GPIO2A_IOMUX_SEL_H[15:12] = 0x5
I2C3_SCL_M1	I/O	DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VENCIOC_GPIO1D_IOMUX_SEL_L[15:12] = 0x3
I2C3_SDA_M1	I/O	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VENCIOC_GPIO1D_IOMUX_SEL_L[11:8] = 0x3
I2C3_SCL_M2	I/O	I2C3_SCL_M2/UART5_RTSN_M2/VICAP_D13/GPIO3_D1_d	VIIOC_GPIO3D_IOMUX_SEL_L[7:4] = 0x3
I2C3_SDA_M2	I/O	I2C3_SDA_M2/UART5_CTSN_M2/VICAP_D14/GPIO3_D2_d	VIIOC_GPIO3D_IOMUX_SEL_L[11:8] = 0x3
I2C4_SCL_M0	I/O	I2C4_SCL_M0/UART1_RTSN_M1/LCD_D9/I2S0_LRCK/SDMMC1_D0_M0/GPIO2_A1_d	VOIOC_GPIO2A_IOMUX_SEL_L[7:4] = 0x5
I2C4_SDA_M0	I/O	I2C4_SDA_M0/UART1_CTSN_M1/LCD_D8/I2S0_SCLK/SDMMC1_D1_M0/GPIO2_A0_d	VOIOC_GPIO2A_IOMUX_SEL_L[3:0] = 0x5
I2C4_SCL_M1	I/O	SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VENCIOC_GPIO1C_IOMUX_SEL_L[11:8] = 0x4
I2C4_SDA_M1	I/O	SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VENCIOC_GPIO1C_IOMUX_SEL_L[15:12] = 0x4
I2C4_SCL_M2	I/O	I2C4_SCL_M2/UART5_TX_M2/VICAP_D11/GPIO3_C7_d	VIIOC_GPIO3C_IOMUX_SEL_H[15:12] = 0x3
I2C4_SDA_M2	I/O	I2C4_SDA_M2/UART5_RX_M2/VICAP_D12/GPIO3_D0_d	VIIOC_GPIO3D_IOMUX_SEL_L[3:0] = 0x3

15.6 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)

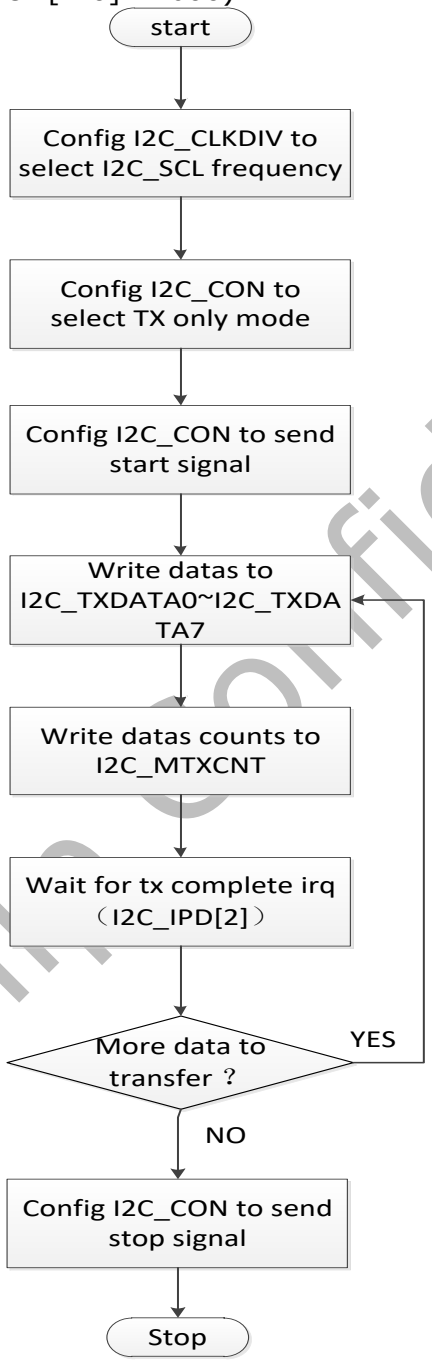


Fig. 15-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

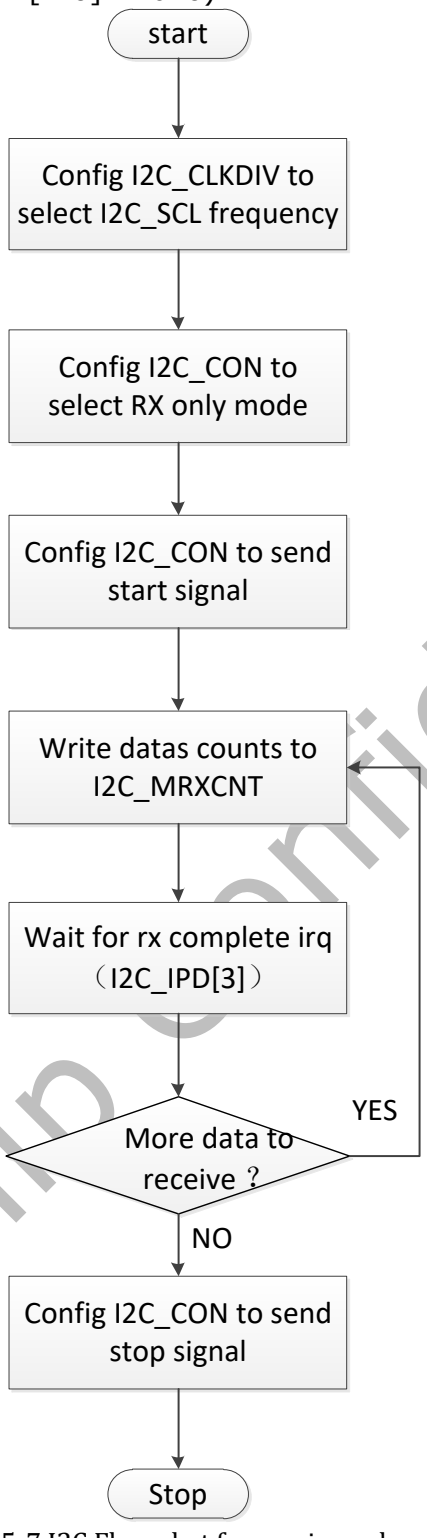


Fig. 15-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

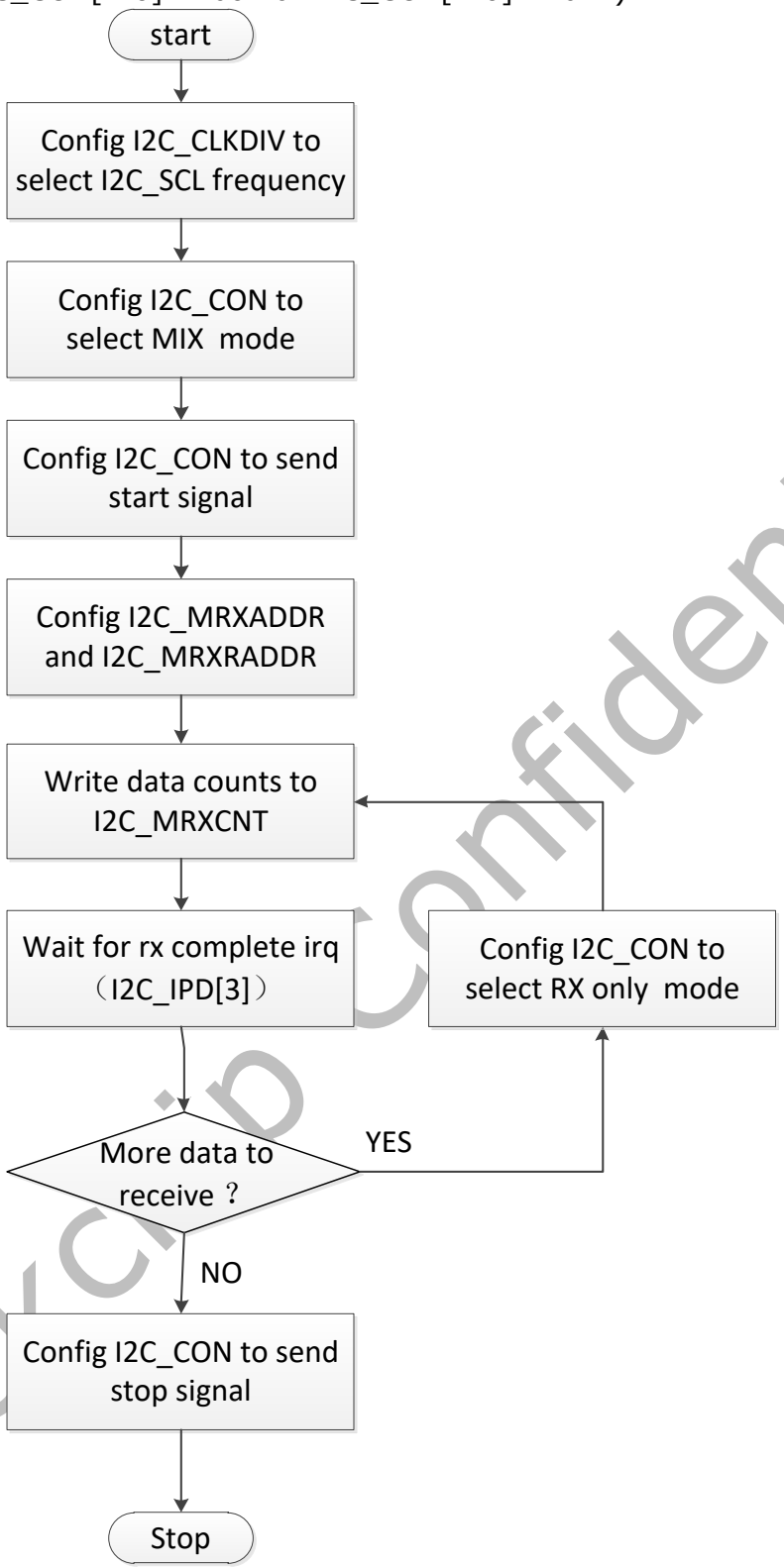


Fig. 15-8 I2C Flow chat for mix mode

Chapter 16 I2S

16.1 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.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.

16.1.1 Features

- Support eight internal 32-bit wide and 32-location deep FIFOs
- 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 2-channel audio transmitting and receiving in PCM mode
- Support 8-channel audio transmitting and receiving in I2S 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 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 configurable SCLK and LRCK polarity
- Support loopback function and rx channel data select

Support SDI,SDO IOMUX.

16.2 Block Diagram

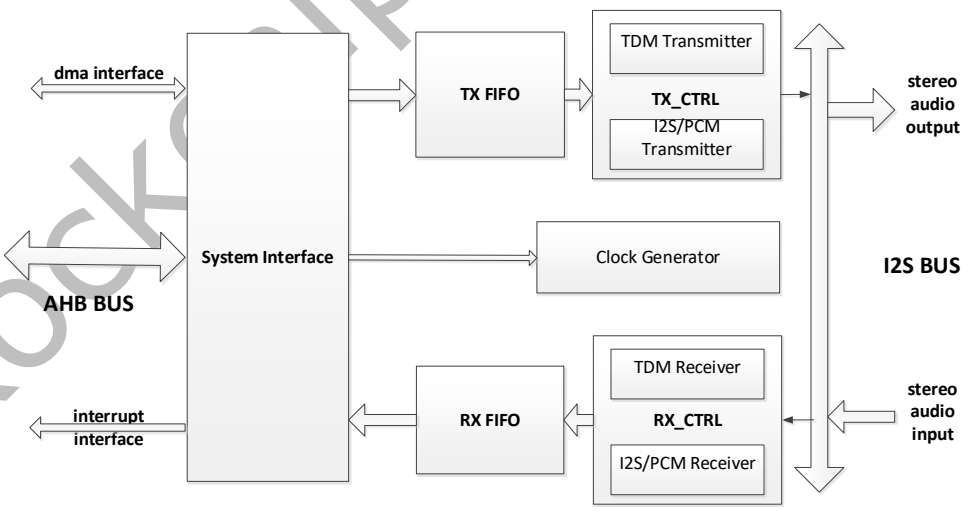


Fig. 16-1 I2S/PCM/TDM Controller (8-channel) Block Diagram

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. By the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

TX_CTRL
For I2S/PCM/TDM Controller, TX_CTRL includes TDM transmitter and I2S/PCM transmitter. 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.
RX_CTRL
For I2S/PCM/TDM Controller, RX_CTRL includes TDM receiver and I2S/PCM 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.
TX FIFO
The transmit FIFO is the buffer to store transmitted audio data. The size of one FIFO is 32bits x 32.
RX FIFO
The receive FIFO is the buffer to store received audio data. The size of one FIFO is 32bits x 32

16.3 Function Description

In the I2S/PCM/TDM, 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.

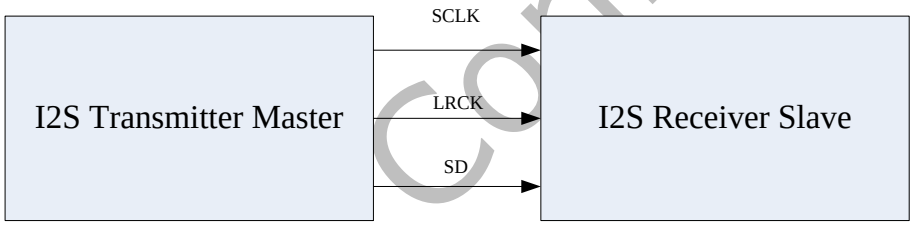


Fig. 16-2 I2S Transmitter-Master & Receiver-Slave Condition

When the transmitter acts as a master, it sends all signals to the receiver (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 transmitter to send data.

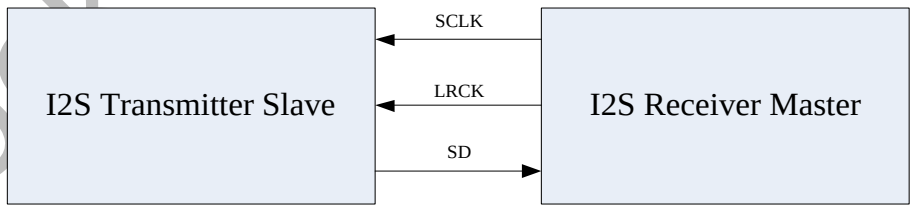


Fig. 16-3 I2S Transmitter-Slave & Receiver-Master Condition

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.

16.3.1 I2S Normal Mode

There is an APB slave interface in RTC. It is responsible for accessing registers.
This is the waveform of I2S normal mode. For LRCK (i2s_lrck) signal, it goes low to indicate

left channel and high to right channel. For SD (i2s_sdo, i2s_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.

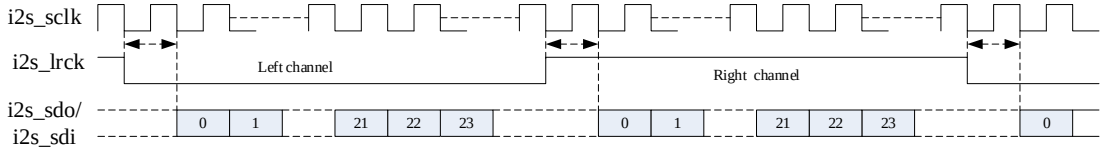


Fig. 16-4 I2S Normal Mode Timing Format

16.3.2 I2S Left Justified Mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_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.

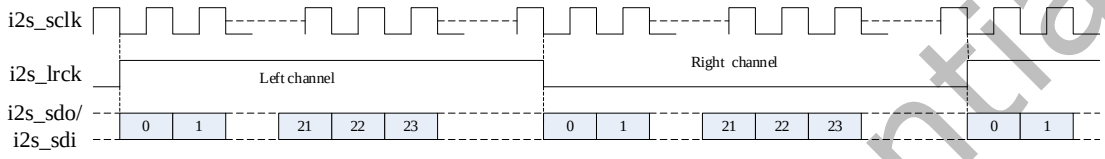


Fig. 16-5 I2S Left Justified Mode Timing Format

16.3.3 I2S Right Justified Mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck) 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 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.

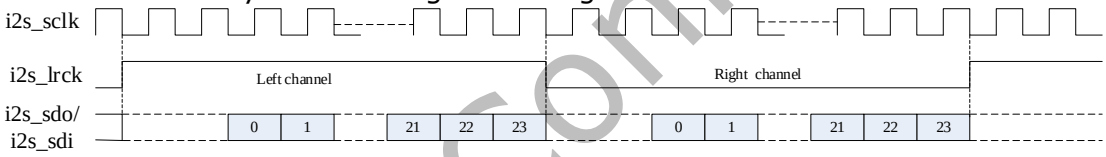


Fig. 16-6 I2S Right Justified Mode Timing Format

16.3.4 PCM Early Mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

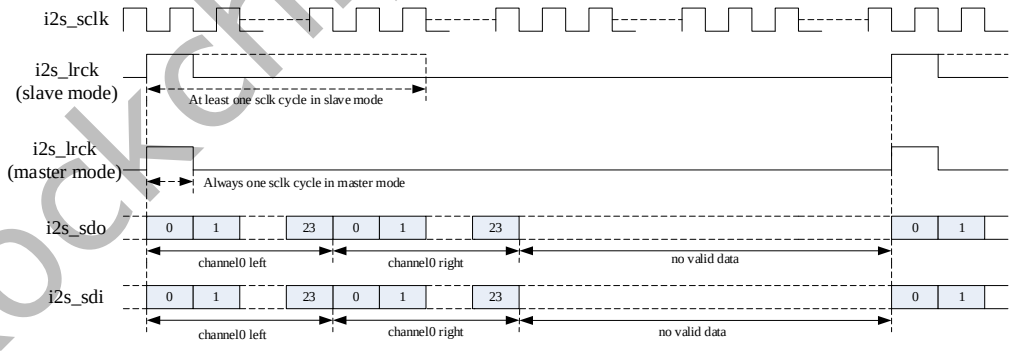


Fig. 16-7 PCM Early Mode Timing Format

16.3.5 PCM Late1 Mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

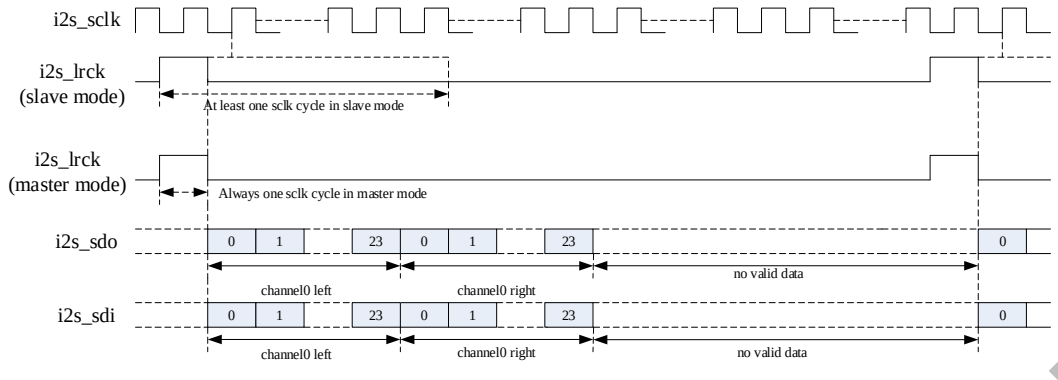


Fig. 16-8 PCM Late1 Mode Timing Format

16.3.6 PCM Late2 Mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

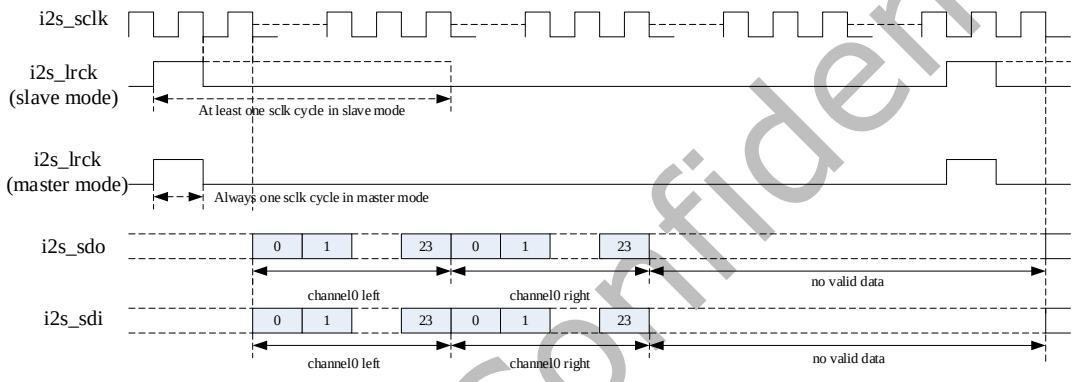


Fig. 16-9 PCM Late2 Mode Timing Format

16.3.7 PCM Late3 Mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

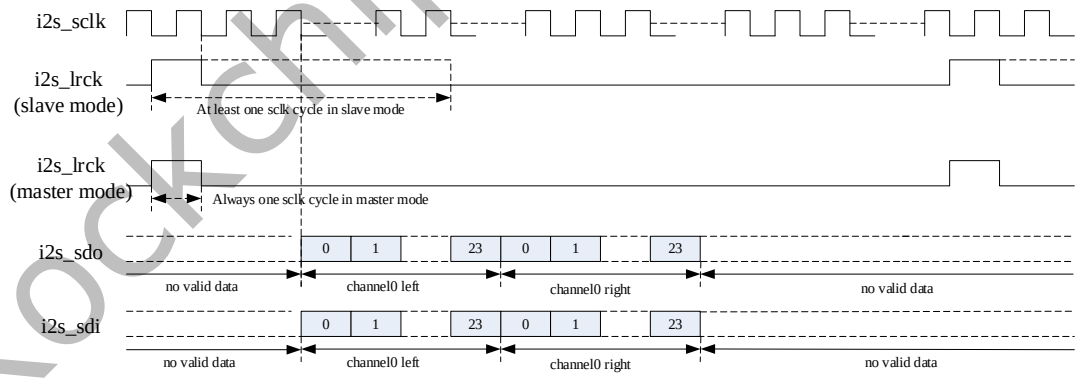


Fig. 16-10 PCM Late3 Mode Timing Format

16.3.8 TDM Normal Mode (PCM Format)

This is the waveform of TDM normal mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

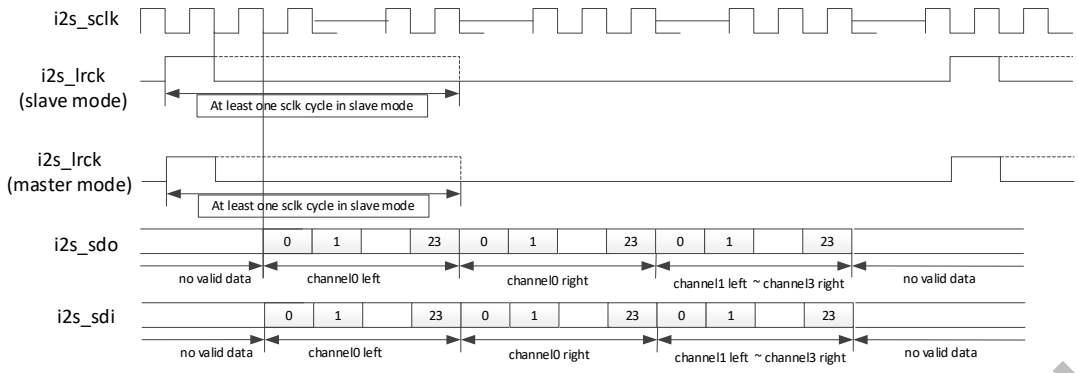


Fig. 16-11 TDM Normal Mode Timing Format (PCM Format)

16.3.9 TDM Left Shift Mode0 (PCM Format)

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

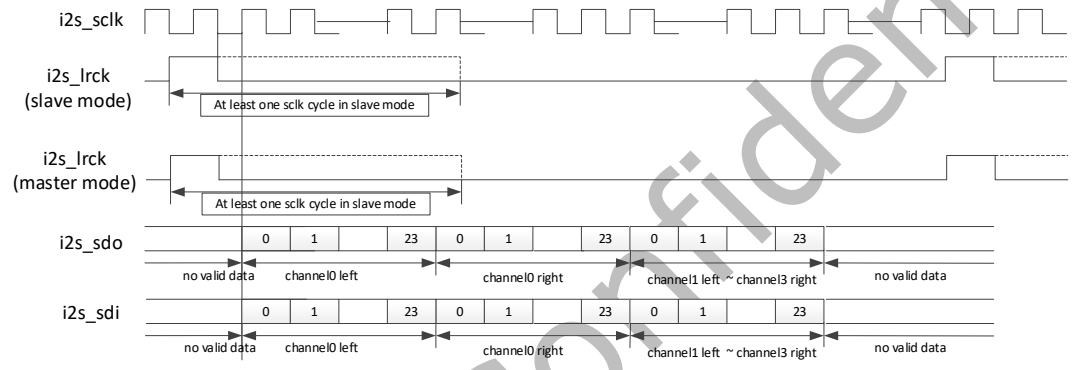


Fig. 16-12 TDM Left Shift Mode 0 Timing Format (PCM Format)

16.3.10 TDM Left Shift Mode1 (PCM Format)

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it sends the first bit (MSB or LSB) on the first falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.

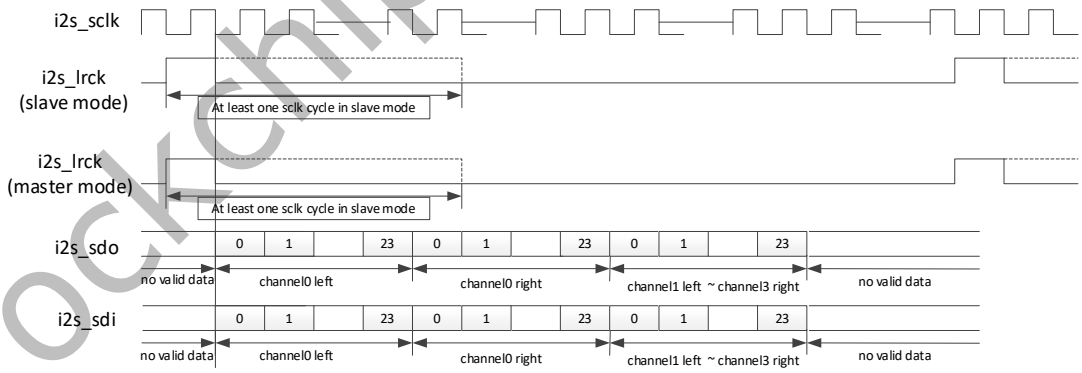


Fig. 16-13 TDM Left Shift Mode 1 Timing Format (PCM Format)

16.3.11 TDM Left Shift Mode2 (PCM Format)

This is the waveform of PCM early mode. For LRCK (i2s_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

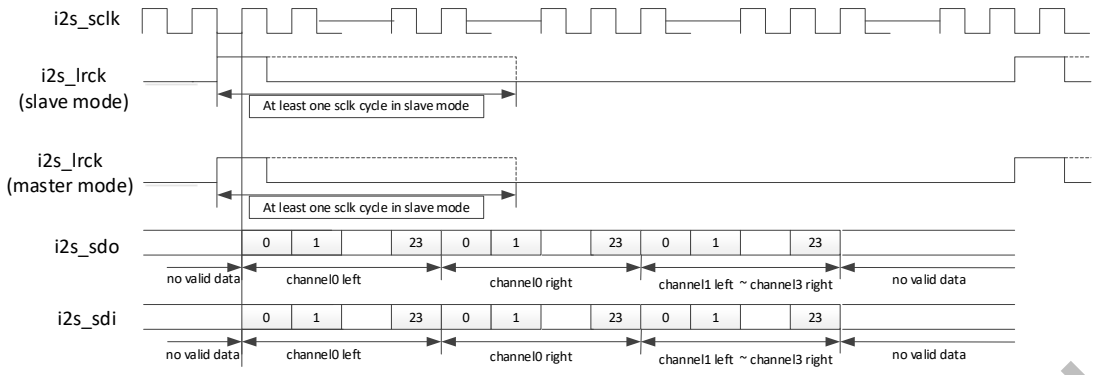


Fig. 16-14 TDM Left Shift Mode 2 Timing Format (PCM Format)

16.3.12 TDM Left Shift Mode3 (PCM Format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_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.

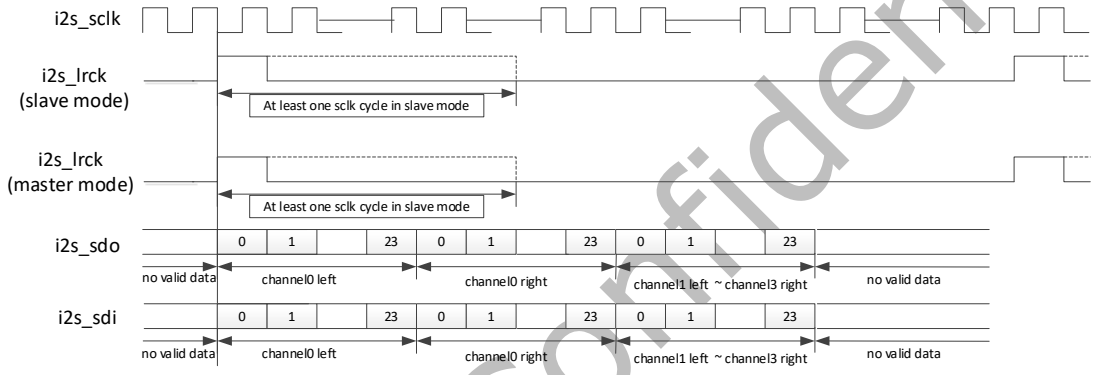
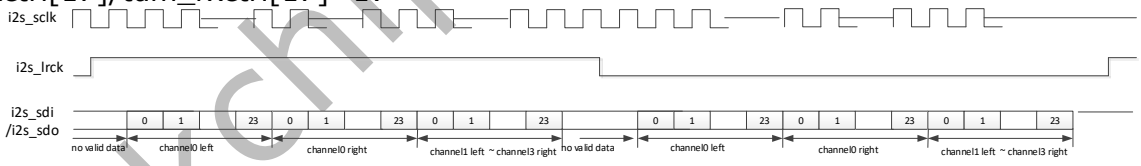


Fig. 16-15 TDM Left Shift Mode 3 Timing Format (PCM Format)

16.3.13 TDM Normal Mode (I2S Format)

This is the waveform of I2S normal mode. For SD (i2s_sdo, i2s_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:

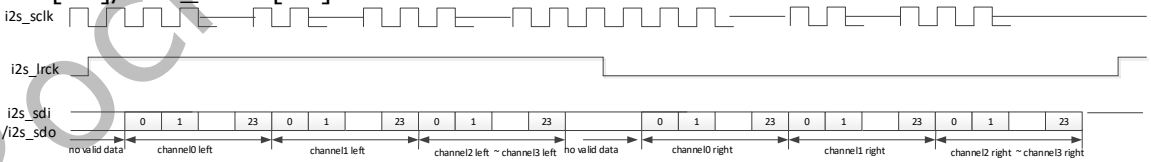
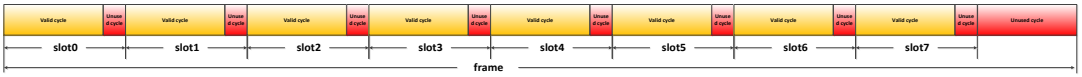


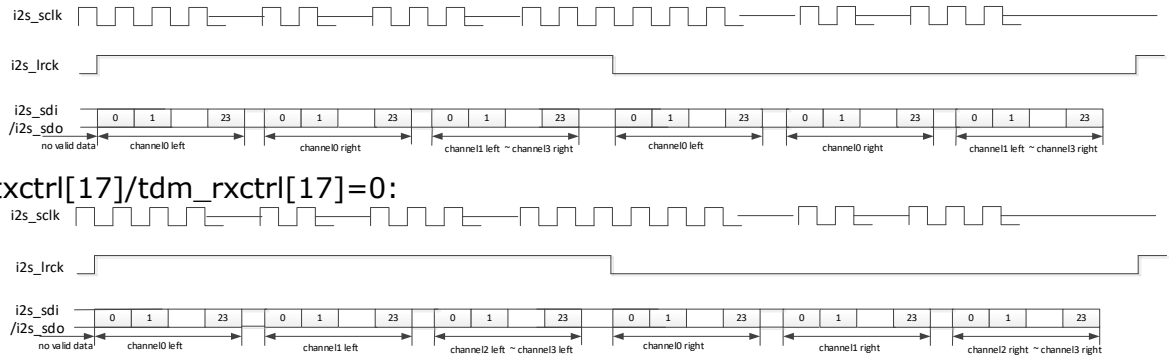
Fig. 16-16 TDM Normal Mode Timing Format (I2S Format)

16.3.14 TDM Left Justified Mode (I2S Format)

This is the waveform of I2S left justified mode. For SD (i2s_sdo, i2s_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.



tdm_txctrl[17]/tdm_rxctrl[17]=1:



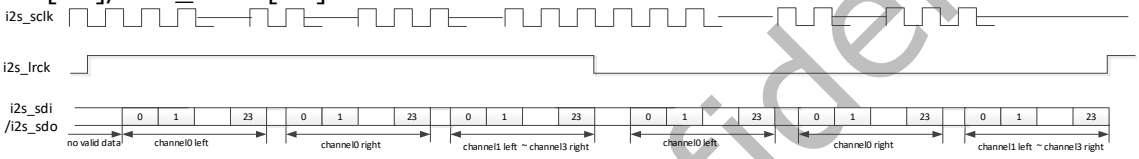
tdm_txctrl[17]/tdm_rxctrl[17]=0:

Fig. 16-17 TDM Left Justified Mode Timing Format (I2S Format)

16.3.15 TDM Right Justified Mode (I2S Format)

This is the waveform of I2S right justified mode. For SD (i2s_sdo, i2s_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.

tdm_txctrl[17]/tdm_rxctrl[17]=1:



tdm_txctrl[17]/tdm_rxctrl[17]=0:

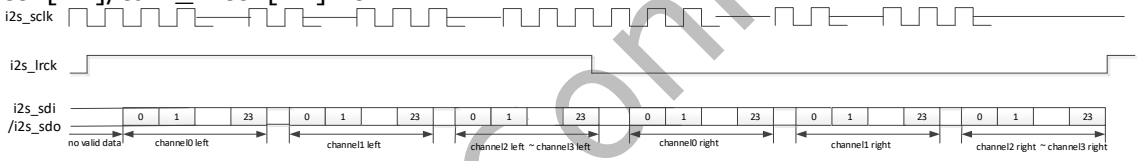


Fig. 16-18 TDM Right Justified Mode Timing Format (I2S Format)

16.4 Register Description

16.4.1 Internal Address Mapping

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

16.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TDM_8CH_TXCR	0x0000	W	0x7200000F	Transmit Operation Control Register
I2S_TDM_8CH_RXCR	0x0004	W	0x01C8000F	Receive Operation Control Register
I2S_TDM_8CH_CKR	0x0008	W	0x00001F1F	Clock Generation Register
I2S_TDM_8CH_TXFIFOLR	0x000C	W	0x00000000	TX FIFO Level Register
I2S_TDM_8CH_DMACR	0x0010	W	0x001F0000	DMA Control Register
I2S_TDM_8CH_INTCR	0x0014	W	0x01F00000	Interrupt Control Register
I2S_TDM_8CH_INTSR	0x0018	W	0x00000000	Interrupt Status Register
I2S_TDM_8CH_XFER	0x001C	W	0x00000000	Transfer Start Register
I2S_TDM_8CH_CLR	0x0020	W	0x00000000	Sclk Domain Logic Clear Register
I2S_TDM_8CH_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_TDM_8CH_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_TDM_8CH_RXFIFOLR	0x002C	W	0x00000000	RX FIFO Level Register
I2S_TDM_8CH_TDM_TXCTRL	0x0030	W	0x00003EFF	TDM Mode Transmit Operation Control Register

Name	Offset	Size	Reset Value	Description
I2S_TDM_8CH_TDM_RXCTRL	0x0034	W	0x00003EFF	TDM Mode Receive Operation Control Register
I2S_TDM_8CH_CLKDIV	0x0038	W	0x00000707	Clock Divider Register
I2S_TDM_8CH_VERSION	0x003C	W	0x20050001	Version Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

16.4.3 Detail Registers Description

I2S_TDM_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.
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. Note: Only function when TX TFS[1]=0.
16:15	RW	0x0	tcsr Transmit channel select register 2'b00: Two channel 2'b01: Four channel 2'b10: Six channel 2'b11: Eight channel

Bit	Attr	Reset Value	Description
14	RW	0x0	hwt Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	sjm Store justified mode (Can be written only when XFER[0] bit is 0.) 1'b0: Right justified 1'b1: Left justified 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 contains two 16bit data and 32 bit space is full, it is impossible to choose justified mode.
11	RW	0x0	fbm First Bit Mode (Can be written only when XFER[0] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	ibm I2S bus mode (Can be written only when XFER[0] bit is 0.) 2'b00: I2S normal 2'b01: I2S left justified 2'b10: I2S right justified 2'b11: Reserved
8:7	RW	0x0	pbm (Can be written only when XFER[0] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode Note: Function when TX TFS[1:0] is 1.
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 Valid data width (Can be written only when XFER[0] bit is 0.) 5'h00~5'b0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1c: 29bit 5'h1d: 30bit 5'h1e: 31bit 5'h1f: 32bit

I2S_TDM_8CH_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:23	RW	0x3	rx_path_select3 Rx path select 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.
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 2'b11: Path0 data from sdi3
16:15	RW	0x0	rcsr Receive channel select register 2'b00: Two channel 2'b01: Four channel 2'b10: Six channel 2'b11: Eight channel
14	RW	0x0	hwt Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	sjm Store justified mode (Can be written only when XFER[1] bit is 0.) 1'b0: Right justified 1'b1: Left justified 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 contains two 16bit data and 32 bit space is full, it is impossible to choose justified mode.

Bit	Attr	Reset Value	Description
11	RW	0x0	fbm First bit mode (Can be written only when XFER[1] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	ibm I2S bus mode (Can be written only when XFER[1] bit is 0.) 2'b00: I2S normal 2'b01: I2S left justified 2'b10: I2S right justified 2'b11: Reserved
8:7	RW	0x0	pbm PCM bus mode (Can be written only when XFER[1] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
6:5	RW	0x0	tfs Transfer format select (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 Valid data width (Can be written only when XFER[1] bit is 0.) 5'h00~5'b0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1c: 29bit 5'h1d: 30bit 5'h1e: 31bit 5'h1f: 32bit

I2S_TDM_8CH_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	lrck_commom TX and RX common use 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 Master/Slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: Master mode(sclk output) 1'b1: Slave mode(sclk input)

Bit	Attr	Reset Value	Description
26	RW	0x0	ckp Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: Sample data at posedge sclk and drive data at negedge sclk 1'b1: Sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	rlp Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: 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'b1: Opposite 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)
24	RW	0x0	tlp Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: 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'b1: Opposite 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	0x00	reserved
15:8	RW	0x1f	rsd Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (receive sclk divider/2)*2*frequency of rx_lrck Note: Function when RX TFS[1:0] is 2'b00 or 2'b01.
7:0	RW	0x1f	tsd Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (Transmit sclk divider/2)*2*frequency of tx_lrck Note: Function when TX TFS[1:0] is 2'b00 or 2'b01.

I2S_TDM_8CH_TXFIFOLR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:18	RW	0x00	tfl3 Transmit fifo3 level Contains the number of valid data entries in the transmit fifo3.
17:12	RW	0x00	tfl2 Transmit fifo2 level Contains the number of valid data entries in the transmit FIFO2.
11:6	RW	0x00	tfl1 Transmit fifo1 level Contains the number of valid data entries in the transmit fifo1.

Bit	Attr	Reset Value	Description
5:0	RO	0x00	tfl0 Transmit fifo0 level Contains the number of valid data entries in the transmit fifo0.

I2S_TDM_8CH_DMACR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	rde Receive DMA enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled
23:21	RO	0x0	reserved
20:16	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.
15:9	RO	0x00	reserved
8	RW	0x0	tde Transmit DMA enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
7:5	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 TX FIFO(TX FIFO0 if CSR=00;TX FIFO1 if CSR=01,TX FIFO2 if CSR=10,TX FIFO3 if CSR=11)is equal to or below this field value.

I2S_TDM_8CH_INTCR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24:20	RW	0x1f	rft Receive fifo threshold 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 RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	rxoie RX overrun interrupt enable 1'b0: Disable 1'b1: Enable
16	RW	0x0	rxfie RX full interrupt enable 1'b0: Disable 1'b1: Enable
15:9	RO	0x00	reserved

Bit	Attr	Reset Value	Description
8:4	RW	0x00	tft Transmit fifo threshold 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	RW	0x0	txuic TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	txuie TX underrun interrupt enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	txeie TX empty interrupt enable 1'b0: Disable 1'b1: Enable

I2S_TDM_8CH_INTSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	rxoi RX overrun interrupt 1'b0: Inactive 1'b1: Active
16	RO	0x0	rxfi RX full interrupt 1'b0: Inactive 1'b1: Active
15:2	RO	0x0000	reserved
1	RO	0x0	txui TX underrun interrupt 1'b0: Inactive 1'b1: Active
0	RO	0x0	txei TX empty interrupt 1'b0: Inactive 1'b1: Active

I2S_TDM_8CH_XFER

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	rxch_sel 0: I2S normal mode 1: I2S left justified and right justified
3	RW	0x0	loopback_mode2 loopback en 1'b0: Stop loopback 1'b1: Start loopback
2	RW	0x0	loopback_mode1 loopback en 1'b0: Stop loopback 1'b1: Start loopback

Bit	Attr	Reset Value	Description
1	RW	0x0	rxs RX start bit 1'b0: Stop RX transfer 1'b1: Start RX transfer
0	RW	0x0	txs TX transfer start bit 1'b0: Stop TX transfer 1'b1: Start TX transfer

I2S_TDM_8CH_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	rxc RX logic clear This is a self-cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	txc TX logic clear This is a self-cleared bit. Write 1 to clear all transmit logic.

I2S_TDM_8CH_TXDR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdr Transmit fifo data register When it is written, data are moved into the transmit fifo.

I2S_TDM_8CH_RXDR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdr Receive fifo data register When the register is read, data in the receive fifo is accessed.

I2S_TDM_8CH_RXFIFOLR

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:18	RW	0x00	rfl3 Receive fifo3 level Contains the number of valid data entries in the receive fifo3.
17:12	RW	0x00	rfl2 Receive fifo2 level Contains the number of valid data entries in the receive fifo2.
11:6	RW	0x00	rfl1 Receive fifo1 level Contains the number of valid data entries in the receive fifo1.
5:0	RW	0x00	rfl0 Receive fifo0 level Contains the number of valid data entries in the receive fifo0.

I2S_TDM_8CH_TDM_TXCTRL

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved

Bit	Attr	Reset Value	Description
20:18	RW	0x0	tx_tdm_fsync_width_sel1 TDM transfer fsync width sel1 (Can be written only when XFER[0] is 0.) 3'b000: Single period of the sclk_tx. 3'b001: 2 period of the sclk_tx. n: n+1 period of the sclk_tx. 3'b110: 7 period of the sclk_tx. 3'b111: 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 TDM transfer fsync width sel0 (Can be written only when XFER[0] is 0.) 1'b0: 1/2 frame width. It should be set to an even number. 1'b1: Frame width.
16:14	RW	0x0	tdm_tx_shift_ctrl TDM transfer shift ctrl (Can be written only when XFER[0] is 0.) 3'b000: PCM format 0: Normal mode, drive data on the second negedge of sclk_tx after rising edge of TX LRCK. I2S format 0: Normal mode 3'b001: PCM format 1: 1/2 cycle shift left, drive data on second posedge of sclk_tx after rising edge of TX LRCK. I2S format 1: Left justified mode 3'b010: PCM format 2: 1 cycle shift left, drive data on first negedge of sclk_tx after rising edge of TX LRCK. I2S format 2: Right justified mode 3'b011: PCM format 3: 3/2 cycle shift left, drive data on first posedge of sclk_tx after rising edge of TX LRCK. I2S format: Not support 3'b100: PCM format 4: 2 cycle shift left, drive data aligned to the posedge of TX LRCK. 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 TDM transfer slot bits (Can be written only when XFER[0] is 0.) 5'h00~5'h0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1f: 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 TDM transfer frame width (Can be written only when XFER[0] is 0.) 9'h000~9'h01e: Reserved 9'h01f: 32bit 9'h020: 33bit 9'h021: 34bit 9'h022: 35bit ..... 9'h1ff: 512bit Note: Functional when TX TFS[1:0] is 2 or 3.

I2S_TDM_8CH_TDM_RXCTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:18	RW	0x0	tx_tdm_fsync_width_sel1 TDM receive fsync width sel1 (Can be written only when XFER[1] is 0.) 3'b000: Single period of the sclk_rx. 3'b001: 2 period of the sclk_rx. n: n+1 period of the sclk_rx. 3'b110: 7 period of the sclk_rx. 3'b111: 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 TDM receive fsync width sel0 (Can be written only when XFER[1] is 0.) 1'b0: 1/2 frame width. It should be set to an even number. 1'b1: Frame width
16:14	RW	0x0	tdm_rx_shift_ctrl TDM receive shift ctrl (Can be written only when XFER[1] is 0.) 3'b000: PCM format 0: Normal mode, sample data on the third posedge of sclk_rx after rising edge of RX LRCK. I2S format 0: Normal mode 3'b001: PCM format 1: 1/2 cycle shift left, sample data on second negedge of sclk_rx after rising edge of RX LRCK. I2S format 1: left justified mode 3'b010: PCM format 2: 1 cycle shift left, sample data on second posedge of sclk_rx after rising edge of RX LRCK. I2S format 2: Right justified mode 3'b011: PCM format 3: 3/2 cycle shift left, sample data on first negedge of sclk_rx after rising edge of RX LRCK. I2S format: Not support 3'b100: PCM format 4: 2 cycle shift left, sample data on the first posedge of sclk_rx after rising edge of RX LRCK. I2S format: Not support 3'b101~3'b111: Not support Note: Function when RX TFS[1:0] is 2 or 3.

Bit	Attr	Reset Value	Description
13:9	RW	0x1f	tdm_rx_slot_bit_width TDM receive slot bits (Can be written only when XFER[1] is 0.) 5'h00~5'h0e: Reserved 5'h0f: 16bit 5'h10: 17bit 5'h11: 18bit 5'h12: 19bit ..... 5'h1f: 32bit Note: Function when RX TFS[1:0] is 2 or 3.
8:0	RW	0x0ff	tdm_rx_frame_width TDM receive frame width (Can be written only when XFER[1] is 0.) 9'h000~9'h01e: Reserved 9'h01f: 32bit 9'h020: 33bit 9'h021: 34bit 9'h022: 35bit ..... 9'h1ff: 512bit Note: Functional when RX TFS[1:0] is 2 or 3.

I2S_TDM_8CH_CLKDIV

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x07	rx_mdiv RX mclk divider (Can be written only when XFER[1] bit is 0.) mclk_rx divider = (mclk_rx/sclk_rx)-1. For example, if mclk_rx divider is 5, then the frequency of sclk_rx is mclk_rx/6.
7:0	RW	0x07	tx_mdiv TX mclk fivider (Can be written only when XFER[0] bit is 0.) mclk_tx divider = (mclk_tx/sclk_tx)-1. For example, if mclk_tx divider is 5, then the frequency of sclk_tx is mclk_tx/6.

I2S_TDM_8CH_VERSION

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RO	0x20050001	i2s_version

16.5 Interface Description

The following table shows the I2S0 interface description.

Table 16-1 I2S0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_mclk	I/O	LCD_D10/I2S0_MCLK/SDMMC1_CLK_M0/GPIO2_A2_d	VO_IOC_GPIO2A_IOMUX_SEL_L[10:8]=3'h2
i2s0_sclk	I/O	I2C4_SDA_M0/UART1_CTSN_M1/LCD_D8/I2S0_SCLK/SDMMC1_D1_M0/GPIO2_A0_d	VO_IOC_GPIO2A_IOMUX_SEL_L[2:0]=3'h2

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_lrck	I/O	SDMMC1_D0_M0/I2S0_LRCK_M0/LCD_D9/UART1_RTSN_M1/I2C4_SCL_M0/VCCIO5/GPIO2_a1	VO_IOC_GPIO2A_IOMUX_SEL_L[6:4]=3'h2
i2s0_sdo0	O	UART1_TX_M1/LCD_D12/I2S0_SDO0/SDMMC1_D3_M0/GPIO2_A4_d	VO_IOC_GPIO2A_IOMUX_SEL_H[2:0]=3'h2
i2s0_sdo1	O	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VO_IOC_GPIO2A_IOMUX_SEL_H[14:12]=3'h2
i2s0_sdo2	O	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VO_IOC_GPIO2A_IOMUX_SEL_H[10:8]=3'h2
i2s0_sdo3	O	LCD_D11/I2S0_SDO3_SDI1/SDMMC1_CMD_M0/GPIO2_A3_d	VO_IOC_GPIO2A_IOMUX_SEL_L[14:12]=3'h2
i2s0_sdi0	I	UART1_RX_M1/LCD_D13/I2S0_SDI0/SDMMC1_D2_M0/GPIO2_A5_d	VO_IOC_GPIO2A_IOMUX_SEL_H[6:4]=3'h2
i2s0_sdi1	I	LCD_D11/I2S0_SDO3_SDI1/SDMMC1_CMD_M0/GPIO2_A3_d	VO_IOC_GPIO2A_IOMUX_SEL_L[14:12]=3'h2
i2s0_sdi2	I	FLASH_TRIG_OUT/I2C3_SCL_M0/PWM2_M1/LCD_D14/I2S0_SDO2_SDI2/UART0_RTSN_M1/GPIO2_A6_d	VO_IOC_GPIO2A_IOMUX_SEL_H[10:8]=3'h2
i2s0_sdi3	I	PRELIGHT_TRIG_OUT/I2C3_SDA_M0/PWM4_M1/LCD_D15/I2S0_SDO1_SDI3/UART0_CTSN_M1/GPIO2_A7_d	VO_IOC_GPIO2A_IOMUX_SEL_H[14:12]=3'h2

16.6 Application Notes

16.6.1 Software Application Notes

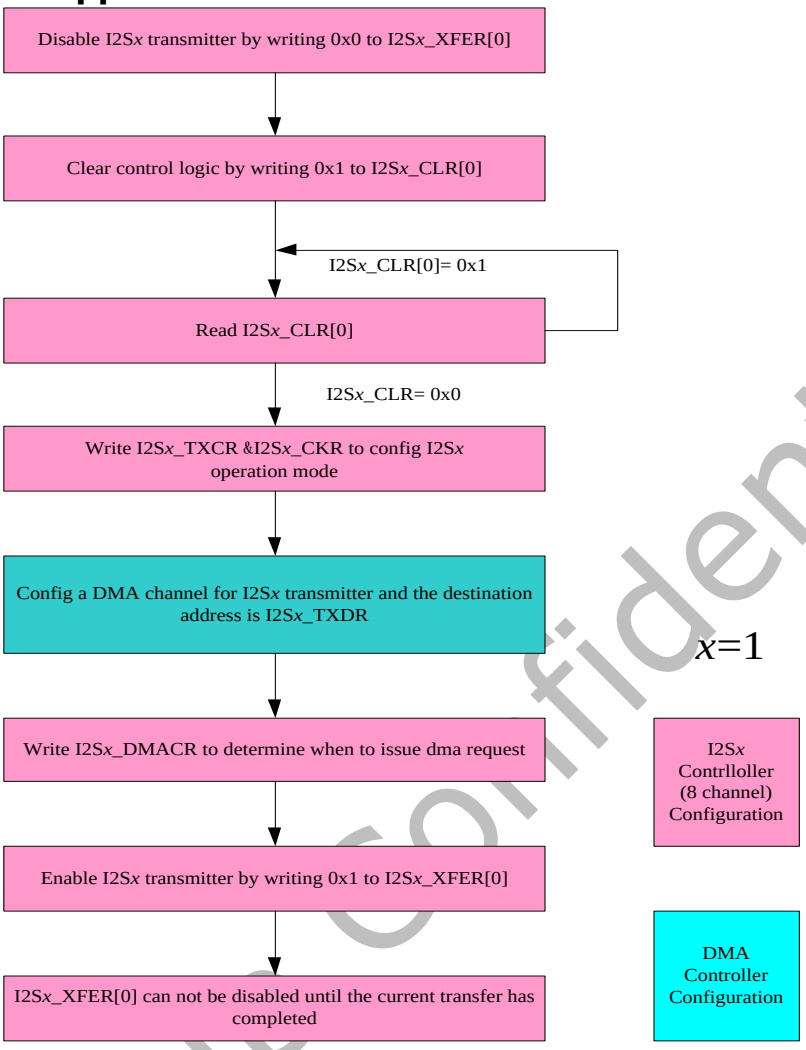


Fig. 16-19 I2S/PCM/TDM Controller Transmit Operation Flow Chart

Note: User should clear TX/RX logical by CLR[0]/CLR[1] and wait clear operation done before configure the other registers.

I2S/PCM/TDM controllers can support up to 16 channels when operate in TDM mode. Following configurations must be observed when in 16-channel TDM mode before starting sending or receiving audio data.

1. Set I2S_TDM_8CH_TXCR.TFS to 2'b11 or I2S_TDM_8CH_RXCR.TFS to 2'b11.
2. Set I2S_TDM_8CH_TDM_TXCTRL.TX_TDM_FSYNC_WIDTH_SEL0 to 1'b1 or I2S_TDM_8CH_TDM_RXCTRL.RX_TDM_FSYNC_WIDTH_SEL0 to 1'b1.
3. Set other registers properly.

Chapter 17 Digital Audio Codec

17.1 Overview

Digital Audio Codec is a 16-bit digital audio encoder which supports multiple sample rates. It is mainly composed of digital DAC. The aim of digital DAC is to process the data received from I2S/PCM interface through filters, volume control and modulation.
The Digital Audio Codec supports the following features.

- Support 8-bit APB bus slave interface
- Support 2-channel digital DAC
- Support 1-channel audio DSM format output
- Support I2S/PCM interface
- Support I2S/PCM master and slave mode
- Support 2-channel audio receiving in I2S mode
- Support 2-channel 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 configurable SCLK and LRCK polarity
- Support 16 bit sample resolution
- Support programmable left and right channel exchangeable in I2S mode and PCM mode
- Support three modes of mixing for every digital DAC channel
- Support volume control

Support programmable negative and positive volume gain

17.2 Block Diagram

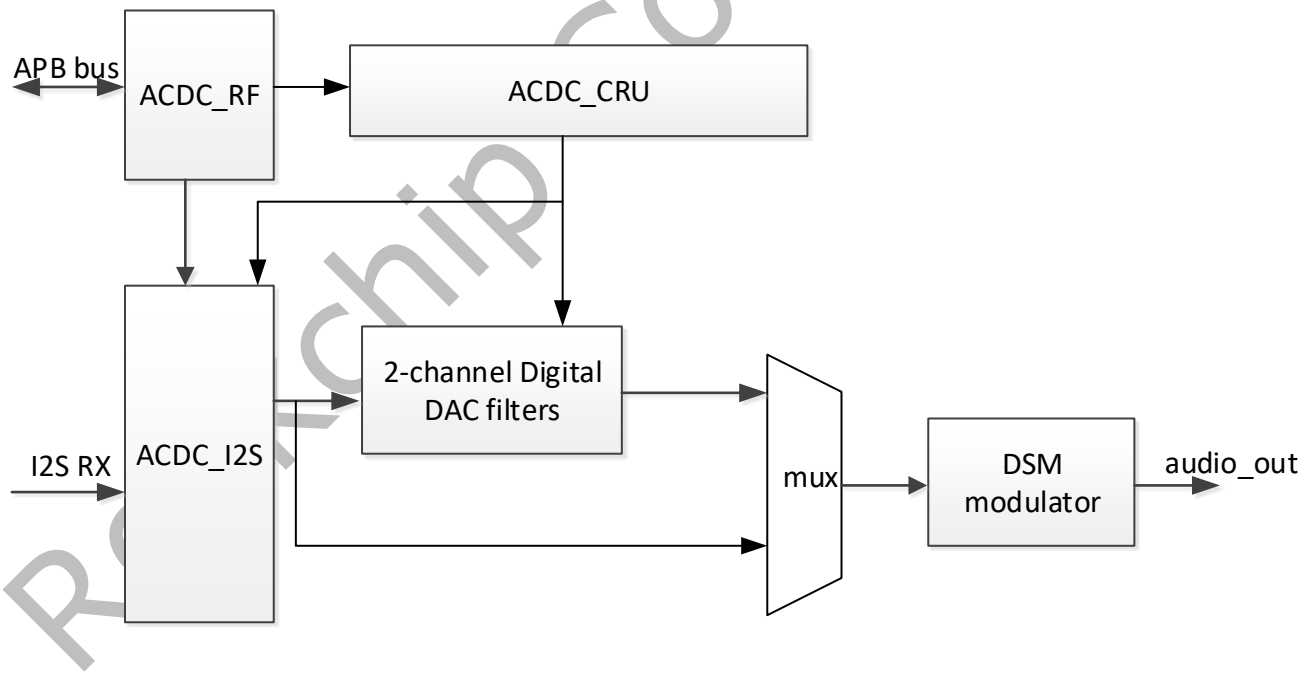


Fig. 17-1 Digital Audio Codec Block Diagram

ACDC_RF
The ACDC_RF implements the 8-bit APB slave operation through APB bus. It is responsible for configuring the operation registers of other modules.
ACDC_CRU
The ACDC_CRU implements clock and reset generation function. It is responsible for generating sample clock, digital DAC operation clock and I2S operation clock.
Digital DAC
There are 2 digital DAC channels. The digital DAC receives audio data from ACDC_I2S

module. It includes one CIC filter, one high-pass filter, several low-pass decimation filters and other audio signal processing related modules. The output of digital DAC is sent to DSM modular before transmitting outside.

ACDC_I2S

The I2S/PCM audio interface can be configured to master mode or slave mode. In Master Mode, SCLK and LRCK are configured as output. In slave mode, SCLK and LRCK are configured as input. The ACDC_I2S module can only operate in RX mode. When in RX mode, it receives audio data from I2S RX interface and sends it to digital DAC.

DSM Modulator

It is a modulator that converts 16 bits audio sample data to 1 bit audio data.

17.3 Function description

The I2S/PCM interface of Digital Audio Codec is connected to the I2S0 controller. Please refer to the I2S chapter for detailed information about I2S and PCM format that Digital Audio Codec supports.

17.3.1 Filters of Digital DAC

I2S module receives two-channel audio data from I2S RX interface and drives it to the digital DAC which supports mixing function or directly to the DSM modulator. How to pour 2-channel audio data into 2-channel digital DAC can be achieved by programming mixing mode.

The 2-channel digital DAC includes a high-pass filter and a maximum of 5 half-band filters. The high-pass filter is used to filter DC components in audio data stream. Result of high-pass filter is sent to the digital DAC volume control module. The input of 5 half-band filters comes from output of volume control module with each perform 2-times interpolation. But not all of them are working all the time. How many of them are needed to work depends on the sample rate of digital DAC. The result of half-band filters is sent to a modulator.

17.3.2 Volume Control

For digital DAC, output of high-pass filters is fed into volume control module. The volume control module inside digital DAC contains several sub-modules such as peak detect, frequency cross zero detect, LIMITER and digital gain control. It can be digitally attenuated over a range of -96dB~0dB in 0.375dB/step for negative gain and amplified over a range of 0dB~96dB in 0.375dB/step for positive gain. Whether is attenuated or amplified can be software programmed.

17.4 Register Description

17.4.1 Internal Address Mapping

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

17.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
ACDCDIG_DACVUCTL	0x0140	W	0x00000001	DAC Volume Control Register
ACDCDIG_DACVUCTIME	0x0144	W	0x00000000	DAC Volume Control Time Limit Register
ACDCDIG_DACDIGEN	0x0148	W	0x00000000	DAC Digital Enable Register
ACDCDIG_DACCLKCTRL	0x014C	W	0x00000000	DAC Clock Control Register
ACDCDIG_DACINT_DIV	0x0154	W	0x00000007	DAC Integer Clock Divider Register
ACDCDIG_DACSCLKRXINT_DIV	0x0160	W	0x0000001F	I2S SCLK RX Integer Divider Register
ACDCDIG_DACDSM_DIV	0x0164	W	0x00000003	DSM Mode Integer Divider Register

Name	Offset	Size	Reset Value	Description
ACDCDIG_DACDSM_CTRL	0x0168	W	0x00000000	DSM Mode Control Register
ACDCDIG_DACCFG1	0x0184	W	0x00000004	DAC Configure Register 1
ACDCDIG_DACMUTE	0x0188	W	0x00000000	DAC Mute Control Register
ACDCDIG_DACMUTEST	0x018C	W	0x00000000	DAC Mute Status Register
ACDCDIG_DACVOLL0	0x0190	W	0x00000000	Volume of DAC Left Channel 0 Register
ACDCDIG_DACVOLR0	0x01A0	W	0x00000000	Volume of DAC right Channel 1 Register
ACDCDIG_DACVOGP	0x01B0	W	0x00000000	DAC Volume Gain Polarity Register
ACDCDIG_DACRVOLL0	0x01B4	W	0x000000FF	Internal Volume of DAC Left Channel 0 Register
ACDCDIG_DACRVOLR0	0x01C4	W	0x000000FF	Internal Volume of DAC right Channel 1 Register
ACDCDIG_DACLMT0	0x01D4	W	0x00000000	DAC Limiter Register 0
ACDCDIG_DACLMT1	0x01D8	W	0x00000000	DAC Limiter Register 1
ACDCDIG_DACLMT2	0x01DC	W	0x00000000	DAC Limiter Register 2
ACDCDIG_DACMIXCTRLL	0x01E0	W	0x00000000	DAC Mixing Control Register Of Left Channels
ACDCDIG_DACMIXCTRLR	0x01E4	W	0x00000000	DAC Mixing Control Register Of right Channels
ACDCDIG_DACHPF	0x01E8	W	0x00000000	DAC High-pass Filter Control Register
ACDCDIG_I2S_RXCR0	0x030C	W	0x0000000F	Receive Operation Control Register 0
ACDCDIG_I2S_RXCR1	0x0310	W	0x00000000	Receive Operation Control Register 1
ACDCDIG_I2S_CKR0	0x0314	W	0x00000000	Clock Generation Register 0
ACDCDIG_I2S_CKR1	0x0318	W	0x00000000	Clock Generation Register 1
ACDCDIG_I2S_XFER	0x031C	W	0x00000000	Transfer Start Register
ACDCDIG_I2S_CLR	0x0320	W	0x00000000	SCLK Domain Logic Clear Register
ACDCDIG_VERSION	0x0380	W	0x00000002	Version Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

17.4.3 Detail Registers Description

ACDCDIG_DACVUCTL

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RW	0x0	dac_byps Digital DAC volume control bypass. 1'b0: Digital DAC volume control enable 1'b1: Digital DAC volume control bypass

Bit	Attr	Reset Value	Description
1	RW	0x0	dacfade Digital DAC volume adjust mode. 1'b0: Update to new volume immediately. 1'b1: Update volume as daczdt field describes.
0	RW	0x1	daczdt Digital DAC volume cross zero detect enable. It works when dac_byps is 1'b0 and dacfade is 1'b1. 1'b0: Volume adjusts every sample. 1'b1: Volume adjusts only when audio waveform crosses zero or volume control time limit condition meets.

ACDCDIG_DACVUCTIME

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	dacvuct Volume control time limit, valid only in fade cross zero mode. Time limit = dacvuct*(1/sample rate) Unit: Sample rate

ACDCDIG_DACDIGEN

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	dacglben Global enable of all Digital DAC channels. Only when dacglben and the enable signal corresponding to each Digital DAC channel is valid before starting work. 1'b0: Disable 1'b1: Enable
3:1	RO	0x0	reserved
0	RW	0x0	dacen_l0r1 Digital DAC left channel 0 and right channel 1 enable. 1'b0: Disable 1'b1: Enable

ACDCDIG_DACCLKCTRL

Address: Operational Base + offset (0x014C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	dac_cke Digital DAC operation clock enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
4	RW	0x0	i2srx_cke Clock enable of internal I2S RX channel. 1'b0: Disable 1'b1: Enable
3	RW	0x0	cke_bclkrx Clock enable of sclk_out_rx. 1'b0: Disable 1'b1: Enable
2	RW	0x0	dac_sync_ena Enable of the synchronization signal used internally generated by Digital DAC. 1'b0: Disable 1'b1: Enable
1	RO	0x0	dac_sync_status There is a counter to generate synchronization signal of Digital DAC. In order to ensure the integrity of synchronization signal, it is necessary to read back dac_sync_status to judge whether the counter stops working when dac_sync_ena is set from 1'b1 to 1'b0. If the signal is read back to 1'b0, it means that the counter stops working and the synchronization signal of Digital DAC is complete.
0	RO	0x0	reserved

ACDCDIG_DACINT_DIV

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x07	int_div_con Integer clock divider to provide 6.144/5.644/4.096MHz sample clock for internal filters. Make sure that int_div_con is an odd number between 7(8 times division) and 15(16 times division).

ACDCDIG_DACSCLKRXINT_DIV

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x1f	sckrxdiv Integer clock divider to generate sclk_out_rx when I2S RX works in master mode. It is ignored when in slave mode.

ACDCDIG_DACDSM_DIV

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x03	audio_dsm_div DSM mode division of Digital DAC's operation clock.

ACDCDIG_DACDSM_CTRL

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	dsm_mode_cke Clock enable of 1-bit DSM modulator 1'b0: Disable 1'b1: Enable
5:4	RW	0x0	dsm_mode Audio DSM mode selection 2'b01: Audio DSM mode 0. The input of 1-bit DSM modulator is from the last filter of Audio DAC. 2'b10: Audio DSM mode 1. The input of 1-bit DSM modulator is directly from output of I2S RX inside the ACDCDIG. Others: Reserved
3	RW	0x0	dsm_en 1-bit DSM modulator enable 1'b0: Disable 1'b1: Enable
2:0	RW	0x0	dith_sel Dith mode selection of 1-bit DSM modulator.

ACDCDIG_DACCFG1

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:2	RW	0x1	dacsrt Sample rates of Digital DAC when in audio DSM mode 0. This field is ignored when in audio DSM mode 1. 3'b000: 12kHz/11.024kHz/8kHz 3'b001: 24kHz/22.05kHz/16kHz 3'b010: 32kHz/48kHz/44.1kHz 3'b011: 96kHz/88.2kHz/64kHz 3'b100: 192kHz/176.4kHz/128kHz 3'b101~3'b111: Reserved
1:0	RO	0x0	reserved

ACDCDIG_DACMUTE

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	dacunmt 1'b0: DAC normal mode 1'b1: DAC unmute mode. In this mode, DAC volume control block will adjust volume to match the value in DACVOLL* and DACVOLR*.
0	RW	0x0	dacmt 1'b0: DAC normal mode 1'b1: DAC mute mode

ACDCDIG_DACMUTEST

Address: Operational Base + offset (0x018C)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x0	unmutest_l0r1 Unmute status for Digital DAC left channel 0 and right channel 1. When unmute is done, it indicates that internal volume is equal to the value programmed in DACVOLL* and DACVOLR*. 1'b0: Unmute not done 1'b1: Unmute done
3:1	RO	0x0	reserved
0	RO	0x0	mutest_l0r1 Mute status for Digital DAC left channel 0 and right channel 1. 1'b0: Not mute 1'b1: Mute

ACDCDIG_DACVOLL0

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	daclv0 Volume of Digital DAC left channel 0. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_DACVOLR0

Address: Operational Base + offset (0x01A0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	dacrv0 Volume of Digital DAC right channel 1. 0db~-95.625db, 0.375db/step. 8'h0: 0db 8'h1: -0.375db 8'h2: -0.75db 8'h3: -1.125db ..... 8'hff: -95.625db

ACDCDIG_DACVOGP

Address: Operational Base + offset (0x01B0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	volgpr1 Gain polarity for the volume of Digital DAC right channel 1. 1'b0: Negative gain 1'b1: Positive gain
0	RW	0x0	volgpl0 Gain polarity for the volume of Digital DAC left channel 0. 1'b0: Negative gain 1'b1: Positive gain

ACDCDIG_DACRVOLL0

Address: Operational Base + offset (0x01B4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvoll0 Internal real-time volume of Digital DAC left channel 0.

ACDCDIG_DACRVOLR0

Address: Operational Base + offset (0x01C4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0xff	rvolr0 Internal real-time volume of Digital DAC right channel 1.

ACDCDIG_DACLMT0

Address: Operational Base + offset (0x01D4)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	limen LIMITER enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	limdct Limiter detect mode. 1'b0: (Left channel + right channel)/2 1'b0: Left channel or right channel independently

ACDCDIG_DACLMT1

Address: Operational Base + offset (0x01D8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	atk_rate LIMITER attack rate=(power(2, atk_rate)*(8*clk)), clk is such as 4.096Mhz, 5.6448Mhz, 6.144Mhz.
3:0	RW	0x0	rls_rate LIMITER release rate=(power(2, rls_rate)*(8*clk)), clk is such as 4.096MHz, 5.6448MHz, 6.144MHz.

ACDCDIG_DACLMT2

Address: Operational Base + offset (0x01DC)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	max_lilmt The highest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step
3	RO	0x0	reserved
2:0	RW	0x0	min_lilmt The lowest threshold of LIMITER. 3'b000~3'b100: 0db~-12db, 3db/step 3'b101~3'b111: -18db~-30db, 6db/step

ACDCDIG_DACMIXCTRLL

Address: Operational Base + offset (0x01E0)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	mixmode_l0 Digital DAC left channel 0 mixing mode. 2'b00: Left channel 2'b01: Right channel 2'b10~2'b11: (Left channel + right channel)/2

ACDCDIG_DACMIXCTRLR

Address: Operational Base + offset (0x01E4)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	mixmode_r0 Digital DAC right channel 1 mixing mode. 2'b00: Left channel 2'b01: Right channel 2'b10~2'b11: (Left channel + right channel)/2

ACDCDIG_DACHPF

Address: Operational Base + offset (0x01E8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	hpfcf High-pass filter control. 2'b00: 80Hz 2'b01: 100Hz 2'b10: 120Hz 2'b11: 140Hz
3:1	RO	0x0	reserved
0	RW	0x0	hpfen_l0r1 High-pass filter enable for left channel 0 and right channel 1. 1'b0: High-pass filter is disabled. 1'b1: High-pass filter is enabled.

ACDCDIG_I2S_RXCR0

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	pbm PCM bus mode 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
5	RW	0x0	tfs Transfer format select 1'b0: I2S format 1'b1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	vdw Valid data width 5'b00000~5'b01110: Reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit ..... 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

ACDCDIG_I2S_RXCR1

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	rcsr Channel select 2'b00: Two channel Others: Reserved
5	RO	0x0	reserved
4	RW	0x0	cex Exchange left channel and right channel in the every receive line. 1'b0: Not exchange 1'b1: Exchange
3	RO	0x0	reserved
2	RW	0x0	fbm First bit mode 1'b0: MSB 1'b1: LSB
1:0	RW	0x0	ibm I2S bus mode 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: Reserved

ACDCDIG_I2S_CKR0

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	rsd I2S rx sclk divider for rx_lrck generator. 2'b00: 64 2'b01: 128 2'b10~2'b11: 256
1:0	RO	0x0	reserved

ACDCDIG_I2S_CKR1

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	mss Master/slave mode select 1'b0: Master mode(sclk output) 1'b1: Slave mode(sclk input)
2	RW	0x0	ckp Sclk polarity 1'b0: Sample data at posedge sclk and drive data at negedge sclk. 1'b1: Sample data at negedge sclk and drive data at posedge sclk.
1	RW	0x0	rlp 1'b0: 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'b1: Opposite 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)
0	RO	0x0	reserved

ACDCDIG_I2S_XFER

Address: Operational Base + offset (0x031C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	rxs 1'b0: Stop RX transfer 1'b1: Start RX transfer
0	RO	0x0	reserved

ACDCDIG_I2S_CLR

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	rxc This is a self-cleared bit. Write 1'b1 to clear all receive logic.
0	RO	0x0	reserved

ACDCDIG_VERSION

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x02	ver Version of ACDCDIG.

17.5 Interface Description

Table 17-1 Digital Audio Codec Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
AUDIO_DSM_P	O	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VENC_IOC_GPIO1D_IOMUXSEL_L[10:8]=3'h7
AUDIO_DSM_N	O	DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VENC_IOC_GPIO1D_IOMUXSEL_L[14:12]=3'h7

Notes: I=input, O=output, I/O=input/output, bidirectional

The I2S RX interface of Digital Audio Codec can only operates in slave mode if combined with I2S0 and ACODEC PHY for loopback application. In this situation, I2S0 acts as a master, Digital Audio Codec and ACODEC PHY act as slave. To enable connections between Digital Audio Codec and I2S0, GRF_PERI_CON1[8] should be set to 1'b1.

17.6 Application Notes

17.6.1 Software Application Notes

Steps to configure Digital DAC to work in audio DSM mode 0 are as follows.

1. Program CRU in the SOC system to achieve the operation frequency of Digital DAC.
2. Program ACDCDIG_DACINT_DIV to 0x07.
3. Program ACDCDIG_DACCLKCTRL to 0x3d.
4. Program ACDCDIG_DACSCLKRXINT_DIV if Digital DAC I2S RX acts as master.
5. Program ACDCDIG_DACDSM_CTRL to 0x58.
6. Program ACDCDIG_I2S_CKR0 and ACDCDIG_I2S_CKR1 to set I2S RX related registers.
7. Program ACDCDIG_I2S_CLR to clear RX logic.
8. Program ACDCDIG_I2S_RXCR0 and ACDCDIG_I2S_RXCR1.
9. Program GRF_PERI_CON1[8] to 1'b1 and program registers of I2S0 that is connected to Digital Audio Codec. Don't start transfer at this time.
10. Program DAC related registers such as ACDCDIG_DACHPF, ACDCDIG_DACVUCTL and ACDCDIG_DACCFG1 to achieve basic configuration.
11. Program ACDCDIG_I2S_XFER to start I2S RX.
12. Program registers of I2S0 outside Digital Audio Codec to start TX.
13. Program ACDCDIG_DACDIGEN to enable digital DAC channels. From now on, the Digital Audio Codec begins to work.

Steps to configure Digital Audio Codec to end audio DSM mode 0 transfer are as follows.

1. Program registers of I2S0 to stop TX and ACDCDIG_I2S_XFER to stop RX.
2. Program ACDCDIG_DACDSM_CTRL to 0x0.
3. Program ACDCDIG_DACDIGEN to disable digital DAC channels.
4. Program ACDCDIG_DACCLKCTRL to 0x0. Wait ACDCDIG_DACCLKCTRL.dac_sync_status until read back to be 1'b0.

Steps to configure Digital DAC to work in audio DSM mode 1 are as follows.

1. Program CRU in the SOC system to achieve the operation frequency of Digital DAC.
2. Program ACDCDIG_DACCLKCTRL to 0x3d.
3. Program ACDCDIG_DACSCLKRXINT_DIV if Digital DAC I2S RX acts as master.
4. Program ACDCDIG_DACDSM_CTRL to 0x68.
5. Program ACDCDIG_I2S_CKR0 and ACDCDIG_I2S_CKR1 to set I2S RX related registers.
6. Program ACDCDIG_I2S_CLR to clear RX logic.
7. Program ACDCDIG_I2S_RXCR0 and ACDCDIG_I2S_RXCR1.
8. Program GRF_PERI_CON1[8] to 1'b1 and program registers of I2S0 to start TX.
9. Program ACDCDIG_I2S_XFER to start I2S RX. From now on, the Digital Audio Codec begins to work.

Steps to configure Digital Audio Codec to end audio DSM mode 1 transfer are as follows.

1. Program registers of I2S0 to stop TX and ACDCDIG_I2S_XFER to stop RX.
2. Program ACDCDIG_DACDSM_CTRL to 0x0.
3. Program ACDCDIG_DACCLKCTRL to 0x0. Wait ACDCDIG_DACCLKCTRL.dac_sync_status until read back to be 1'b0

Chapter 18 Flexible Serial Peripheral Interface（FSPI）

18.1 Overview

The FSPI is a flexible serial peripheral interface host controller to interface with external device.

The FSPI supports the following features:

- Support various vendor devices with flexible command sequencer engine
  - Serial NOR Flash
  - Serial NAND Flash
  - Serial pSRAM
  - Serial SRAM
- Support SDR mode
- Support Single/Dual/Quad IO mode
- Support a 32-bit AHB slave to read and write controller register bank and initiate command sequence, including transfer data from/to external device indirectly
- Support a 32-bit AHB master with embedded DMA engine to transfer data from/to external device indirectly
- Support independent clock for system bus HCLK and controller core SCLK
- Support maskable interrupts generation
- Support sampling clock with optionally configurable delay line
- Support 1 CS# operation

18.2 Block Diagram

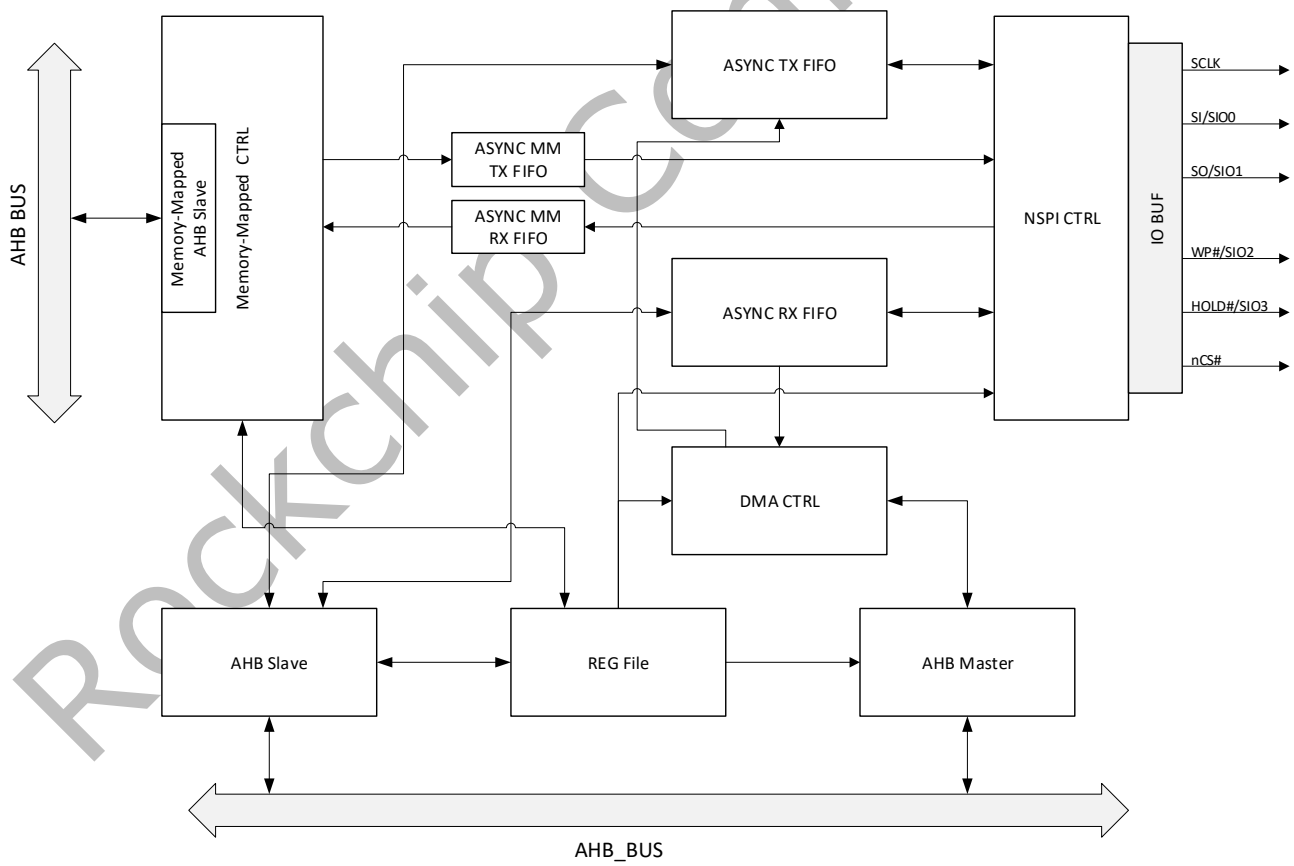


Fig. 18-1 FSPI Architecture

Memory-Mapped AHB slave is available when XIP feature is present. In current chip, this feature is not provided.

18.3 Function Description

18.3.1 AHB Slave

The AHB slave block is used to configure the register of controller to generate flexible command sequence, process the interrupt exception, target various device feature and AC timing specification. It is also used to write CMD/ADDR/DATA to TX FIFO and read DATA from RX FIFO which buffer the DATA from external device.

18.3.2 AHB Master

When the embedded DMA CTRL is used to transfer DATA, the AHB master is used to transfer data to other system region, such as internal SRAM, peripheral, external DRAM.

18.3.3 REG File

The REG File is configurable register bank to control the store the static configuration and dynamic status of controller.

18.3.4 DMA CTRL

A block takes responsible for splitting a long length transfer trans into AHB bus transaction and interfacing with ASYNC TX or RX FIFO.

18.3.5 FIFO

There are four FIFO in the FSPI controller. ASYNC TX FIFO and ASYNC RX FIFO is for normal transaction that initiated by command sequence driver. The ASYNC MM (Memory-Mapped) TX FIFO and ASYNC MM (Memory-Mapped) RX FIFO is only used to buffer DATA from or to external device initiated by system bus master directly.

18.3.6 NSPI CTRL

Sequence decode engine which generates specialized timing sequence for various device. The NSPI decode the transaction from TX FIFO and Memory-Mapped Controller and convert it to relative CMD/ADDR/DATA frame based on the configuration.

18.4 Register Description

18.4.1 Internal Address Mapping

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

Table 18-1 FSPI Address Mapping

Base Address	Device	Address Length	Offset Address Range
32'hFFAC_0000	FSPI CFG	64K BYTE	0x0000 ~ 0x0268

18.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
FSPI_CTRL0	0x0000	W	0x00000000	Control Register for CS0 Device
FSPI_IMR	0x0004	W	0x000001FF	Interrupt Mask Register
FSPI_ICLR	0x0008	W	0x00000000	Interrupt Clear Register
FSPI_FTLR	0x000C	W	0x00001010	FIFO Threshold Level Register
FSPI_RCVR	0x0010	W	0x00000000	FSPI Recover Register
FSPI_AX0	0x0014	W	0x00000000	FSPI Auxiliary Data Value for CS0 Device
FSPI_ABIT0	0x0018	W	0x00000000	Extend Address Bits for CS0 Device
FSPI_ISR	0x001C	W	0x00000000	Interrupt Status
FSPI_FSR	0x0020	W	0x00000001	FIFO Status Register
FSPI_SR	0x0024	W	0x00000000	FSPI Status Register
FSPI_RISR	0x0028	W	0x00000000	Raw Interrupt Status Register
FSPI_VER	0x002C	W	0x00000006	Version Register
FSPI_QOP	0x0030	W	0x00000000	Quad Line Operation IO Level Pre-set Register

Name	Offset	Size	Reset Value	Description
FSPI_EXT_CTRL	0x0034	W	0x00004023	Extend Control Register
FSPI_DLL_CTRL0	0x003C	W	0x00000001	Delay Line Control Register for CS0 Device
FSPI_EXT_AX	0x0044	W	0x0000F0FF	Extend Auxiliary Data Control Register
FSPI_SCLK_INATM_CNT	0x0048	W	0xFFFFFFFF	SCLK Inactive Timeout Counter
FSPI_XMMC_WCMD0	0x0050	W	0x00000000	Memory Mapped Control Write Command Register for CS0 Device
FSPI_XMMC_RCMD0	0x0054	W	0x00000000	Memory Mapped Control Read Command Register for CS0 Device
FSPI_XMMC_CTRL	0x0058	W	0x000072E0	Memory Mapped Control Register
FSPI_MODE	0x005C	W	0x00000000	Controller Working Mode Register
FSPI_DEVRGN	0x0060	W	0x00000017	Device Region Size Register
FSPI_DEVSIZE0	0x0064	W	0x00000012	Device Size Register for CS0 Device
FSPI_TME0	0x0068	W	0x00000000	Timeout Enable Control Register for CS0 Device
FSPI_XMMC_RX_WTMRK	0x0070	W	0x00000002	Memory Mapped Mode Receiver FIFO Water Mark Register
FSPI_DMATR	0x0080	W	0x00000000	DMA Trigger Register
FSPI_DMAADDR	0x0084	W	0x00000000	DMA Address Register
FSPI_LEN_CTRL	0x0088	W	0x00000000	Length Control Register
FSPI_LEN_EXT	0x008C	W	0x00000000	Length Extended Register
FSPI_XMMCSR	0x0094	W	0x00000000	Memory Mapped Status Register
FSPI_CMD	0x0100	W	0x00000000	Indirect Command Register
FSPI_ADDR	0x0104	W	0x00000000	Address Register
FSPI_DATA	0x0108	W	0x00000000	Data Register
FSPI_CTRL1	0x0200	W	0x00000000	Control Register for CS1 Device
FSPI_AX1	0x0214	W	0x00000000	FSPI Auxiliary Data Value for CS1 Device
FSPI_ABIT1	0x0218	W	0x00000000	Extend Address Bits for CS1 Device
FSPI_DLL_CTRL1	0x023C	W	0x00000001	Delay Line Control Register for CS1 Device
FSPI_XMMC_WCMD1	0x0250	W	0x00000000	Memory Mapped Control Write Command Register for CS1 Device
FSPI_XMMC_RCMD1	0x0254	W	0x00000000	Memory-Mapped Command Control Register for CS1 Device
FSPI_DEVSIZE1	0x0264	W	0x00000012	Device Size Register for CS1 Device
FSPI_TME1	0x0268	W	0x00000000	Timeout Enable Control Register for CS1 Device

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

18.4.3 Detail Registers Description

FSPI_CTRL0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	DATB Data Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
11:10	RW	0x0	ADRB Address Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
7:4	RW	0x0	IDLE_CYCLE Idle Cycles When Switching IO from Output to Input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE Shift Phase of Data Input in Controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".
0	RW	0x0	SPIM Serial Peripheral Interface Mode 1'b0: Mode 0 1'b1: Mode 3 SPIM is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.

FSPI_IMR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7	RW	0x1	DMAM DMA Finish Interrupt Mask 1'b0: DMA finish interrupt is not masked 1'b1: DMA finish interrupt is masked Only valid in indirect access mode.
6	RW	0x1	NSPIM NSPI Interrupt Mask 1'b0: NSPI interrupt is not masked 1'b1: NSPI interrupt is masked Valid in indirect access mode and memory mapped mode.
5	RW	0x1	AHBM AMBA AHB Error Interrupt Mask 1'b0: AMBA AHB Error interrupt is not masked 1'b1: AMBA AHB Error interrupt is masked Only valid in indirect access mode.
4	RW	0x1	TRANSM Transfer Finish Interrupt Mask 1'b0: Transfer finish interrupt is not masked 1'b1: Transfer finish interrupt is masked Only valid in indirect access mode.
3	RW	0x1	TXEM Transmit FIFO Empty Interrupt Mask 1'b0: Transmit FIFO empty interrupt is not masked 1'b1: Transmit FIFO empty interrupt is masked Only valid in indirect access mode.
2	RW	0x1	TXOM Transmit FIFO Overflow Interrupt Mask 1'b0: Transmit FIFO overflow interrupt is not masked 1'b1: Transmit FIFO overflow interrupt is masked Only valid in indirect access mode.
1	RW	0x1	RXUM Receive FIFO Underflow Interrupt Mask 1'b0: Receive FIFO underflow interrupt is not masked 1'b1: Receive FIFO underflow interrupt is masked Only valid in indirect access mode.
0	RW	0x1	RXFM Receive FIFO Full Interrupt Mask 1'b0: Receive FIFO full interrupt is not masked 1'b1: Receive FIFO full interrupt is masked Only valid in indirect access mode.

FSPI_ICLR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	W1C	0x0	DMAC DMA Finish Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the DMAS
6	W1C	0x0	NSPIC NSPI Error Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the NSPIS.

Bit	Attr	Reset Value	Description
5	W1C	0x0	AHBC AMBA AHB Error Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the AHBS.
4	W1C	0x0	TRANSC Transfer Finish Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TRANSS.
3	W1C	0x0	TXEC Transmit FIFO Empty Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXES.
2	W1C	0x0	TXOC Transmit FIFO Overflow Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXOS.
1	W1C	0x0	RXUC Receive FIFO Underflow Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXUS.
0	W1C	0x0	RXFC Receive FIFO Full Interrupt Clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXFS.

FSPI_FTLR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x10	RXFTLR Receive FIFO Threshold Level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ... 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	0x10	TXFTLR Transmit FIFO Threshold Level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ... When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

FSPI_RCVR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	R/W SC	0x0	RCVR FSPI Recover Write any values to trigger the recovery of SMC NSPI state machine, FIFO state and other logic state.

FSPI_AX0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	AX Auxiliary Data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".

FSPI_ABIT0

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RW	0x00	ABIT Address Bits Extend 5'h0: 1 bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.

FSPI_ISR

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
6	RO	0x0	NSPIS NSPI Transaction Decode Error Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
5	RO	0x0	AHBS AMBA AHB Error Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
4	RO	0x0	TRANSS Transfer Finish Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated

Bit	Attr	Reset Value	Description
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated
0	RW	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: No interrupt 1'b1: Active interrupt generated

FSPI_FSR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:16	RW	0x00	RXWLVL RX FIFO Water Level 5'h0: FIFO is empty 5'h1: 1 entry is taken ... 5'h10: 16 entry is taken, FIFO is full
15:13	RO	0x0	reserved
12:8	RO	0x00	TXWLVL TX FIFO Water Level 5'h0: FIFO is full 5'h1: 1 entry is left ... 5'h10: 16 entry is left, FIFO is empty
7:4	RO	0x0	reserved
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

FSPI_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	SR Status Register 1'b0: NSPI Controller is idle 1'b1: NSPI Controller is busy When controller is busy, don't change the setting of control register. When NSPI is idle, the software can initiate new transaction to external device.

FSPI_RISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
6	RO	0x0	NSPIS NSPI Error Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
5	RO	0x0	AHBS AMBA AHB Error Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
4	RO	0x0	TRANSS Transfer Finish Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.
0	RO	0x0	RXFS Receive FIFO Full Interrupt Status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.

FSPI_VER

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RO	0x0006	VER The Version ID of FSPI

FSPI_QOP

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	SO123BP SO123 Bypass Mode 1'b0: Disable bypass 1'b1: Enable bypass Default is enabled.
0	RW	0x0	SO123 D1/D2/D3 Data Value During Inactive When CS is Active 1'b0: Set to "0" 1'b1: Set to "1" The value of SO1, SO2 and SO3 during command and address bits input.

FSPI_EXT_CTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:15	RO	0x00000	reserved
14	RW	0x1	SR_GEN_MODE Status Register Generation Mode 1'b0: Compatible mode with old controller 1'b1: Robust generation to indicates the status of controller If set to "1", the controller will only clear the SR bit after operation sequence done and CS is high.
13	RW	0x0	TRANS_INT_MODE Transmit Done Interrupt Generation Mode 1'b0: Trigger NSPI end in data done 1'b1: Trigger NSPI end in CS inactive Default Generation is compatible with old controller.
12	RO	0x0	reserved
11:8	RW	0x0	SWITCH_IO_O2I_CNT Switch IO Attribute Cycles in O2I Idle Phase 4'h0: 1st cycle 4'h1: 2nd cycle 4'h2: 3rd cycle 4'h3: 4th cycle ... 4'hf: 16th cycle The target cycle when switching from output to input in O2I idle phase.

Bit	Attr	Reset Value	Description
7:4	RW	0x2	SWITCH_IO_DUMM_CNT Switch IO Attribute Cycles in Dummy Phase 4'h0: 1st cycle 4'h1: 2nd cycle 4'h2: 3rd cycle 4'h3: 4th cycle ... 4'hf: 16th cycle The target cycle when switching from output to input in Dummy data phase.
3:0	RW	0x3	CS_DESEL_CTRL CS Inactive Control 4'h0: 1 cycle 4'h1: 2 cycles 4'h2: 3 cycles 4'h3: 4 cycles ... 4'hf: 16 cycles The target cycles to hold CS inactive after de-assert the CS. Default value are 4 SCLK cycles that is enough for normal device.

FSPI_DLL_CTRL0

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	SCLK_SMP_SEL SCLK Sampling Selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
14:9	RO	0x00	reserved
8:0	RW	0x001	SMP_DLL_CFG Sample Delay Line Configuration 9'h0: 1 DLL element cell 9'h1: 1 DLL element cell 9'h2: 2 DLL element cells ... 9'h1ff: 511 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI_EXT_AX

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0xf0	AX_SETUP_PAT Auxiliary Setup Data Pattern The AX data pattern that setup the continuous/enhance/XIP read mode
7:0	RW	0xff	AX_CANCEL_PAT Auxiliary Cancel Data Pattern The AX data pattern that cancel the continuous/enhance/XIP read mode.

FSPI_SCLK_INATM_CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	SCLK_INATM_CNT SCLK Inactive Timeout Counter When CS is active and SCLK is hold in high or low due to TX FIFO is empty or RX FIFO is full, if SCLK_INATM_EN is enabled, and timeout occurs, the controller will go back to idle and RX FIFO is flushed.

FSPI_XMMC_WCMD0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_XMMC_RCMD0

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.

Bit	Attr	Reset Value	Description
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_XMMC_CTRL

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x1	PFT_EN Prefetch Enable 1'b0: Disable 1'b1: Enable Should disable prefetch if controller communicate with pSRAM which need refresh.
5	RW	0x1	DEV_HWEN Device AMBA AHB HWRITE Enable 1'b0: Disable 1'b1: Enable
4:0	RO	0x00	reserved

FSPI_MODE

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	XMMC_MODE_EN Memory Mapped Mode Enable 1'b0: Disable, indirect access mode 1'b1: Enable, Memory-Mapped mode Before switching from indirect access mode to Memory-Mapped mode, the application should make sure the controller is in idle state and no pending transaction. If switch from Memory-Mapped to indirect access mode, software should initiate a dummy read by CPU before that.

FSPI_DEVRGN

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:8	RW	0x0	DEC_CTRL Decode Control 2'b00: 1 CS# 2'b01: 2 CS# 2'b10: 4 CS# 2'b11: Reserved Only valid in XMMC mode.
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x17	RSIZE Region Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G bytes In Memory-Mapped mode, the CS is controlled by AHB address bus, region size is used to generate CS.

FSPI_DEVSIZE0

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RW	0x12	DSIZE Device Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G byte Device size is used to generate slop over status.

FSPI_TME0

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	SCLK_INATM_EN SCLK Inactive Timeout Enable 1'b0: Disable 1'b1: Enable
0	RO	0x0	reserved

FSPI_XMMC_RX_WTMRK

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x02	RX_FULL_WTMRK Memory Mapped Mode Receiver FIFO Water Mark. Default is enough.

FSPI_DMATR

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	W1C	0x0	DMATR DMA Trigger Write "1" to start the DMA transfer.

FSPI_DMAADDR

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAADDR DMA Address The destination or source data address in current system.

FSPI_LEN_CTRL

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	TRB_SEL Total Transfer Bytes Selection 1'b0: TRB controlled by CMD[TRB] 1'b1: TRB controlled by LEN_EXT

FSPI_LEN_EXT

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TRB_EXT Total Transfer Bytes Extended 32'd0: No data 32'd1: 1 Byte 32'd2: 2 Bytes ... Total data bytes number that will write to /read from the device.

FSPI_XMMCSR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	R/WSC	0x0	SLOPOVER1 Slop Over Register for CS1 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.
0	R/WSC	0x0	SLOPOVER0 Slop Over Register for CS0 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.

FSPI_CMD

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	WO	0x0	CS Device Chip Select. 2'b00: Chip select 0 2'b01: Chip select 1 2'b10: Reserved 2'b11: Reserved
29:16	WO	0x0000	TRB Total Transfer Bytes 14'd0: No data 14'd1: 1 Byte 14'd2: 2 Bytes ... Total data bytes number that will write to or read from the device.
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in indirect access mode. If there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	WO	0x0	WR Write or Read 1'b0: Read 1'b1: Write
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in indirect access mode.
7:0	WO	0x00	CMD Command Command data in indirect access mode.

FSPI_ADDR

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ADDR Address Register Indirect access start address data for current command sequence.

FSPI_DATA

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DATA Data Register Device data read or write from/to device.

FSPI_CTRL1

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:12	RW	0x0	DATB Data Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
11:10	RW	0x0	ADRB Address Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command Line Width 2'b00: 1 bit, x1 mode 2'b01: 2 bits, x2 mode 2'b10: 4 bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
7:4	RW	0x0	IDLE_CYCLE Idle Cycles When Switching IO from Output to Input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE Shift Phase of Data Input in Controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".
0	RW	0x0	SPIM Serial Peripheral Interface Mode 1'b0: Mode 0 1'b1: Mode 3 SPIM is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.

FSPI_AX1

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	AX Auxiliary Data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".

FSPI_ABIT1

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:0	RW	0x00	ABIT Address Bits Extend 5'h0: 1 bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.

FSPI_DLL_CTRL1

Address: Operational Base + offset (0x023C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	SCLK_SMP_SEL SCLK sampling selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
14:9	RO	0x00	reserved
8:0	RW	0x001	SMP_DLL_CFG Sample Delay Line Configuration 9'h0: 1 DLL element cell 9'h1: 1 DLL element cell 9'h2: 2 DLL element cells ... 9'h1ff: 511 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI_XMMC_WCMD1

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.

Bit	Attr	Reset Value	Description
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_XMMC_RCMD1

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:14	WO	0x0	ADDRB Address Bits 2'b00: 0 bit 2'b01: 24 bits 2'b10: 32 bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy Cycles 4'h0: No dummy cycle ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI_DEVSIZE1

Address: Operational Base + offset (0x0264)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x12	DSIZE Device Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes ..... 5'd10: 1K bytes ..... 5'd20: 1M bytes ..... 5'd31: 4G bytes Device size is used to generate slop over status.

FSPI_TME1

Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	SCLK_INATM_EN SCLK Inactive Timeout Enable 1'b0: Disable 1'b1: Enable
0	RO	0x0	reserved

18.5 Interface Description

Table 18-2 FSPI Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
sfc_sclk	O	FSPI_CLK/EMMC_CLK/GPIO4_B1_d	VCCIO3_IOC_GPIO4B_IOMUX_SEL_L[6:4]=3'h2
sfc_csn	O	FSPI_CS0n/EMMC_CMD/GPIO4_B0_u	VCCIO3_IOC_GPIO4B_IOMUX_SEL_L[2:0]=3'h2
sfc_sio0	I/O	FSPI_D0/EMMC_D0/GPIO4_A4_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[2:0]=3'h2
sfc_sio1	I/O	FSPI_D1/EMMC_D1/GPIO4_A3_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[14:12]=3'h2
sfc_sio2	I/O	FSPI_D2/EMMC_D2/GPIO4_A2_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_L[10:8]=3'h2
sfc_sio3	I/O	FSPI_D3/EMMC_D3/GPIO4_A6_u	VCCIO3_IOC_GPIO4A_IOMUX_SEL_H[10:8]=3'h2

Notes: I=input, O=output, I/O=input/output, bidirectional

18.6 Application Notes

18.6.1 Typical Program Flow Without DMA

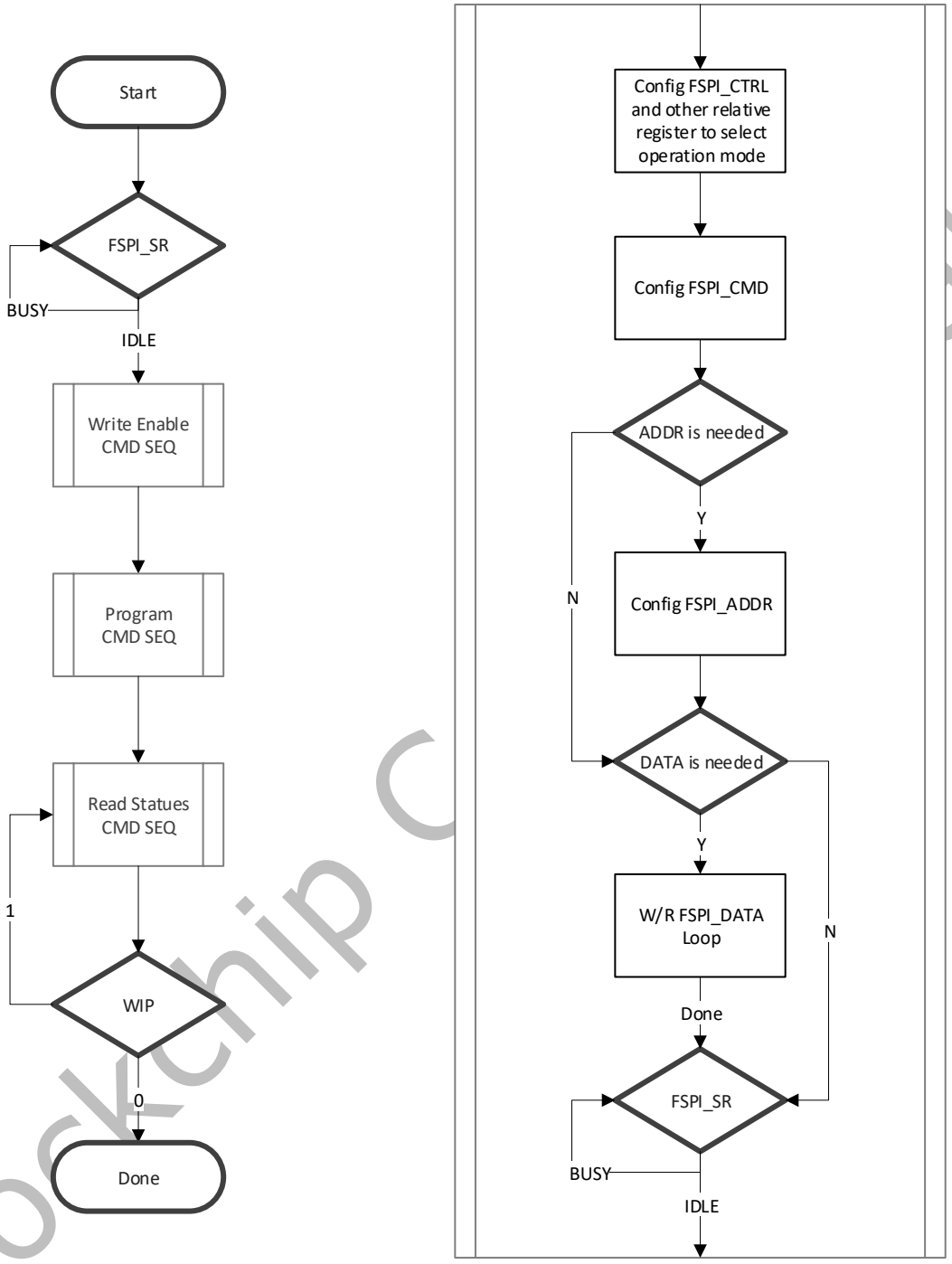


Fig. 18-2 Program Flow

All the AHB bus write data to FSPI_CMD, FSPI_ADDR and FSPI_DATA will be marked with different header and then pushed into transmit FIFO by writing order.

18.6.2 Typical READ Flow Without DMA

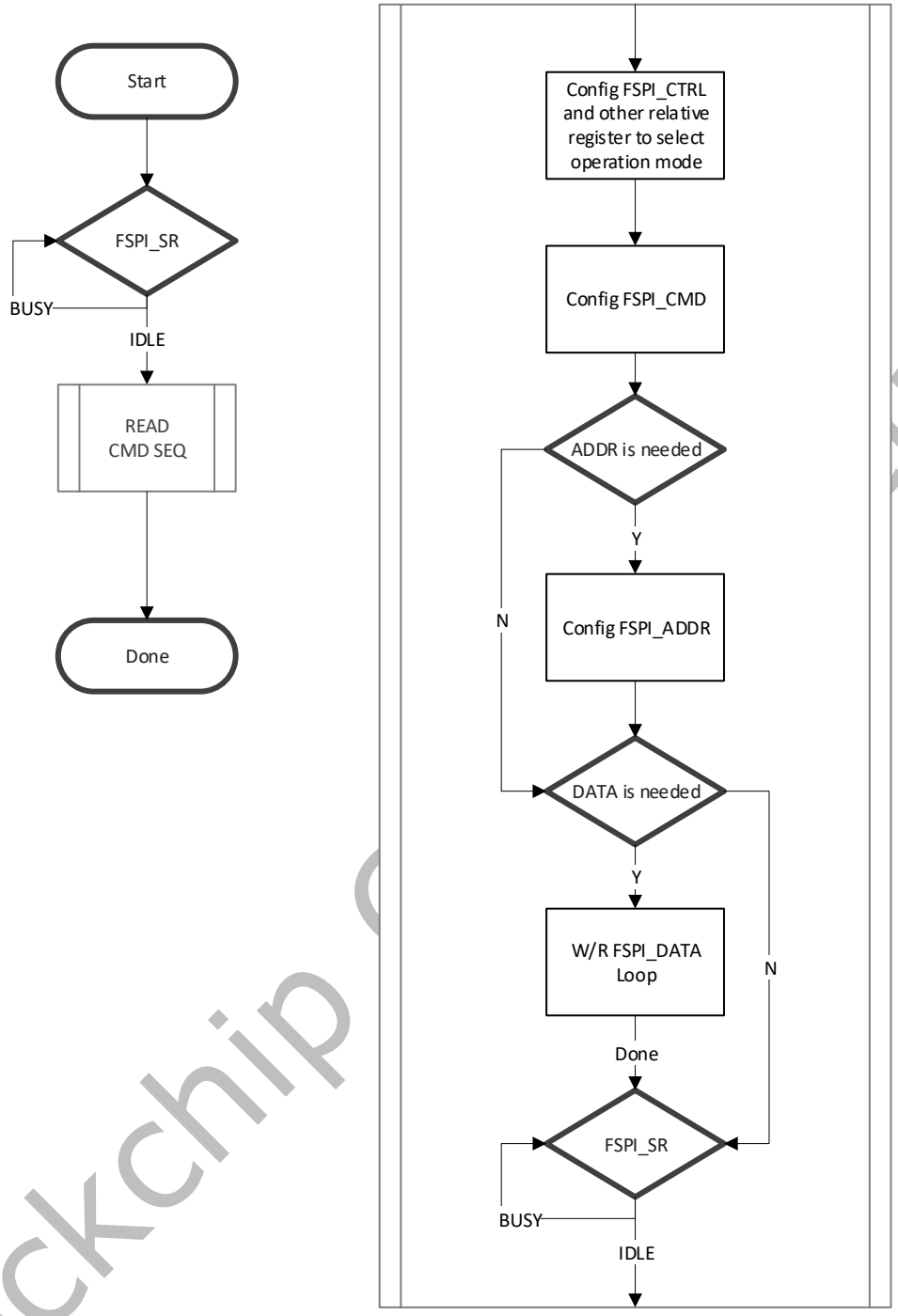


Fig. 18-3 Read Flow

18.6.3 Command Flow with DMA

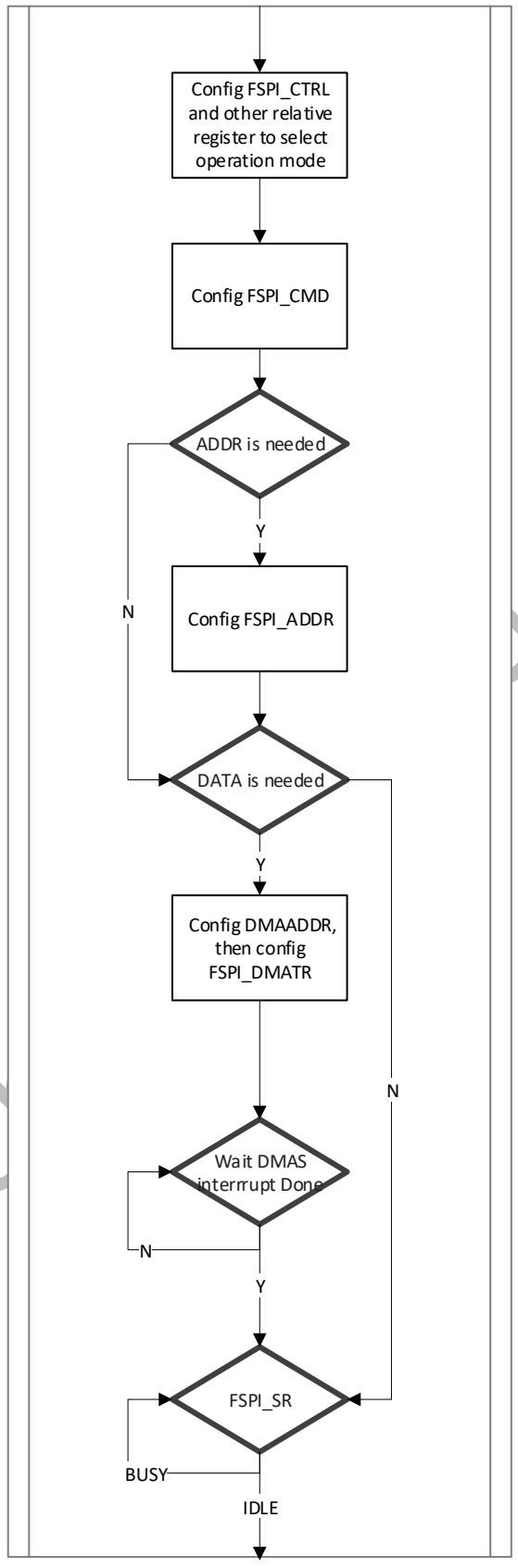


Fig. 18-4 Command with DMA Flow

The total transfer bytes is decided by TRB register in FSPI_CMD and must be aligned to 2 bytes.

18.6.4 SPI Mode and Sampling Phase Control

The register SPIM in FSPI_CTRL will decide the default value of SCLK when CS# is inactive. When SPIM=0, the default value is 0, means Mode 0. When SPIM=1, the default value is 1,

means SPI Mode 3.

The register SHIFTPAHSE in FSPI_CTRL will decide when to sample the SIO data. If SHIFTPAHSE=0, it will sample the data at the posedge of SCLK sampling clock. If SHIFTPHASE=1, it will sample the data at the negedge of SCLK sampling. The phase delay of sampling clock SCLK is configurable by SMP_DLL_CFG. It is strongly recommended that the SMP_DLL_CFG is set to 1 when SCLK_SMP_SEL is in bypass mode.

Individual FSPI_DLL_CTRL[n] allows flexibly control the DLL for each CSn channel, and the DLL is switched dynamically when the CSn is in operation.

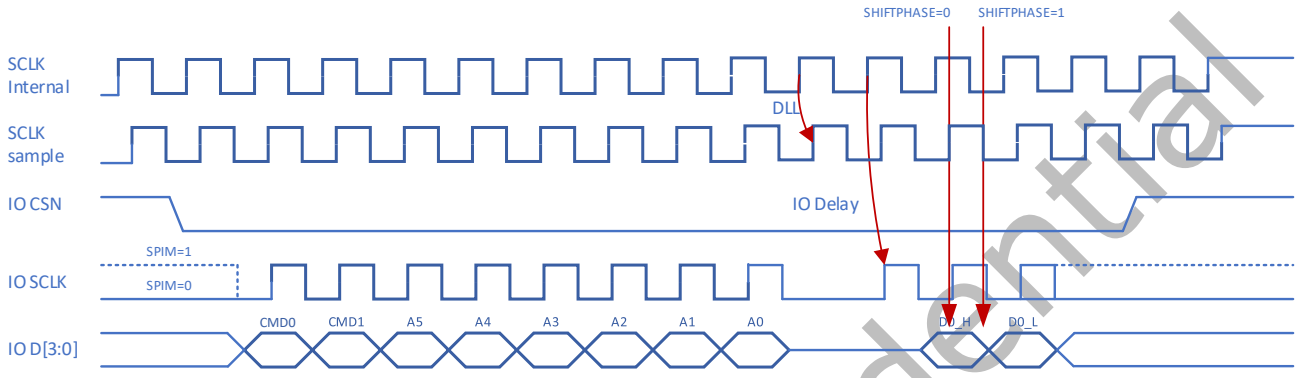


Fig. 18-5 SPI mode

18.6.5 Idle Cycles

The FSPI_CTRL register is a global control register, when the controller is in busy state (FSPI_SR), FSPI_CTRL cannot be set. The field IDLE_CYCLE (FSPI_CTRL[7:4]) of this register are used to configure the idle level cycles of FSPI core clock (SCLK) before reading the first bit of the read command.

Like the following picture shows, the highlighted line of the SCLK is the idle cycles, during these cycles, the chip pad is switched to output. When IDLE_CYCLE =0, it means there will be no idle level cycles.

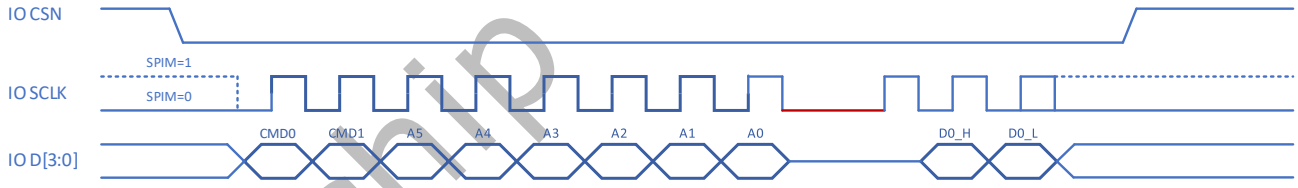


Fig. 18-6 Idle cycles

Chapter 19 Gigabit Media Access Controller (GMAC)

19.1 Overview

The Ethernet Quality-of-Service controller(EQOS is commonly referred to as GMAC in this document) provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) and Reduced Gigabit Media Independent Interface (RGMII) compliant Ethernet PHY. The GMAC 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.
The following features may or may not be present in the actual product. Please contact Rockchip for the actual feature configurations and the third-party licensing requirements.

19.1.1 MAC Features

19.1.1.1 MAC Tx and Rx Common Features

- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Half-duplex operation:
  - CSMA/CD Protocol support
  - Flow control using back-pressure support (based on implementation-specific white papers and UNH Ethernet Clause 4 MAC Test Suite - Annex D)
  - Packet bursting and packet extension in 1000 Mbps half-duplex operation
- 64-bit data transfer interface on the application side
- Full-duplex flow control operations (IEEE 802.3x Pause packets and Priority flow control)
- network statistics with RMON or MIB Counters (RFC2819/RFC2665)
- Media clock generation and recovery
- MDIO (Clause 22 and Clause 45) master interface for PHY device configuration and management

19.1.1.2 MAC Tx Features

- Preamble and start of packet data (SFD) insertion
- Separate 32-bit status for each packet transmitted from the application
- Automatic CRC and pad generation controllable on a per-packet basis
- Programmable packet length to support Standard or Jumbo Ethernet packets with up to 16 KB of size
- Programmable Inter Packet Gap (40–96 bit times in steps of 8)
- IEEE 802.3x Flow Control automatic transmission of zero-quanta Pause packet when flow control input transitions from assertion to de-assertion (in full-duplex mode)
- Source Address field insertion or replacement, and VLAN insertion, replacement, and deletion in transmitted packets with per-packet or static-global control
- Insertion, replacement, or deletion of up to two VLAN tags
- Frame Preemption for MAC Tx

19.1.1.3 MAC Rx Features

- Automatic Pad and CRC Stripping options
- Option to disable Automatic CRC checking
- Preamble and SFD deletion
- Separate 112-bit or 128-bit status
- Programmable watchdog timeout limit
- Flexible address filtering modes:
  - Up to 31 additional 48-bit perfect (DA) address filters with masks for each byte
  - Up to 96 additional 48-bit perfect (DA) address filters that can be selected in blocks of 32 and 64 registers
  - Up to 31 48-bit SA address comparison check with masks for each byte
  - 32 bit, 64 bit, 128 bit, or 256 bit Hash filter (optional) for multicast and unicast (DA) addresses

  - Option to pass all multi-cast addressed packets
  - Promiscuous mode to pass all packets without any filtering for network monitoring
  - Pass all incoming packets (as per filter) with a status report
- Additional packet filtering:
  - VLAN tag-based: Perfect match and Hash-based (optional) filtering. Filtering based on either outer or inner VLAN tag is possible
  - Layer 3 and Layer 4-based: TCP or UDP over IPv4 or IPv6
  - Extended VLAN tag based filtering 4, 8, 16, or 32 filter selection
- IEEE 802.1Q VLAN tag detection and option to delete the VLAN tags in received packets
- Optional module to detect remote wake-up packets and AMD magic packets
- Optional forwarding of received Pause packets to the application (in full-duplex mode)
- Frame Preemption for MAC Rx

19.1.2 MTL Features

19.1.2.1 MTL Tx and Rx Common Features

- 32-bit, 64-bit, or 128-bit Transaction Layer block (bridges the application and the MAC)
- Data transfers executed using simple FIFO protocol
- Synchronization for all clocks in the design (Transmit, Receive, and Application clocks)
- Optimization for packet-oriented transfers with packets delimiters
- Option to have dual-port RAM based asynchronous FIFO controllers or Single-port RAM based synchronous FIFO controllers
- RAM memory instantiation outside the top-level module to facilitate memory testing or instantiation
- Programmable burst length, up to half the size of the MTL Rx queue or Tx queue size, to support burst data transfer in the EQOS-MTL configuration
- Programmable threshold capability for each queue (default of 64 bytes)
- Optional Debug and slave mode operation on Queue 0 (default queue)

19.1.2.2 MTL Tx Features

- TX FIFO sizes on transmission is 16 KB
- Store-and-Forward mechanism or threshold mode (cut-through) for transmission to the MAC
- Programmable queue size in configurations with multiple queues. Each queue size can be programmed in terms of 256 bytes
- Automatic retransmission of collision packets in half-duplex mode
- Discard packets on late collision, excessive collisions, excessive deferral, and under-run conditions with appropriate status
- Disabling of Data Memory RAM chip-select when inactive to reduce power consumption
- Optional module to calculate and insert IPv4 header checksum and TCP, UDP, or ICMP checksum
- Programmable interrupt options for different operational conditions
- Statistics by generating pulses for packets dropped (because of underflow) in the Tx FIFO
- Optional packet-level control for
  - VLAN tag insertion or replacement
  - Ethernet source address insertion
  - Layer3/Layer4 Checksum insertion control
  - One-step timestamp
  - Timestamp control
  - CRC and pad control
- Following scheduling algorithms in configurations with multiple queues:
  - Weighted Round Robin (WRR)
  - (When Data Center Bridging is enabled) Deficit Weighted Round Robin (DWRR)
  - (When Data Center Bridging is enabled) Weighted Fair Queuing (WFQ)
  - Strict Priority (SP)
  - (When Audio-Video Bridging is enabled) Credit-based Shaper (CBS)
  - (When TSN is enabled), Enhancement to Scheduled Traffic (EST)

  - (When TSN is enabled), Time Based Scheduling (TBS)
- Option to support dropping of Tx Status to improve the Transmit throughput

19.1.2.3 MTL Rx Features

- Rx queue sizes in the Receive path is 32 KB
- Insertion of Rx Status vectors into the Rx queue after the EOP transfer (in Threshold mode) and before SOP (in Store-and-Forward mode) in EQOS-MTL configuration
- Programmable Rx queue threshold (default fixed at 64 bytes) in Threshold (or cut-through) mode
- Option to filter all error packets on reception and not forward them to the application in the store-and-forward mode
- Option to forward the undersized good packets
- Statistics by generating pulses for packets dropped (because of overflow) in the Rx FIFO
- Automatic generation of Pause packet control or backpressure signal to the MAC based on the Rx Queue fill level
- Arbitration among queues when multiple queues are present. The following arbitration schemes are supported:
  - Weighted Round Robin (WRR)
  - Weighted Strict priority (WSP)
  - Strict Priority (SP)
- Option to replicate received multicast packets for transfer by multiple Rx DMA channels
- Option to have a programmable lookup table based flexible Parser for filtering and steering the Rx packets

19.1.3 DMA Block Features

- 64-bit data transfers
- Separate DMA channel in the Transmit path for each queue in MTL
- Single or multiple DMA channels for any number of queues in MTL Receive path
- Fully synchronous design operating on a single application clock (except for CSR module, when a separate CSR clock is configured)
- Optimization for packet-oriented DMA transfers with packet delimiters
- Byte-aligned addressing for data buffer support
- Dual-buffer (ring) descriptor support
- Descriptor architecture to allow large blocks of data transfer with minimum CPU intervention (each descriptor can transfer up to 32 KB of data)
- Comprehensive status reporting for normal operation and transfers with errors
- Individual programmable burst length for Tx DMA and Rx DMA engines for optimal host bus utilization
- Programmable interrupt options for different operational conditions
- Per-packet Transmit or Receive Complete Interrupt control
- Round-robin or fixed-priority arbitration between the Receive and Transmit engines
- Start and Stop modes
- Separate ports for host CSR access and host data interface
- support for TCP Segmentation Offload (TSO) and UDP Segmentation Offload (USO)
- Selectable number of Tx DMA channels with TSO/USO feature enabled
- Routing of received packets to the DMA channels based on the DA or VLAN Priority in multi-channel DMA configurations
- Option to split the packet header (Layer 3 and Layer 4) and payload in a different buffers
- Time-sensitive conditional packet fetching from system memory by comparing the Slot Time or IEEE 1588 time information provided in the descriptor (useful for AV applications)
- Programmable control for Transmit Descriptor posted writes to improve the throughput
- Sideband signals to control starting and stopping of DMA channels

19.2 Block Diagram

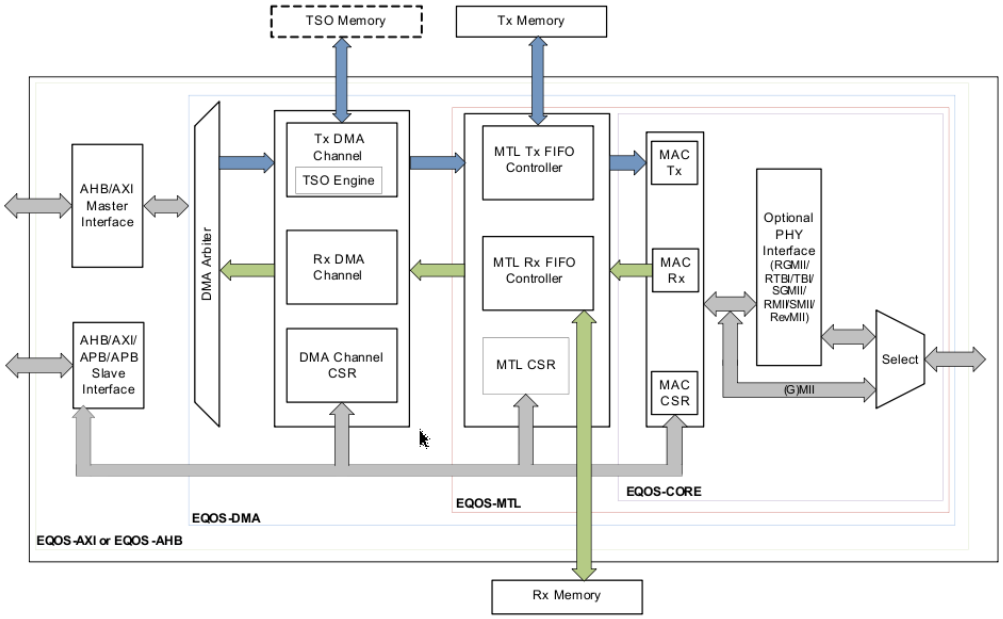


Fig. 19-1 GMAC Block Diagram

The GMAC 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 GMAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the GMAC subsystem's control and status registers (CSRs).

The GMAC only 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 GMAC 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 GMAC line clocks.

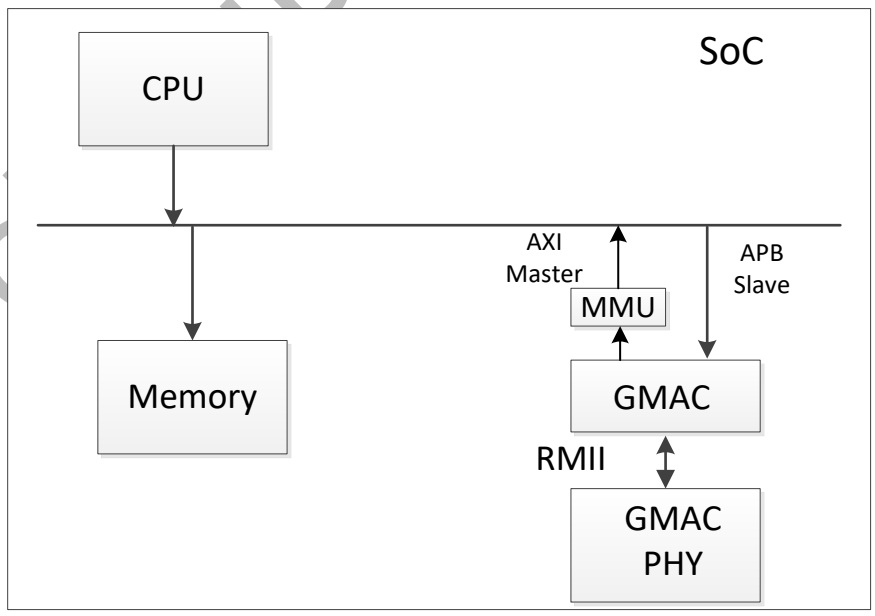


Fig. 19-2 GMAC in SOC

GMAC Supports 10/100-Mbps data transfer rates with the RMII interfaces.

19.3 Function Description

19.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig.1-3.

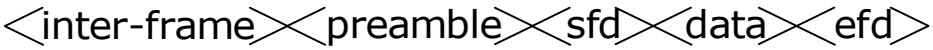


Fig. 19-3 Frame Format

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.

19.3.1.1 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 GMAC 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.

19.3.1.2 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).

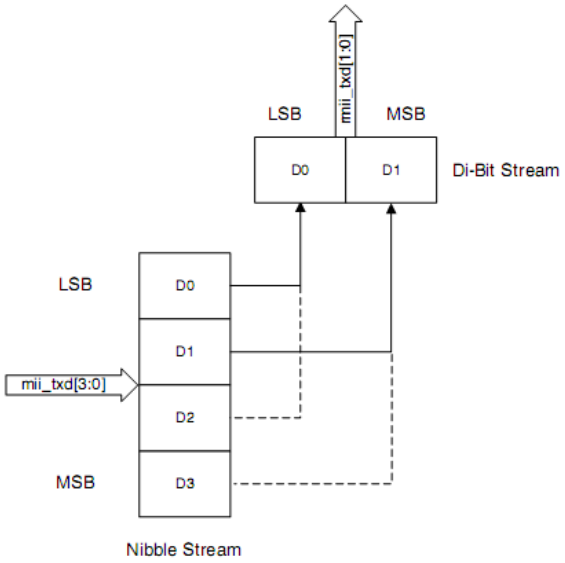


Fig. 19-4 RMII Transmission Bit Ordering

19.3.1.3 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.

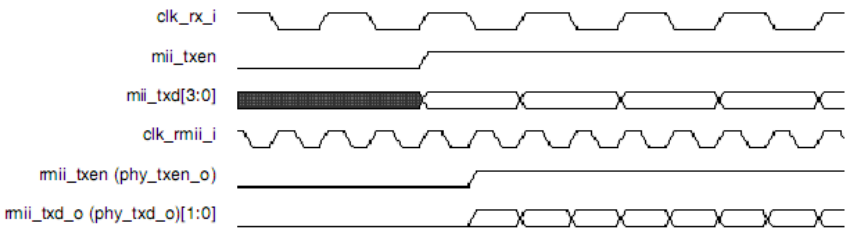


Fig. 19-5 Start of MII and RMII Transmission in 100-Mbps Mode

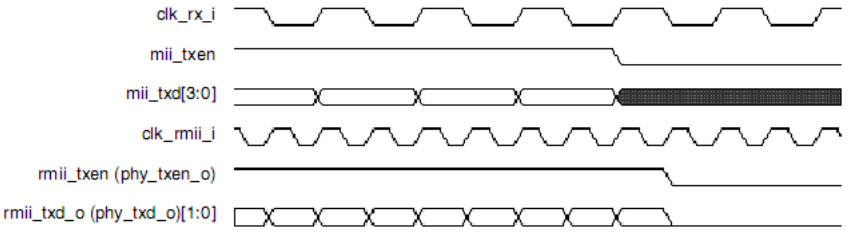


Fig. 19-6 End of MII and RMII Transmission in 100-Mbps Mode

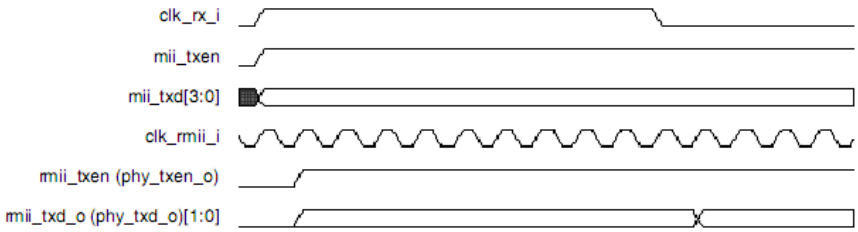


Fig. 19-7 Start of MII and RMII Transmission in 10-Mbps Mode

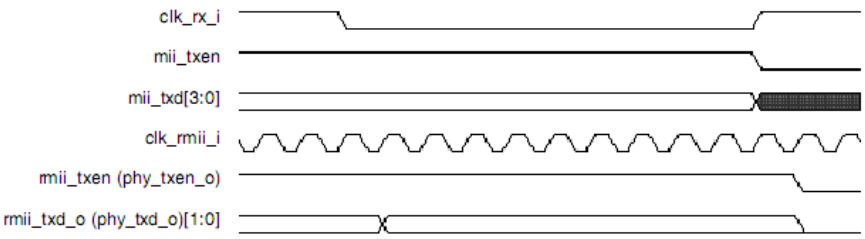


Fig. 19-8 End of MII and RMII Transmission in 10-Mbps Mode

19.3.1.4 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).

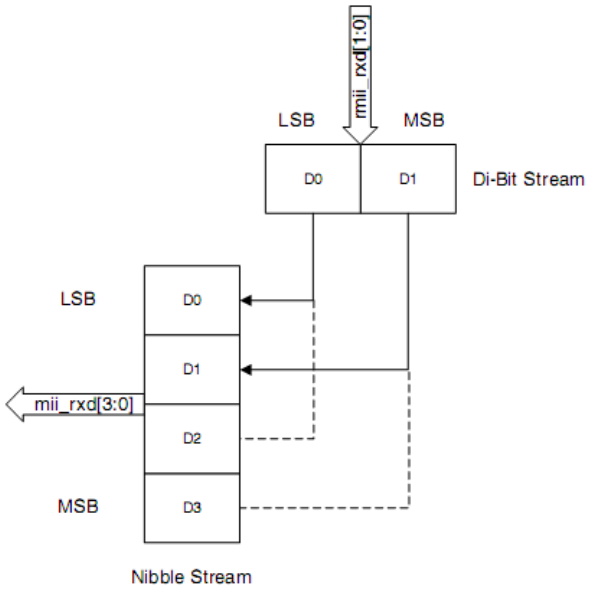


Fig. 19-9 RMII Receive Bit Ordering

19.3.2 Station Management Agent

The application can access the PHY registers through the Station Management Agent (SMA) module. SMA is a two-wire Station Management interface (MIM).

For MIM accesses, the maximum operating frequency of the MDC (gmii_mdc_o) is 2.5 MHz, as specified in the IEEE 802.3. In the GMAC core, the gmii_mdc_o clock is derived from the application clock or clk_csr_i, using a divider-counter. The divide factor depends on the clock range setting (CR field) in the MAC_MDIO_Address register Select the clock divide factor as mentioned in the description of CR field of MAC_MDIO_Address register, to meet IEEE specifications. However, if your system supports higher clock frequencies on the MIM interface, there is a provision to select a different divider.

The MDIO frame structure is as follows:

Table 19-1 MDIO Clause 45 Frame Structure

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on gmii_mdc_o signal.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2'b00
OPCODE	■ 2'b00 ■ 2'b01 ■ 2'b10 ■ 2'b11
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address select for one of 32 devices
TA	Turnaround ■ 2'bZ0: Read and post-read increment address ■ 2'b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	16-bit value: For an address cycle (OPCODE = 2'b00), this frame contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, this field contains the data to be written to the register. For read or post-read increment address frames, this field contains the contents of the register read from the PHY.

Field	Description
	■ In address and data write cycles, the GMAC drives the MDIO line during the transfer of these 16 bits. ■ In read and post-read increment address cycles, the PHY drives the MDIO line during the transfer of these 16 bits.

The frame structure for Clause 22 frames is also supported. The C45E bit in the MAC_MDIO_Address register can be programmed to enable Clause 22 or Clause 45 mode of operation. Table.1-2 shows the Clause 22 frame format.

Table 19-2 MDIO Clause 22 Frame Structure

Field	Description
IDLE	The mdio line is in tri-state; there is no clock on gmii_mdc_o signal.
PREAMBLE	32 continuous bits of value 1
START	Start of packet is 2'b01
OPCODE	■ 2'b01 : Write ■ 2'b10 : Read
PHY ADDR	5-bit address select for one of 32 PHYs
DEV ADDR	5-bit address to select the register within each MMD
TA	Turnaround ■ 2'bZ0: Read and post-read increment address ■ 2'b10: Write and address MDIO accesses Where Z is the tri-state level
DATA/ADDRESS	Any 16-bit value: ■ In a write operation, the GMAC drives MDIO ■ In read operation, the PHY drives MDIO

In addition to normal read and write operations, the SMA also supports post-read increment address while operating in Clause 45 mode.

19.3.3 TCP/IP Segmentation Offload (TSO) Engine

The TCP Segmentation Offload (TSO) engine is useful in offloading the TCP segmentation functions to the hardware.
It also supports UDP Segmentation Offload (USO) in which the UDP payload is segmented in the hardware. There are no sequencing/ordering controls available or updated in the segments, so it can be used in point to point applications in which out of order packets are not expected by the receiver. The description and flow of TSO mentioned in this section is same as USO, any deviation is provided as notes.
In the TCP segmentation offload (TSO) feature, the DMA splits a large TCP packet into multiple small packets and passes these packets to the MTL Tx Queue.

19.3.4 MAC Management Counters

The GMAC supports storing the statistics about the received and transmitted packets in registers that are accessible through the application.
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 packets. The register set includes a control register for controlling the behavior of the registers, two status registers containing interrupts generated (receive and transmit), and Interrupt Enable registers (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Each register is 32-bits wide. The write data is qualified with the corresponding mci_be_i signals. Therefore, non-32-bit accesses are allowed as long as the address is word-aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.

The MMC counters are free running. There is no separate enable for the counters to start. If a particular MMC counter is present in the RTL, it starts counting when corresponding packet is received or transmitted. The Receive MMC counters are updated for packets that are passed by the Address Filter (AFM) block. The statistics of packets, dropped by the AFM module, are not updated unless they are runt packets of less than 6 bytes (DA bytes are not received fully). To get statistics of all packets, set Bit 0 in the "MAC_Packet_Filter" register.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet packets.

19.4 Register Description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Configuration	0x0000	W	0x00008000	The MAC Configuration Register establishes the operating mode of the MAC
GMAC_MAC_Ext_Configuration	0x0004	W	0x00000000	The MAC Extended Configuration Register establishes the operating mode of the MAC
GMAC_MAC_Packet_Filter	0x0008	W	0x00000000	The MAC Packet Filter register contains the filter controls for receiving packets
GMAC_MAC_Watchdog_Timeout	0x000C	W	0x00000000	The Watchdog Timeout register controls the watchdog timeout for received packets
GMAC_MAC_Hash_Table_Reg0	0x0010	W	0x00000000	The Hash Table Register 0 contains the first 32 bits of the hash table
GMAC_MAC_Hash_Table_Reg1	0x0014	W	0x00000000	The Hash Table Register 1 contains the second 32 bits of the hash table
GMAC_MAC_VLAN_Tag	0x0050	W	0x00000000	The VLAN Tag register identifies the IEEE 802.1Q VLAN type packets
GMAC_MAC_VLAN_Hash_Table	0x0058	W	0x00000000	When VTHM bit of the MAC_VLAN_Tag register is set, the 16-bit VLAN Hash Table register is used for group address filtering based on the VLAN tag
GMAC_MAC_Q0_Tx_Flow_Ctrl	0x0070	W	0x00000000	The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC
GMAC_MAC_Rx_Flow_Ctrl	0x0090	W	0x00000000	The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet
GMAC_MAC_Interrupt_Status	0x00B0	W	0x00000000	The Interrupt Status register contains the status of interrupts
GMAC_MAC_Interrupt_Enable	0x00B4	W	0x00000000	The Interrupt Enable register contains the masks for generating the interrupts

Name	Offset	Size	Reset Value	Description
GMAC_MAC_Rx_Tx_Status	0x00B8	W	0x00000000	The Receive Transmit Status register contains the Receive and Transmit Error status
GMAC_MAC_PMT_Control_Status	0x00C0	W	0x00000000	The PMT Control and Status Register
GMAC_MAC_RWK_Packet_Filter	0x00C4	W	0x00000000	The Remote Wakeup Filter registers are implemented as 8, 16, or 32 indirect access registers (wkuppktfilter_reg#i) based on whether 4, 8, or 16 Remote Wakeup Filters are selected in the configuration and accessed by application through MAC_RWK_Packet_Filter register
GMAC_RWK_Filter0_Byte_Mask	0x10C0	W	0x00000000	RWK Filter0 Byte Mask
GMAC_RWK_Filter1_Byte_Mask	0x10C4	W	0x00000000	RWK Filter1 Byte Mask
GMAC_RWK_Filter2_Byte_Mask	0x10C8	W	0x00000000	RWK Filter2 Byte Mask
GMAC_RWK_Filter3_Byte_Mask	0x10CC	W	0x00000000	RWK Filter3 Byte Mask
GMAC_RWK_Filter01_CRC	0x10D0	W	0x00000000	RWK Filter 0/1 CRC-16
GMAC_RWK_Filter23_CRC	0x10D4	W	0x00000000	RWK Filter 2/3 CRC-16
GMAC_RWK_Filter_Offset	0x10D8	W	0x00000000	RWK Filter Offset
GMAC_RWK_Filter_Command	0x10DC	W	0x00000000	RWK Filter Command
GMAC_MAC_Version	0x0110	W	0x00001E51	The version register identifies the version of the GMAC
GMAC_MAC_Debug	0x0114	W	0x00000000	The Debug register provides the debug status of various MAC blocks
GMAC_MAC_HW_Feature0	0x011C	W	0x400103E1	This register indicates the presence of first set of the optional features or functions
GMAC_MAC_HW_Feature1	0x0120	W	0x010E4166	This register indicates the presence of second set of the optional features or functions
GMAC_MAC_HW_Feature2	0x0124	W	0x10000000	This register indicates the presence of third set of the optional features or functions
GMAC_MAC_HW_Feature3	0x0128	W	0x00000000	This register indicates the presence of fourth set the optional features or functions
GMAC_MAC_MDIO_Address	0x0200	W	0x00000000	The MDIO Address register controls the management cycles to external PHY through a management interface
GMAC_MAC_MDIO_Data	0x0204	W	0x00000000	The MDIO Data register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_Address

Name	Offset	Size	Reset Value	Description
GMAC_MAC_ARP_Address	0x0210	W	0x00000000	The ARP Address register contains the IPv4 Destination Address of the MAC
GMAC_MAC_CSR_SW_Ctrl	0x0230	W	0x00000000	This register contains SW programmable controls for changing the CSR access response and status bits clearing
GMAC_MAC_Ext_Cfg1	0x0238	W	0x00000002	This register contains Split mode control field and offset field for Split Header feature
GMAC_MAC_Address0_High	0x0300	W	0x0000FFFF	The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station
GMAC_MAC_Address0_Low	0x0304	W	0xFFFFFFFF	The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station
GMAC_MMC_Control	0x0700	W	0x00000000	This register establishes the operating mode of MMC
GMAC_MMC_Rx_Interrupt	0x0704	W	0x00000000	Maintains the interrupts generated from all Receive statistics counters
GMAC_MMC_Tx_Interrupt	0x0708	W	0x00000000	Maintains the interrupts generated from all Transmit statistics counters
GMAC_MMC_Rx_Interrupt_Mask	0x070C	W	0x00000000	This register maintains the masks for interrupts generated from all Receive statistics counters
GMAC_MMC_Tx_Interrupt_Mask	0x0710	W	0x00000000	This register maintains the masks for interrupts generated from all Transmit statistics counters
GMAC_Tx_Octet_Count_Good_Bad	0x0714	W	0x00000000	This register provides the number of bytes transmitted by the GMAC, exclusive of preamble and retried bytes, in good and bad packets
GMAC_Tx_Packet_Count_Good_Bad	0x0718	W	0x00000000	This register provides the number of good and bad packets, exclusive of retried packets
GMAC_Tx_Underflow_Error_Packets	0x0748	W	0x00000000	This register provides the number of packets aborted because of packets underflow error
GMAC_Tx_Carrier_Error_Packets	0x0760	W	0x00000000	This register provides the number of packets aborted because of carrier sense error (no carrier or loss of carrier)
GMAC_Tx_Octet_Count_Good	0x0764	W	0x00000000	This register provides the number of bytes exclusive of preamble, only in good packets
GMAC_Tx_Packet_Count_Good	0x0768	W	0x00000000	This register provides the number of good packets transmitted by GMAC
GMAC_Tx_Pause_Packets	0x0770	W	0x00000000	This register provides the number of good Pause packets by GMAC

Name	Offset	Size	Reset Value	Description
GMAC_Rx_Packets_Count_Good_Bad	0x0780	W	0x00000000	This register provides the number of good and bad packets received by GMAC
GMAC_Rx_Octet_Count_Good_Bad	0x0784	W	0x00000000	This register provides the number of bytes received by GMAC, exclusive of preamble, in good and bad packets
GMAC_Rx_Octet_Count_Good	0x0788	W	0x00000000	This register provides the number of bytes received by GMAC, exclusive of preamble, only in good packets
GMAC_Rx_Multicast_Packets_Good	0x0790	W	0x00000000	This register provides the number of good multicast packets received by
GMAC_Rx_CRC_Error_Packets	0x0794	W	0x00000000	This register provides the number of packets received by GMAC with CRC error
GMAC_Rx_Length_Error_Packets	0x07C8	W	0x00000000	This register provides the number of packets received by GMAC with length error (Length Type field not equal to packet size), for all packets with valid length field
GMAC_Rx_Pause_Packets	0x07D0	W	0x00000000	This register provides the number of good and valid Pause packets received by GMAC
GMAC_Rx_FIFO_Overflow_Packets	0x07D4	W	0x00000000	This register provides the number of missed received packets because of FIFO overflow
GMAC_MMC_IPC_Rx_Interrupt_Mask	0x0800	W	0x00000000	This register maintains the mask for the interrupt generated from the receive IPC statistic counters
GMAC_MMC_IPC_Rx_Interrupt	0x0808	W	0x00000000	This register maintains the interrupt that the receive IPC statistic counters generate
GMAC_RxIPv4_Good_Packets	0x0810	W	0x00000000	This register provides the number of good IPv4 datagrams received by GMAC with the TCP, UDP, or ICMP payload
GMAC_RxIPv4_Header_Error_Packets	0x0814	W	0x00000000	This register provides the number of IPv4 datagrams received by GMAC with header (checksum, length, or version mismatch) errors
GMAC_RxIPv6_Good_Packets	0x0824	W	0x00000000	This register provides the number of good IPv6 datagrams received by GMAC
GMAC_RxIPv6_Header_Error_Packets	0x0828	W	0x00000000	This register provides the number of IPv6 datagrams received by GMAC with header (length or version mismatch) errors
GMAC_RxUDP_Error_Packets	0x0834	W	0x00000000	This register provides the number of good IP datagrams received by GMAC whose UDP payload has a checksum error

Name	Offset	Size	Reset Value	Description
GMAC_RxTCP_Error_Packets	0x083C	W	0x00000000	This register provides the number of good IP datagrams received by GMAC whose TCP payload has a checksum error
GMAC_RxICMP_Error_Packets	0x0844	W	0x00000000	This register provides the number of good IP datagrams received by GMAC whose ICMP payload has a checksum error
GMAC_RxIPv4_Header_Error_Octets	0x0854	W	0x00000000	This register provides the number of bytes received by GMAC in IPv4 datagrams with header errors (checksum, length, version mismatch)
GMAC_RxIPv6_Header_Error_Octets	0x0868	W	0x00000000	This register provides the number of bytes received by GMAC in IPv6 datagrams with header errors (length, version mismatch)
GMAC_RxUDP_Error_Octets	0x0874	W	0x00000000	This register provides the number of bytes received by GMAC in a UDP segment that had checksum errors
GMAC_RxTCP_Error_Octets	0x087C	W	0x00000000	This register provides the number of bytes received by GMAC in a TCP segment that had checksum errors
GMAC_RxICMP_Error_Octets	0x0884	W	0x00000000	This register provides the number of bytes received by GMAC in a good ICMP segment
GMAC_MTL_Operation_Mode	0x0C00	W	0x00000000	The Operation Mode register establishes the Transmit and Receive operating modes and commands
GMAC_MTL_DBG_CTL	0x0C08	W	0x00000000	The FIFO Debug Access Control and Status register controls the operation mode of FIFO debug access
GMAC_MTL_DBG_STS	0x0C0C	W	0x01900000	The FIFO Debug Status register contains the status of FIFO debug access
GMAC_MTL_FIFO_Debug_Data	0x0C10	W	0x00000000	The FIFO Debug Data register contains the data to be written to or read from the FIFOs
GMAC_MTL_Interrupt_Status	0x0C20	W	0x00000000	The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC
GMAC_MTL_TxQ0_Operation_Mode	0x0D00	W	0x00000000	The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands

Name	Offset	Size	Reset Value	Description
GMAC_MTL_TxQ0_Underflow	0x0D04	W	0x00000000	The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush
GMAC_MTL_TxQ0_Debug	0x0D08	W	0x00000000	The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue
GMAC_MTL_Q0_Interrupt_Ctrl_Status	0x0D2C	W	0x00000000	This register contains the interrupt enable and status bits for the queue 0 interrupts
GMAC_MTL_RxQ0_Operation_Mode	0x0D30	W	0x00000000	The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command
GMAC_MTL_RxQ0_Miss_Pkt_Ovf_Cnt	0x0D34	W	0x00000000	The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow
GMAC_MTL_RxQ0_Debug	0x0D38	W	0x00000000	The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue
GMAC_DMA_Mode	0x1000	W	0x00000000	The Bus Mode register establishes the bus operating modes for the DMA
GMAC_DMA_SysBus_Mode	0x1004	W	0x00010000	The System Bus mode register controls the behavior of the AHB or AXI master
GMAC_DMA_Interrupt_Status	0x1008	W	0x00000000	The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC
GMAC_DMA_Debug_Status0	0x100C	W	0x00000000	The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose
GMAC_AXI_LPI_Entry_Interval	0x1040	W	0x00000000	This register is used to control the AXI LPI entry interval
GMAC_DMA_CH0_Control	0x1100	W	0x00000000	The register specifies the MSS value for segmentation, length to skip between two descriptors, and 8xPBL mode
GMAC_DMA_CH0_Tx_Control	0x1104	W	0x00000000	The register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights

Name	Offset	Size	Reset Value	Description
GMAC_DMA_CH0_Rx_Control	0x1108	W	0x00000000	The DMA Channel0 Receive Control register controls the Rx features such as PBL, buffer size, and extended status
GMAC_DMA_CH0_TxDesc_List_HAddress	0x1110	W	0x00000000	The Channel0 Tx Descriptor List HAddress register has the higher 8 or 16 bits of the start address of the Transmit descriptor list
GMAC_DMA_CH0_TxDesc_List_Address	0x1114	W	0x00000000	The Channel0 Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list
GMAC_DMA_CH0_RxDesc_List_HAaddress	0x1118	W	0x00000000	The Channel0 Rx Descriptor List HAddress register has the higher 8 or 16 bits of the start address of the Receive descriptor list
GMAC_DMA_CH0_RxDesc_List_Address	0x111C	W	0x00000000	The Channel0 Rx Descriptor List Address register points the DMA to the start of Receive descriptor list
GMAC_DMA_CH0_TxDesc_Tail_Pointer	0x1120	W	0x00000000	The Channel0 Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor
GMAC_DMA_CH0_RxDesc_Tail_Pointer	0x1128	W	0x00000000	The Channel0 Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location of the last valid descriptor
GMAC_DMA_CH0_TxDesc_Ring_Length	0x112C	W	0x00000000	The Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring
GMAC_DMA_CH0_RxDesc_Ring_Length	0x1130	W	0x00000000	The Channel0 Rx Descriptor Ring Length register contains the length of the Receive descriptor circular ring
GMAC_DMA_CH0_Interrupt_Enable	0x1134	W	0x00000000	The Channel0 Interrupt Enable register enables the interrupts reported by the Status register
GMAC_DMA_CH0_Rx_Interrupt_Watchdog_Timer	0x1138	W	0x00000000	The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA
GMAC_DMA_CH0_Current_App_TxDesc	0x1144	W	0x00000000	The Channel0 Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA
GMAC_DMA_CH0_Current_App_RxDesc	0x114C	W	0x00000000	The Channel0 Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA

Name	Offset	Size	Reset Value	Description
GMAC_DMA_CH0_Curren_App_TxBuffer_H	0x1150	W	0x00000000	The Channeli Current Application Transmit Buffer Address High register has the higher 8 or 16 bits of the current address of the Transmit buffer address read by the DMA
GMAC_DMA_CH0_Current_App_TxBuffer	0x1154	W	0x00000000	The Channel0 Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA
GMAC_DMA_CH0_Current_App_RxBuffer_H	0x1158	W	0x00000000	The Channeli Current Application Receive Buffer Address High register has the higher 8 or 16 bits of the current address of the Receive buffer address read by the DMA
GMAC_DMA_CH0_Current_App_RxBuffer	0x115C	W	0x00000000	The Channel0 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA
GMAC_DMA_CH0_Status	0x1160	W	0x00000000	The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA
GMAC_DMA_CH0_Miss_Frame_Cnt	0x1164	W	0x00000000	This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH$i_Rx_Control register
GMAC_DMA_CH0_RX_ERI_Cnt	0x1168	W	0x00000000	The DMA_CH0_RX_ERI_Cnt registers provides the count of the number of times ERI was asserted

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

19.4.2 Detail Registers Description

GMAC_MAC_Configuration

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RW	0x0	ARPEN ARP Offload Enable When this bit is set, the MAC can recognize an incoming ARP request packet and schedules the ARP packet for transmission. It forwards the ARP packet to the application and also indicate the events in the RxStatus. When this bit is reset, the MAC receiver does not recognize any ARP packet and indicates them as Type frame in the RxStatus. This bit is available only when the Enable IPv4 ARP Offload is selected. Values: 1'b0: ARP Offload is disabled 1'b1: ARP Offload is enabled
30:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RW	0x0	IPC Checksum Offload When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled.The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit. Values: 1'b0: IP header/payload checksum checking is disabled 1'b1: IP header/payload checksum checking is enabled
26:24	RW	0x0	IPG Inter-Packet Gap These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode.In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered. When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG.The above function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_Ext_Configuration register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_Ext_Configuration register. Values: 3'b000: 96 bit times IPG 3'b001: 88 bit times IPG 3'b010: 80 bit times IPG 3'b011: 72 bit times IPG 3'b100: 64 bit times IPG 3'b101: 56 bit times IPG 3'b110: 48 bit times IPG 3'b111: 40 bit times IPG
23	RW	0x0	GPSLCE Giant Packet Size Limit Control Enable When this bit is set, the MAC considers the value in GPSL field in MAC_Ext_Configuration register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit. When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet).The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status. Values: 1'b0: Giant Packet Size Limit Control is disabled 1'b1: Giant Packet Size Limit Control is enabled

Bit	Attr	Reset Value	Description
22	RW	0x0	S2KP IEEE 802.3as Support for 2K Packets When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets.When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. Note: When the JE bit is set, setting this bit has no effect on the giant packet status. Values: 1'b0: Support upto 2K packet is disabled 1'b1: Support upto 2K packet is Enabled
21	RW	0x0	CST CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits. Values: 1'b0: CRC stripping for Type packets is disabled 1'b1: CRC stripping for Type packets is enabled
20	RW	0x0	ACS Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field.When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit . Values: 1'b0: Automatic Pad or CRC Stripping is disabled 1'b1: Automatic Pad or CRC Stripping is enabled
19	RW	0x0	WD Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes.When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes. Values: 1'b0: Watchdog is enabled 1'b1: Watchdog is disabled
18:17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	JE Jumbo Packet Enable When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status. Values: 1'b0: Jumbo packet is disabled 1'b1: Jumbo packet is enabled
15	RW	0x1	PS Port Select This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit. Values: 1'b0: For 1000 or 2500 Mbps operations 1'b1: For 10 or 100 Mbps operations
14	RO	0x0	reserved
13	RW	0x0	DM Duplex Mode When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations. Values: 1'b0: Half-duplex mode 1'b1: Full-duplex mode
12	RW	0x0	LM Loopback Mode When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back. Values: 1'b0: Loopback is disabled 1'b1: Loopback is enabled
11:10	RO	0x0	reserved
9	RW	0x0	DCRS Disable Carrier Sense During Transmission When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission. When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission. Values: 1'b0: Enable Carrier Sense During Transmission 1'b1: Disable Carrier Sense During Transmission

Bit	Attr	Reset Value	Description
8	RW	0x0	DR Disable Retry When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status.When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode. Values: 1'b0: Enable Retry 1'b1: Disable Retry
7	RO	0x0	reserved
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 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt. The random integer r takes the value in the range 0 <= r < 2^k This bit is applicable only in the half-duplex mode. Values: 2'b00: k = min(n,10) 2'b01: k = min(n,8) 2'b10: k = min(n,4) 2'b11: k = min(n,1)
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode. If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode. Values: 1'b0: Deferral check function is disabled 1'b1: Deferral check function is enabled

Bit	Attr	Reset Value	Description
3:2	RW	0x0	PRELEN Preamble Length for Transmit packets These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode. Values: 2'b00: 7 bytes of preamble 2'b01: 5 bytes of preamble 2'b10: 3 bytes of preamble 2'b11: Reserved
1	RO	0x0	reserved
0	RW	0x0	RE Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII or MII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII or MII interface. Values: 1'b0: Receiver is disabled 1'b1: Receiver is enabled

GMAC_MAC_Ext_Configuration

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:25	RW	0x00	EIPG Extended Inter-Packet Gap The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in MAC_Configuration register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: {EIPG, IPG} 8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ---------- 8'hFF - 2144 bit times
24	RW	0x0	EIPGEN Extended Inter-Packet Gap Enable When this bit is set, the MAC interprets EIPG field and IPG field in MAC_Configuration register together as minimum IPG greater than 96 bit times in steps of 8 bit times. When this bit is reset, the MAC ignores EIPG field and interprets IPG field in MAC_Configuration register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times. Note: The extended Inter-Packet Gap feature must be enabled when operating in Full-Duplex mode only. There may be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode. Values: 1'b0: Extended Inter-Packet Gap is disabled 1'b1: Extended Inter-Packet Gap is enabled
23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22:20	RW	0x0	HDSMS Maximum Size for Splitting the Header Data These bits indicate the maximum header size allowed for splitting the header data in the received packet. Values: 0x0 (64BYTES): Maximum Size for Splitting the Header Data is 64 bytes 0x1 (128BYTES): Maximum Size for Splitting the Header Data is 128 bytes 0x2 (256BYTES): Maximum Size for Splitting the Header Data is 256 bytes 0x3 (512BYTES): Maximum Size for Splitting the Header Data is 512 bytes 0x4 (1024BYTES): Maximum Size for Splitting the Header Data is 1024 bytes 0x5 (RSVD): Reserved
19	RO	0x0	reserved
18	RW	0x0	USP Unicast Slow Protocol Packet Detect When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_Address0_High and MAC_Address0_Low registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02). When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5. Values: 1'b0: Unicast Slow Protocol Packet Detection is disabled 1'b1: Unicast Slow Protocol Packet Detection is enabled
17	RW	0x0	SPEN Slow Protocol Detection Enable When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Rx status. The MAC discards the Slow Protocol packets with invalid sub-types. When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets. Values: 1'b0: Slow Protocol Detection is disabled 1'b1: Slow Protocol Detection is enabled
16	RW	0x0	DCRCC Disable CRC Checking for Received Packets When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets. Values: 1'b0: CRC Checking is enabled 1'b1: CRC Checking is disabled
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	GPSL Giant Packet Size Limit If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes. For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_Configuration register.

GMAC_MAC_Packet_Filter

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the MAC Receiver module passes all received packets to the application, irrespective of whether they pass the address filter or not. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bit in the Rx Status Word. When this bit is reset, the Receiver module passes only those packets to the application that pass the SA or DA address filter. 0x0 (DISABLE): Receive All is disabled 0x1 (ENABLE): Receive All is enabled
30:17	RO	0x0000	reserved
16	RW	0x0	VTFE VLAN Tag Filter Enable When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag. Values: 1'b0: VLAN Tag Filter is disabled 1'b1: VLAN Tag Filter is enabled
15:11	RO	0x00	reserved
10	RW	0x0	HPF Hash or Perfect Filter When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit. When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter. Values: 1'b0: Hash or Perfect Filter is disabled 2'b1: Hash or Perfect Filter is enabled
9:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	PCF Pass Control Packets These bits control the forwarding of all control packets (including unicast and multicast Pause packets). Values: 2'b00: MAC filters all control packets from reaching the application 2'b01: MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 2'b10: MAC forwards all control packets to the application even if they fail the Address filter 2'b11: MAC forwards the control packets that pass the Address filter
5	RW	0x0	DBF Disable Broadcast Packets When this bit is set, the AFM module blocks all incoming broadcast packets. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast packets. Values: 1'b0: Enable Broadcast Packets 1'b1: Disable Broadcast Packets
4	RW	0x0	PM Pass All Multicast When this bit is set, it indicates that all received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit. Values: 1'b0: Pass All Multicast is disabled 1'b1: Pass All Multicast is enabled
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 packets. When this bit is reset, normal filtering of packets is performed. Values: 1'b0: DA Inverse Filtering is disabled 1'b1: DA Inverse Filtering is enabled
2	RW	0x0	HMC Hash Multicast When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table. When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers. Values: 1'b0: Hash Multicast is disabled 1'b1: Hash Multicast is enabled
1:0	RO	0x0	reserved

GMAC_MAC_Watchdog_Timeout

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	PWE Programmable Watchdog Enable When this bit is set and the WD bit of the MAC_Configuration register is reset, the WTO field is used as watchdog timeout for a received packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_Configuration register. Values: 1'b0: Programmable Watchdog is disabled 1'b1: Programmable Watchdog is enabled
7:4	RO	0x0	reserved
3:0	RW	0x0	WTO Watchdog Timeout When the PWE bit is set and the WD bit of the MAC_Configuration register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet. Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped. Values: 4'b0000: 2 KB 4'b0001: 3 KB 4'b0010: 4 KB 4'b0011: 5 KB 4'b0100: 6 KB 4'b0101: 7 KB 4'b0110: 8 KB 4'b0111: 9 KB 4'b1000: 10 KB 4'b1001: 11 KB 4'b1010: 12 KB 4'b1011: 13 KB 4'b1100: 14 KB 4'b1101: 15 KB 4'b1110: 16383 Bytes 4'b1111: Reserved

GMAC_MAC_Hash_Table_Reg0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HT31T0 MAC Hash Table First 32 Bits This field contains the first 32 Bits [31:0] of the Hash table. The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5. The hash value of the destination address is calculated in the following way: 1. Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32). 2. Perform bitwise reversal for the value obtained in Step 1. 3. Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2. If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values. If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

GMAC_MAC_Hash_Table_Reg1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HT63T32 MAC Hash Table Second 32 Bits This field contains the second 32 Bits [63:32] of the Hash table.

GMAC_MAC_VLAN_Tag

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	EVLRXS Enable VLAN Tag in Rx status When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status. Values: 1'b0: VLAN Tag in Rx status is disabled 1'b1: VLAN Tag in Rx status is enabled
23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22:21	RW	0x0	EVLS Enable VLAN Tag Stripping on Receive This field indicates the stripping operation on the outer VLAN Tag in received packet. Values: 2'b00: Do not strip 2'b01: Strip if VLAN filter passes 2'b10: Strip if VLAN filter fails 2'b11: Always strip
20	RW	0x0	DOVLTC Disable VLAN Type Check When this bit is set, the MAC does not check whether the VLAN Tag specified by the ERIVLT bit is of type S-VLAN or C-VLAN. When this bit is reset, the MAC filters or matches the VLAN Tag specified by the ERIVLT bit only when VLAN Tag type is similar to the one specified by the ERSVLM bit. Values: 1'b0: VLAN Type Check is enabled 1'b1: VLAN Type Check is disabled
19	RW	0x0	ERSVLM Enable Receive S-VLAN Match When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets. The ERIVLT bit determines the VLAN tag position considered for filtering or matching. Values: 1'b0: Receive S-VLAN Match is disabled 1'b1: Receive S-VLAN Match is enabled
18	RW	0x0	ESVL Enable S-VLAN When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets. Values: 1'b0: S-VLAN is disabled 1'b1: S-VLAN is enabled
17	RW	0x0	VTIM VLAN Tag Inverse Match Enable When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched. Values: 1'b0: VLAN Tag Inverse Match is disabled 1'b1: VLAN Tag Inverse Match is enabled

Bit	Attr	Reset Value	Description
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits[11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet. Similarly, when enabled, only 12 bits of the VLAN tag in the received packet are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN packet are used for comparison and VLAN hash filtering. Values: 1'b0: 12-Bit VLAN Tag Comparison is disabled 1'b1: 12-Bit VLAN Tag Comparison is enabled
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Packets This field contains the 802.1Q VLAN tag to identify the VLAN packets. This VLAN tag identifier is compared to the 15th and 16th bytes of the packets being received for VLAN packets. The following list describes the bits of this field: 1. Bits[15:13]: User Priority 2. Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) 3. Bits[11:0]: VLAN Identifier (VID) field of VLAN tag When the ETV bit is set, only the VID is used for comparison. If this field ([11:0] if ETV is set) is all zeros, the MAC does not check the 15th and 16th bytes for VLAN tag comparison and declares all packets with Type field value of 0x8100 or 0x88a8 as VLAN packets.

GMAC_MAC_VLAN_Hash_Table

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	VLHT VLAN Hash Table This field contains the 16-bit VLAN Hash Table. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_Tag Register) in the incoming packet is passed through the CRC logic. When ETV bit of MAC_VLAN_Tag register is set, the upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. When ETV bit of MAC_VLAN_Tag register is reset, the ones-complement of upper four bits of the calculated CRC are used to index the contents of the VLAN Hash table. For example, when ETV bit is set a hash value of 4b'1000 selects Bit 8 of the VLAN Hash table, whereas when ETV bit is reset a hash value of 4b'1000 selects Bit 7 of the VLAN Hash table. The hash value of the destination address is calculated in the following way: 1.Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3). 2.Perform bitwise reversal for the value obtained in step 1. 3.Take the upper four bits from the value obtained in step 2. If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[15:8] (in little-endian mode) or Bits[7:0] (in big-endian mode) of this register are written. 1.If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

GMAC_MAC_Q0_Tx_Flow_Ctrl

Address: Operational Base + offset (0x0070)

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 Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.
15:8	RO	0x00	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i). When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled. Values: 1'b0: Zero-Quanta Pause packet generation is enabled 1'b1: Zero-Quanta Pause packet generation is disabled
6:2	RO	0x00	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	TFE Transmit Flow Control Enable Full-Duplex Mode: In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets. Half-Duplex Mode: In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled. Values: 1'b0: Transmit Flow Control is disabled 1'b1: Transmit Flow Control is enabled
0	RW	0x0	FCB_BPA Flow Control Busy or Backpressure Activate This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. Full-Duplex Mode: In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared. Half-Duplex Mode: When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Flow Control Busy or Backpressure Activate is disabled 1'b1: Flow Control Busy or Backpressure Activate is enabled

GMAC_MAC_Rx_Flow_Ctrl

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	UP Unicast Pause Packet Detect A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_Address0_High and MAC_Address0_Low. When this bit is reset, the MAC only detects Pause packets with unique multicast address. Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011. Values: 1'b0: Unicast Pause Packet Detect disabled 1'b1: Unicast Pause Packet Detect enabled
0	RW	0x0	RFE Receive Flow Control Enable When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled. When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time. Values: 1'b0: Receive Flow Control is disabled 1'b1: Receive Flow Control is enabled

GMAC_MAC_Interrupt_Status

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18	RO	0x0	MDIOIS MDIO Interrupt Status This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit of MAC_CSR_SW_Ctrl register is set. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: MDIO Interrupt status not active 1'b1: MDIO Interrupt status active
17:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RO	0x0	RXSTSIS Receive Status Interrupt This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_Rx_Tx_Status register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. Values: 1'b0: Receive Interrupt status not active 1'b1: Receive Interrupt status active
13	RO	0x0	TXSTSIS Transmit Status Interrupt This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_Rx_Tx_Status register: 1. Excessive Collision (EXCOL) 2. Late Collision (LCOL) 3. Excessive Deferral (EXDEF) 4. Loss of Carrier (LCARR) 5. No Carrier (NCARR) 6. Jabber Timeout (TJT) This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register. Values: 1'b0: Transmit Interrupt status not active 1'b1: Transmit Interrupt status active
12	RO	0x0	reserved
11	RO	0x0	MMCRXIPIS MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) and Enable Receive TCP/IP Checksum Check options. Values: 1'b0: MMC Receive Checksum Offload Interrupt status not active 1'b1: MMC Receive Checksum Offload Interrupt status active
10	RO	0x0	MMCTXIS MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Transmit Interrupt status not active 1'b1: MMC Transmit Interrupt status active

Bit	Attr	Reset Value	Description
9	RO	0x0	MMCRXIS MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all bits in this interrupt register are cleared. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Receive Interrupt status not active 1'b1: MMC Receive Interrupt status active
8	RO	0x0	MMCIS MMC Interrupt Status This bit is set high when Bit 11, Bit 10, or Bit 9 is set high. This bit is cleared only when all these bits are low. This bit is valid only when you select the Enable MAC Management Counters (MMC) option. Values: 1'b0: MMC Interrupt status not active 1'b1: MMC Interrupt status active
7:5	RO	0x0	reserved
4	RO	0x0	PMTIS PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN packet is received in the power-down mode (RWKPRCVD and MGKPRCVD bits in MAC_PMT_Control_Status register). This bit is cleared when corresponding interrupt source bit are cleared because of a Read operation to the MAC_PMT_Control_Status register (or corresponding interrupt source bit of MAC_PMT_Control_Status register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). This bit is valid only when you select the Enable Power Management option. Values: 1'b0: PMT Interrupt status not active 1'b1: PMT Interrupt status active
3	RO	0x0	PHYIS PHY Interrupt This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set). Values: 1'b0: PHY Interrupt not detected 1'b1: PHY Interrupt detected
2:0	RO	0x0	reserved

GMAC_MAC_Interrupt_Enable

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18	RW	0x0	MDIOIE MDIO Interrupt Enable When this bit is set, it enables the assertion of the interrupt when MDIOIS field is set in the MAC_Interrupt_Status register. Values: 1'b0: MDIO Interrupt is disabled 1'b1: MDIO Interrupt is enabled
17:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	RXSTSIE Receive Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of RXSTSIS bit in the MAC_Interrupt_Status register. Values: 1'b0: Receive Status Interrupt is disabled 1'b1: Receive Status Interrupt is enabled
13	RW	0x0	TXSTSIE Transmit Status Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of TXSTSIS bit in the MAC_Interrupt_Status register. Values: 1'b0: Timestamp Status Interrupt is disabled 1'b1: Timestamp Status Interrupt is enabled
12:5	RO	0x00	reserved
4	RW	0x0	PMTIE PMT Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PMTIS bit in MAC_Interrupt_Status register. Values: 1'b0: PMT Interrupt is disabled 1'b1: PMT Interrupt is enabled
3	RW	0x0	PHYIE PHY Interrupt Enable When this bit is set, it enables the assertion of the interrupt signal because of the setting of PHYIS bit in MAC_Interrupt_Status register. Values: 1'b0: PHY Interrupt is disabled 1'b1: PHY Interrupt is enabled
2:0	RO	0x0	reserved

GMAC_MAC_Rx_Tx_Status

Address: Operational Base + offset (0x00B8)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved
8	RO	0x0	RWT Receive Watchdog Timeout This bit is set when a packet with length greater than 2,048 bytes is received (10, 240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_Configuration register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_Configuration register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No receive watchdog timeout 1'b1: Receive watchdog timed out
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RO	0x0	LCOL Late Collision When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode). This bit is not valid if the Underflow error occurs. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No collision 1'b1: Late collision is sensed
3	RO	0x0	EXDEF Excessive Deferral When the DTXSTS bit is set in the MTL_Operation_Mode register and the DC bit is set in the MAC_Configuration register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled). Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Excessive deferral 1'b1: Excessive deferral
2	RO	0x0	LCARR Loss of Carrier When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Carrier is present 1'b1: Loss of carrier
1	RO	0x0	NCARR No Carrier When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Carrier is present 1'b1: No carrier

Bit	Attr	Reset Value	Description
0	RO	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_Configuration register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_Configuration register. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Transmit Jabber Timeout 1'b1: Transmit Jabber Timeout occurr

GMAC_MAC_PMT_Control_Status

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31	RW	0x0	RWKFILTRST Remote Wake-Up Packet Filter Register Pointer Reset When this bit is set, the remote wake-up packet filter register pointer is reset to 3'b000. It is automatically cleared after 1 clock cycle. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Remote Wake-Up Packet Filter Register Pointer is not Reset 1'b1: Remote Wake-Up Packet Filter Register Pointer is Reset
30:29	RO	0x0	reserved
28:24	RO	0x00	RWKPTR Remote Wake-up FIFO Pointer This field gives the current value (0 to 7, 15, or 31 when 4, 8, or 16 Remote Wake-up Packet Filters are selected) of the Remote Wake-up Packet Filter register pointer. When the value of this pointer is equal to maximum for the selected number of Remote Wake-up Packet Filters, the contents of the Remote Wake-up Packet Filter Register are transferred to the clk_rx_i domain when a Write occurs to that register.
23:11	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	RWKPFE Remote Wake-up Packet Forwarding Enable When this bit is set along with RWKPKTEN, the MAC receiver drops all received frames until it receives the expected Wake-up frame. All frames after that event including the received wake-up frame are forwarded to application. This bit is then self-cleared on receiving the wake-up packet. The application can also clear this bit before the expected wake-up frame is received. In such cases, the MAC reverts to the default behavior where packets received are forwarded to the application. This bit must only be set when RWKPKTEN is set high and PWRDWN is set low. The setting of this bit has no effect when PWRDWN is set high. Note: If Magic Packet Enable and Wake-Up Frame Enable are both set along with setting of this bit and Magic Packet is received prior to wake-up frame, this bit is self-cleared on receiving Magic Packet, the received Magic packet is dropped, and all frames after received Magic Packet are forwarded to application. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Remote Wake-up Packet Forwarding is disabled 1'b1: Remote Wake-up Packet Forwarding is enabled
9	RW	0x0	GLBLUCAST Global Unicast When this bit set, any unicast packet filtered by the MAC (DAF) address recognition is detected as a remote wake-up packet. Values: 1'b0: Global unicast is disabled 1'b1: Global unicast is enabled
8:7	RO	0x0	reserved
6	RO	0x0	RWKPRCVD Remote Wake-Up Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a remote wake-up packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: Remote wake-up packet is received 1'b1: Remote wake-up packet is received
5	RO	0x0	MGKPRCVD Magic Packet Received When this bit is set, it indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared when this register is read. Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event. Values: 1'b0: No Magic packet is received 1'b1: Magic packet is received
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	RWKPKTEN Remote Wake-Up Packet Enable When this bit is set, a power management event is generated when the MAC receives a remote wake-up packet. Values: 1'b0: Remote wake-up packet is disabled 1'b1: Remote wake-up packet is enabled
1	RW	0x0	MGKPKTEN Magic Packet Enable When this bit is set, a power management event is generated when the MAC receives a magic packet. Values: 1'b0: Magic Packet is disabled 1'b1: Magic Packet is enabled
0	RW	0x0	PWRDWN Power Down When this bit is set, the MAC receiver drops all received packets until it receives the expected magic packet or remote wake-up packet. This bit is then self-cleared and the power-down mode is disabled. The software can clear this bit before the expected magic packet or remote wake-up packet is received. The packets received by the MAC after this bit is cleared are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Remote Wake-Up Packet Enable bit is set high. Note: You can gate-off the CSR clock during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, the Software cannot clear this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Power down is disabled 1'b1: Power down is enabled

GMAC_MAC_RWK_Packet_Filter

Address: Operational Base + offset (0x00C4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	WKUPFRMFTR(cont.) (4) If filters chained by And_Previous bit setting have complementary programming, then a frame may never pass the AND chained filter. For example, if Filter 2 And_Previous bit is set (bit 1 in Filter 2 command is set), Filter 1 Address_Type bit is set (bit 3 in Filter 1 command is set) indicating multicast detection and Filter 2 Address_Type bit is reset (bit 3 in Filter 2 command is reset) indicating unicast detection or vice versa, a remote wakeup frame does not pass the AND chained filter as a remote wakeup frame cannot be of both unicast and multicast address type. 4. Bit 0 is the enable for filter i. If Bit 0 is not set, filter i is disabled. Filter i Byte Mask: The filter i byte mask register defines the bytes of the packet that are examined by filter i (0, 1, 2, 3, .., 15) to determine whether or not a packet is a wake-up packet. 1. The MSB (31st bit) must be zero. 2. Bit j[30:0] is the byte mask. 3. If Bit j (byte number) of the byte mask is set, the CRC block processes the Filter i Offset + j of the incoming packet; otherwise Filter i Offset + j is ignored. Filter i Offset: The filter i offset register defines the offset (within the packet) from which the filter i examines the packets. 1. This 8-bit pattern-offset is the offset for the filter i first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet. Filter i CRC-16: The filter i CRC-16 register contains the CRC-16 value calculated from the pattern and the byte mask programmed in the Remote Wakeup filter register. 1. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1 2. Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following: (1)32-bit Mask: Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1, the corresponding byte is taken into the CRC16 calculation. (2) 8-bit Offset Pointer: Specifies the byte to start the CRC-16 computation. The pointer and the mask are used together to locate the bytes to be used in the CRC-16 calculations.

GMAC_RWK_Filter0_Byte_Mask

Address: Operational Base + offset (0x10C0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter0_Byte_Mask Filter0 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter1_Byte_Mask

Address: Operational Base + offset (0x10C4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter1_Byte_Mask Filter1 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter2_Byte_Mask

Address: Operational Base + offset (0x10C8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter2_Byte_Mask Filter2 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter3_Byte_Mask

Address: Operational Base + offset (0x10CC)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Filter3_Byte_Mask Filter3 32-bit Mask Each bit in this mask corresponds to one byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.

GMAC_RWK_Filter01_CRC

Address: Operational Base + offset (0x10D0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	Filter1_CRC Filter1 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1
15:0	RW	0x0000	Filter0_CRC Filter0 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1

GMAC_RWK_Filter23_CRC

Address: Operational Base + offset (0x10D4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	Filter3_CRC Filter3 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15 + x^2 + 1
15:0	RW	0x0000	Filter2_CRC Filter2 CRC-16 This filter CRC-16 contains the CRC_16 value of the pattern. The 16-bit CRC calculation uses the following polynomial: G(x) = x^16 + x^15+ x^2 + 1

GMAC_RWK_Filter_Offset

Address: Operational Base + offset (0x10D8)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	Filter3_Offset Filter3 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
23:16	RW	0x00	Filter2_Offset Filter2 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
15:8	RW	0x00	Filter1_Offset Filter1 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.
7:0	RW	0x00	Filter0_Offset Filter0 Offset This filter offset defines the offset (within the packet) from which the filter examines the packets. 1. This 8-bit pattern-offset is the offset for the filter first byte to be examined. 2. The minimum allowed offset is 12, which refers to the 13th byte of the packet. 3. The offset value 0 refers to the first byte of the packet.

GMAC_RWK_Filter_Command

Address: Operational Base + offset (0x10DC)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved

Bit	Attr	Reset Value	Description
19:16	RW	0x0	Filter2_Command Filter2 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.
15:12	RO	0x0	reserved
11:8	RW	0x0	Filter1_Command Filter1 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	Filter0_Command Filter0 Command The 4-bit filter command controls the filter operation. 1. Bit 3 specifies the address type, defining the destination address type of the pattern. When the bit is set, the pattern applies to only multicast packets; when the bit is reset, the pattern applies only to unicast packet. 2. Bit 2 (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value. 3. Bit 2, along with Bit 1, allows a MAC to reject a subset of remote wake-up packets by creating filter logic such as "Pattern 1 AND NOT Pattern 2". 4. Bit 1 (And_Previous) implements the Boolean logic. When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set. 5. Bit 0 is the enable for filter. If Bit 0 is not set, filter is disabled.

GMAC_MAC_Version

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:8	RW	0x1e	USERVER User-defined Version
7:0	RW	0x51	RKVER Rockchip-defined Version

GMAC_MAC_Debug

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:19	RO	0x0000	reserved
18:17	RO	0x0	TFCSTS MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. Values: 2'b00: Idle state 2'b01: Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 2'b10: Generating and transmitting a Pause control packet (in full-duplex mode) 2'b11: Transferring input packet for transmission
16	RO	0x0	TPESTS MAC GMII or MII Transmit Protocol Engine Status When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state. Values: 1'b0: MAC GMII or MII Transmit Protocol Engine Status not detected 1'b1: MAC GMII or MII Transmit Protocol Engine Status detected
15:3	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
2:1	RO	0x0	RFCFCSTS MAC Receive Packet Controller FIFO Status When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.
0	RO	0x0	reserved

GMAC_MAC_HW_Feature0

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:28	RO	0x4	ACTPHYSEL Active PHY Selected When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion. Values: 4'b0000: GMII or MII 4'b0001: RGMII 4'b0010: SGMII 4'b0011: TBI 4'b0100: RMII 4'b0101: RTBI 4'b0110: SMII 4'b0111: RevMII
27	RO	0x0	SAVLANINS Source Address or VLAN Insertion Enable This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected. Values: 1'b0: Source Address or VLAN Insertion Enable option is not selected 1'b1: Source Address or VLAN Insertion Enable option is selected
26:25	RO	0x0	TSSTSSEL Timestamp System Time Source This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. Values: 2'b00: Internal 2'b01: External 2'b10: Both 2'b11: Reserved
24	RO	0x0	reserved
23	RO	0x0	MACADR32SEL MAC Addresses 32-63 Selected This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected. Values: 1'b0: MAC Addresses 32-63 Select option is not selected 1'b1: MAC Addresses 32-63 Select option is selected
22:18	RO	0x00	ADDMACADRSEL MAC Addresses 1-31 Selected This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option.
17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RO	0x1	RXCOESEL Receive Checksum Offload Enabled This bit is set to 1 when the Enable Receive TCP/IP Checksum Check option is selected. Values: 1'b0: Receive Checksum Offload Enable option is not selected 1'b1: Receive Checksum Offload Enable option is selected
15:14	RO	0x0	reserved
13	RO	0x0	EEESEL Energy Efficient Ethernet Enabled This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected. Values: 1'b0: Energy Efficient Ethernet Enable option is not selected 1'b1: Energy Efficient Ethernet Enable option is selected
12	RO	0x0	TSSEL IEEE 1588-2008 Timestamp Enabled This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected. Values: 1'b0: IEEE 1588-2008 Timestamp Enable option is not selected. 1'b1: IEEE 1588-2008 Timestamp Enable option is selected.
11:10	RO	0x0	reserved
9	RO	0x1	ARPOFFSEL ARP Offload Enabled This bit is set to 1 when the Enable IPv4 ARP Offload option is selected. Values: 1'b0: ARP Offload Enable option is not selected 1'b1: ARP Offload Enable option is selected
8	RO	0x1	MMCSEL RMON Module Enable This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected. Values: 1'b0: RMON Module Enable option is not selected 1'b1: RMON Module Enable option is selected
7	RO	0x1	MGKSEL PMT Magic Packet Enable This bit is set to 1 when the Enable Magic Packet Detection option is selected. Values: 1'b0: PMT Magic Packet Enable option is not selected 1'b1: PMT Magic Packet Enable option is selected
6	RO	0x1	RWKSEL PMT Remote Wake-up Packet Enable This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected. Values: 1'b0: PMT Remote Wake-up Packet Enable option is not selected 1'b1: PMT Remote Wake-up Packet Enable option is selected

Bit	Attr	Reset Value	Description
5	RO	0x1	SMASEL SMA (MDIO) Interface This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected. Values: 1'b0: SMA (MDIO) Interface not selected 1'b1: SMA (MDIO) Interface selected
4	RO	0x0	VLHASH VLAN Hash Filter Selected This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected. Values: 1'b0: VLAN Hash Filter not selected 1'b1: VLAN Hash Filter selected
3	RO	0x0	PCSSEL PCS Registers (TBI, SGMII, or RTBI PHY interface) This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected. Values: 1'b0: No PCS Registers (TBI, SGMII, or RTBI PHY interface) 1'b1: PCS Registers (TBI, SGMII, or RTBI PHY interface)
2	RO	0x0	HDSEL Half-duplex Support This bit is set to 1 when the half-duplex mode is selected. Values: 1'b0: No Half-duplex support 1'b1: Half-duplex support
1	RO	0x0	GMIISEL 1000 Mbps Support This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation. Values: 1'b0: No 1000 Mbps support 1'b1: 1000 Mbps support
0	RO	0x1	MIISEL 10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation. Values: 1'b0: No 10 or 100 Mbps support 1'b1: 10 or 100 Mbps support

GMAC_MAC_HW_Feature1

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:27	RO	0x0	L3L4FNUM Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters: Values: 4'b0000: No L3 or L4 Filter 4'b0001: 1 L3 or L4 Filter 4'b0010: 2 L3 or L4 Filters 4'b0011: 3 L3 or L4 Filters 4'b0100: 4 L3 or L4 Filters 4'b0101: 5 L3 or L4 Filters 4'b0110: 6 L3 or L4 Filters 4'b0111: 7 L3 or L4 Filters 4'b1000: 8 L3 or L4 Filters
26	RO	0x0	reserved
25:24	RO	0x1	HASHTBLSZ Hash Table Size This field indicates the size of the hash table: Values: 2'b00: No hash table 2'b01: 64 2'b10: 128 2'b11: 256
23	RO	0x0	POUOST One Step for PTP over UDP/IP Feature Enable This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected. Values: 1'b0: One Step for PTP over UDP/IP Feature is not selected 1'b1: One Step for PTP over UDP/IP Feature is selected
22	RO	0x0	reserved
21	RO	0x0	RAVSEL Rx Side Only AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected. Values: 1'b0: Rx Side Only AV Feature is not selected 1'b1: Rx Side Only AV Feature is selected
20	RO	0x0	AVSEL AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. Values: 1'b0: AV Feature is not selected 1'b1: AV Feature is selected
19	RO	0x1	DBGMEMA DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected. Values: 1'b0: DMA Debug Registers option is not selected 1'b1: DMA Debug Registers option is selected

Bit	Attr	Reset Value	Description
18	RO	0x1	TSOEN TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected. Values: 1'b0: TCP Segmentation Offload Feature is not selected 1'b1: TCP Segmentation Offload Feature is selected
17	RO	0x1	SPHEN Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected. Values: 1'b0: Split Header Feature is not selected 1'b1: Split Header Feature is selected
16	RO	0x0	DCBEN DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is selected. Values: 1'b0: DCB Feature is not selected 1'b1: DCB Feature is selected
15:14	RO	0x1	ADDR64 Address Width This field indicates the configured address width: Values: 2'b00: 32 2'b01: 40 2'b10: 48 2'b11: Reserved
13	RO	0x0	reserved
12	RO	0x0	PTOEN PTP Offload Enable This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected. Values: 1'b0: PTP Offload feature is not selected 1'b1: PTP Offload feature is selected
11	RO	0x0	OSTEN One-Step Timestamping Enable This bit is set to 1 when the Enable One-Step Timestamp Feature is selected. Values: 1'b0: One-Step Timestamping feature is not selected 1'b1: One-Step Timestamping feature is selected

Bit	Attr	Reset Value	Description
10:6	RO	0x05	TXFIFOSIZE MTL Transmit FIFO Size This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(TXFIFO_SIZE) -7: Values: 0x0 (128B): 128 bytes 0x1 (256B): 256 bytes 0x2 (512B): 512 bytes 0x3 (1024B): 1024 bytes 0x4 (2048B): 2048 bytes 0x5 (4096B): 4096 bytes 0x6 (8192B): 8192 bytes 0x7 (16384B): 16384 bytes 0x8 (32KB): 32 KB 0x9 (64KB): 64 KB 0xa (128KB): 128 KB 0xb (RSVD): Reserved
5	RO	0x1	SPRAM Single Port RAM Enable This bit is set to 1 when the Use single port RAM Feature is selected. Values: 1'b0: Single Port RAM feature is not selected 1'b1: Single Port RAM feature is selected
4:0	RO	0x06	RXFIFOSIZE MTL Receive FIFO Size This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, Log2(RXFIFO_SIZE) -7: Values: 5'b00000: 128 bytes 5'b00001: 256 bytes 5'b00010: 512 bytes 5'b00011: 1024 bytes 5'b00100: 2048 bytes 5'b00101: 4096 bytes 5'b00110: 8192 bytes 5'b00111: 16384 bytes 5'b01000: 32 KB 5'b01001: 64 KB 5'b01010: 128 KB 5'b01011: 256 KB 5'b01100: Reserved

GMAC_MAC_HW_Feature2

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:28	RO	0x1	AUXSNAPNUM Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: Values: 3'b000: No auxiliary input 3'b001: 1 auxiliary input 3'b010: 2 auxiliary input 3'b011: 3 auxiliary input 3'b100: 4 auxiliary input 3'b101: Reserved
27	RO	0x0	reserved
26:24	RO	0x0	PPSOUTNUM Number of PPS Outputs This field indicates the number of PPS outputs: Values: 3'b000: No PPS output 3'b001: 1 PPS output 3'b010: 2 PPS output 3'b011: 3 PPS output 3'b100: 4 PPS output 3'b101: Reserved
23:22	RO	0x0	reserved
21:18	RO	0x0	TXCHCNT Number of DMA Transmit Channels This field indicates the number of DMA Transmit channels: Values: 4'b0000: 1 MTL Tx Channel 4'b0001: 2 MTL Tx Channels 4'b0010: 3 MTL Tx Channels 4'b0011: 4 MTL Tx Channels 4'b0100: 5 MTL Tx Channels 4'b0101: 6 MTL Tx Channels 4'b0110: 7 MTL Tx Channels 4'b0111: 8 MTL Tx Channels
17:16	RO	0x0	reserved
15:12	RO	0x0	RXCHCNT Number of DMA Receive Channels This field indicates the number of DMA Receive channels: Values: 4'b0000: 1 MTL Rx Channel 4'b0001: 2 MTL Rx Channels 4'b0010: 3 MTL Rx Channels 4'b0011: 4 MTL Rx Channels 4'b0100: 5 MTL Rx Channels 4'b0101: 6 MTL Rx Channels 4'b0110: 7 MTL Rx Channels 4'b0111: 8 MTL Rx Channels
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:6	RO	0x0	TXQCNT Number of MTL Transmit Queues This field indicates the number of MTL Transmit queues: Values: 4'b0000: 1 MTL Tx Queue 4'b0001: 2 MTL Tx Queues 4'b0010: 3 MTL Tx Queues 4'b0011: 4 MTL Tx Queues 4'b0100: 5 MTL Tx Queues 4'b0101: 6 MTL Tx Queues 4'b0110: 7 MTL Tx Queues 4'b0111: 8 MTL Tx Queues
5:4	RO	0x0	reserved
3:0	RO	0x0	RXQCNT Number of MTL Receive Queues This field indicates the number of MTL Receive queues: Values: 4'b0000: 1 MTL Rx Queue 4'b0001: 2 MTL Rx Queues 4'b0010: 3 MTL Rx Queues 4'b0011: 4 MTL Rx Queues 4'b0100: 5 MTL Rx Queues 4'b0101: 6 MTL Rx Queues 4'b0110: 7 MTL Rx Queues 4'b0111: 8 MTL Rx Queues

GMAC_MAC_HW_Feature3

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RO	0x0	ASP Automotive Safety Package Following are the encoding for the different Safety features. Values: 2'b00: No Safety features selected 2'b01: Only "ECC protection for external memory" feature is selected 2'b10: All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 2'b11: All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature
27	RO	0x0	TBSSEL Time Based Scheduling Enable This bit is set to 1 when the Time Based Scheduling feature is selected. Values: 1'b0: Time Based Scheduling Enable feature is not selected 1'b1: Time Based Scheduling Enable feature is selected
26	RO	0x0	FPESEL Frame Preemption Enable This bit is set to 1 when the Enable Frame preemption feature is selected. Values: 1'b0: Frame Preemption Enable feature is not selected 1'b1: Frame Preemption Enable feature is selected
25:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:20	RO	0x0	ESTWID Width of the Time Interval field in the Gate Control List This field indicates the width of the Configured Time Interval Field. Values: 2'b00: Width not configured 2'b01: 16 2'b10: 20 2'b11: 24
19:17	RW	0x0	ESTDEP Depth of the Gate Control List This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5. Values: 3'b000: No Depth configured 3'b001: 64 3'b010: 128 3'b011: 256 3'b100: 512 3'b101: 1024 3'b110: Reserved
16	RO	0x0	ESTSEL Enhancements to Scheduling Traffic Enable This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected. Values: 1'b0: Enable Enhancements to Scheduling Traffic feature is not selected 1'b1: Enable Enhancements to Scheduling Traffic feature is selected
15	RO	0x0	reserved
14:13	RO	0x0	FRPES Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. Values: 2'b00: 64 Entries 2'b01: 128 Entries 2'b10: 256 Entries 2'b11: Reserved
12:11	RO	0x0	FRPBS Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. Values: 2'b00: 64 Bytes 2'b01: 128 Bytes 2'b10: 256 Bytes 2'b11: Reserved
10:6	RO	0x00	reserved

Bit	Attr	Reset Value	Description
5	RO	0x0	DVLAN Double VLAN Tag Processing Selected This bit is set to 1 when the Enable Double VLAN Processing Feature is selected. Values: 1'b0: Double VLAN option is not selected 1'b1: Double VLAN option is selected
4	RO	0x0	CBTISEL Queue/Channel based VLAN tag insertion on Tx Enable This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected. Values: 1'b0: Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 1'b1: Enable Queue/Channel based VLAN tag insertion on Tx feature is selected
3	RO	0x0	reserved
2:0	RO	0x0	NRVF Number of Extended VLAN Tag Filters Enabled This field indicates the Number of Extended VLAN Tag Filters selected: Values: 3'b000: No Extended Rx VLAN Filters 3'b001: 4 Extended Rx VLAN Filters 3'b010: 8 Extended Rx VLAN Filters 3'b011: 16 Extended Rx VLAN Filters 3'b100: 24 Extended Rx VLAN Filters 3'b101: 32 Extended Rx VLAN Filters 3'b110: Reserved

GMAC_MAC_MDIO_Address

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	PSE Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. Values: 1'b0: Preamble Suppression disabled 1'b1: Preamble Suppression enabled

Bit	Attr	Reset Value	Description
26	RW	0x0	BTB Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0. Values: 1'b0: Back to Back transactions disabled 1'b1: Back to Back transactions enabled
25:21	RW	0x00	PA Physical Layer Address This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. For RevMII, this field gives the PHY Address of the RevMII module. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.
20:16	RW	0x00	RDA Register/Device Address These bits select the PHY register in selected Clause 22 PHY device. For RevMII, these bits select the CSR register in the RevMII Registers set. These bits select the Device (MMD) in selected Clause 45 capable PHY.
15	RO	0x0	reserved
14:12	RW	0x0	NTC Number of Trailing Clocks This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.

Bit	Attr	Reset Value	Description
11:8	RW	0x0	CR CSR Clock Range The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design: 4'b0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 4'b0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 4'b0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 4'b0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 4'b0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 4'b0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 4'b0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204 4'b0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324 The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz frequency range. When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is above the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks: 4'b1000: CSR clock/4 4'b1001: CSR clock/6 4'b1010: CSR clock/8 4'b1011: CSR clock/10 4'b1100: CSR clock/12 4'b1101: CSR clock/14 4'b1110: CSR clock/16 4'b1111: CSR clock/18 These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.
7:5	RO	0x0	reserved
4	RW	0x0	SKAP Skip Address Packet When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set. Values: 1'b0: Skip Address Packet is disabled 1'b1: Skip Address Packet is enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	GOC_1 GMII Operation Command 1 This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_O is encoded as follows: 2'b00: Reserved 2'b01: Write 2'b10: Post Read Increment Address for Clause 45 PHY 2'b11: Read When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid. Values: 1'b0: GMII Operation Command 1 is disabled 1'b1: GMII Operation Command 1 is enabled
2	RW	0x0	GOC_0 GMII Operation Command 0 This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1. Values: 1'b0: GMII Operation Command 0 is disabled 1'b1: GMII Operation Command 0 is enabled
1	RW	0x0	C45E Clause 45 PHY Enable When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO. Values: 1'b0: Clause 45 PHY is disabled 1'b1: Clause 45 PHY is enabled
0	RW	0x0	GB GMII Busy The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_Address and MAC_MDIO_Data registers as long as this bit is set. For write transfers, the application must first write 16-bit data in the GDl field (and also RA field when C45E is set) in MAC_MDIO_Data register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_Data register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_Data register. Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: GMII Busy is disabled 1'b1: GMII Busy is enabled

GMAC_MAC_MDIO_Data

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	RA Register Address This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.
15:0	RW	0x0000	GD GMII Data This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.

GMAC_MAC_ARP_Address

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ARPPA ARP Protocol Address This field contains the IPv4 Destination Address of the MAC. This address is used for perfect match with the Protocol Address of Target field in the received ARP packet. This field is available only when the Enable IPv4 ARP Offload option is selected.

GMAC_MAC_CSR_SW_Ctrl

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	RCWE Register Clear on Write 1 Enable When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it. When this bit is reset, the access mode of these register fields remain as Clear on Read. Values: 1'b0: Register Clear on Write 1 is disabled 1'b1: Register Clear on Write 1 is enabled

GMAC_MAC_Ext_Cfg1

Address: Operational Base + offset (0x0238)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:8	RW	0x0	SPLM Split Mode These bits indicate the mode of splitting the incoming Rx packets. They are Values: 0x0 (L3L4): Split at L3/L4 header 0x1 (L2OFST): Split at L2 header with an offset. Always Split at SPLOFST bytes from the beginning of Length/Type field of the Frame 0x2 (COMBN): Combination mode: Split similar to SPLM=00 for IP packets that are untagged or tagged and VLAN stripped 0x3 (RSVD): Reserved
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x02	SPLOFST Split Offset These bits indicate the value of offset from the beginning of Length/Type field at which header split should take place when the appropriate SPLM is selected. The reset value of this field is 2 bytes indicating a split at L2 header. Value is in terms of bytes.

GMAC_MAC_Address0_High

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31	RO	0x0	AE Address Enable This bit is always set to 1. Values: 1'b0: This bit must be always set to 1 1'b1: This bit is always set to 1
30:16	RO	0x0000	reserved
15:0	RW	0xffff	ADDRHI MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

GMAC_MAC_Address0_Low

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	ADDRLO MAC Address0[31:0] This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.

GMAC_MMC_Control

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:9	RO	0x000000	reserved
8	RW	0x0	UCDBC Update MMC Counters for Dropped Broadcast Packets Note: The CNTRST bit has a higher priority than the CNTPRST bit. Therefore, when the software tries to set both bits in the same write cycle, all counters are cleared and the CNTPRST bit is not set. When set, the MAC updates all related MMC Counters for Broadcast packets that are dropped because of the setting of the DBF bit of MAC_Packet_Filter register. When reset, the MMC Counters are not updated for dropped Broadcast packets. Values: 1'b0: Update MMC Counters for Dropped Broadcast Packets is disabled 1'b1: Update MMC Counters for Dropped Broadcast Packets is enabled
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	CNTPRSTLVL Full-Half Preset When this bit is low and the CNTPRST bit is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (Half 2KBytes) and all packet-counters gets preset to 0x7FFF_FFF0 (Half 16). When this bit is high and the CNTPRST bit is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (Full 2KBytes) and all packet-counters gets preset to 0xFFFF_FFF0 (Full 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and packet counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0. Values: 1'b0: Full-Half Preset is disabled 1'b1: Full-Half Preset is enabled
4	RW	0x0	CNTPRST Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to the CNTPRSTLVL bit. This bit is cleared automatically after 1 clock cycle. This bit, along with the CNTPRSTLVL bit, is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Counters Preset is disabled 1'b1: Counters Preset is enabled
3	RW	0x0	CNTFREEZ MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received packet. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode. Values: 1'b0: MMC Counter Freeze is disabled 1'b1: MMC Counter Freeze is enabled
2	RW	0x0	RSTONRD Reset on Read When this bit is set, the MMC counters are reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (Bits[7:0]) is read. Values: 1'b0: Reset on Read is disabled 1'b1: Reset on Read is enabled
1	RW	0x0	CNTSTOPRO Counter Stop Rollover When this bit is set, the counter does not roll over to zero after reaching the maximum value. Values: 1'b0: Counter Stop Rollover is disabled 1'b1: Counter Stop Rollover is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	CNTRST Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Counters are not reset 1'b1: All counters are reset

GMAC_MMC_Rx_Interrupt

Address: Operational Base + offset (0x0704)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RO	0x0	RXFOVPIS MMC Receive FIFO Overflow Packet Counter Interrupt Status This bit is set when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive FIFO Overflow Packet Counter Interrupt Status not detected 1'b1: MMC Receive FIFO Overflow Packet Counter Interrupt Status detected
20	RO	0x0	RXPAUSPIS MMC Receive Pause Packet Counter Interrupt Status This bit is set when the rxpausepackets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 0x0 (INACTIVE): MMC Receive Pause Packet Counter Interrupt Status not detected 0x1 (ACTIVE): MMC Receive Pause Packet Counter Interrupt Status detected
19	RO	0x0	reserved
18	RO	0x0	RXLENERPIS MMC Receive Length Error Packet Counter Interrupt Status This bit is set when the rxlengtherror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Length Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive Length Error Packet Counter Interrupt Status detected
17:6	RO	0x000	reserved

Bit	Attr	Reset Value	Description
5	RO	0x0	RXCRCERPIS MMC Receive CRC Error Packet Counter Interrupt Status This bit is set when the rxcrcerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive CRC Error Packet Counter Interrupt Status not detecte 1'b1: MMC Receive CRC Error Packet Counter Interrupt Status detected
4	RO	0x0	RXMCGPIS MMC Receive Multicast Good Packet Counter Interrupt Status This bit is set when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Multicast Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive Multicast Good Packet Counter Interrupt Status detected
3	RO	0x0	reserved
2	RO	0x0	RXGOCTIS MMC Receive Good Octet Counter Interrupt Status This bit is set when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Octet Counter Interrupt Status not detected 1'b1: MMC Receive Good Octet Counter Interrupt Status detected
1	RO	0x0	RXGBOCTIS MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Bad Octet Counter Interrupt Status not detected 1'b1: MMC Receive Good Bad Octet Counter Interrupt Status detected
0	RO	0x0	RXGBPKTIS MMC Receive Good Bad Packet Counter Interrupt Status This bit is set when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive Good Bad Packet Counter Interrupt Status not detected 1'b1: MMC Receive Good Bad Packet Counter Interrupt Status detected

GMAC_MMC_Tx_Interrupt

Address: Operational Base + offset (0x0708)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23	RO	0x0	TXPAUSPIS MMC Transmit Pause Packet Counter Interrupt Status This bit is set when the txpausepacketserror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 0x0 (INACTIVE): MMC Transmit Pause Packet Counter Interrupt Status not detected 0x1 (ACTIVE): MMC Transmit Pause Packet Counter Interrupt Status detected
22	RO	0x0	reserved
21	RO	0x0	TXGPKTIS MMC Transmit Good Packet Counter Interrupt Status This bit is set when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Packet Counter Interrupt Status detected
20	RO	0x0	TXGOCTIS MMC Transmit Good Octet Counter Interrupt Status This bit is set when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Octet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Octet Counter Interrupt Status detected
19	RO	0x0	TXCARERPIS MMC Transmit Carrier Error Packet Counter Interrupt Status This bit is set when the txcarriererror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Carrier Error Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Carrier Error Packet Counter Interrupt Status detected
18:14	RO	0x00	reserved

Bit	Attr	Reset Value	Description
13	RO	0x0	TXUFLOWERPIS MMC Transmit Underflow Error Packet Counter Interrupt Status This bit is set when the txunderflowerror counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Underflow Error Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Underflow Error Packet Counter Interrupt Status detected
12:2	RO	0x000	reserved
1	RO	0x0	TXGBPKTIS MMC Transmit Good Bad Packet Counter Interrupt Status This bit is set when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Bad Packet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Bad Packet Counter Interrupt Status detected
0	RO	0x0	TXGBOCTIS MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Transmit Good Bad Octet Counter Interrupt Status not detected 1'b1: MMC Transmit Good Bad Octet Counter Interrupt Status detected

GMAC_MMC_Rx_Interrupt_Mask

Address: Operational Base + offset (0x070C)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RW	0x0	RXFOVPIM MMC Receive FIFO Overflow Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxfifooverflow counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive FIFO Overflow Packet Counter Interrupt Mask is enabled
20:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RW	0x0	RXLENERPIM MMC Receive Length Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxlengtherror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Length Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Length Error Packet Counter Interrupt Mask is enabled
17:11	RO	0x00	reserved
10	RW	0x0	RXPAUSPIM MMC Receive Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpausepackets counter reaches half of the maximum value or the maximum value. Values: 0x0 (DISABLE): MMC Receive Pause Packet Counter Interrupt Mask is disabled 0x1 (ENABLE): MMC Receive Pause Packet Counter Interrupt Mask is enabled
9:6	RO	0x0	reserved
5	RW	0x0	RXCRCERPIM MMC Receive CRC Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxcrcerror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive CRC Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive CRC Error Packet Counter Interrupt Mask is enabled
4	RW	0x0	RXMCGPIM MMC Receive Multicast Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxmulticastpackets_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Multicast Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Multicast Good Packet Counter Interrupt Mask is enabled
3	RO	0x0	reserved
2	RW	0x0	RXGOCTIM MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Octet Counter Interrupt Mask is enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	RXGBOCTIM MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Bad Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Bad Octet Counter Interrupt Mask is enabled
0	RW	0x0	RXGBPKTIM MMC Receive Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxpacketcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive Good Bad Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive Good Bad Packet Counter Interrupt Mask is enabled

GMAC_MMC_Tx_Interrupt_Mask

Address: Operational Base + offset (0x0710)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23	RW	0x0	TXPAUSPIM MMC Transmit Pause Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpausepackets counter reaches half of the maximum value or the maximum value. Values: 0x0 (DISABLE): MMC Transmit Pause Packet Counter Interrupt Mask is disabled 0x1 (ENABLE): MMC Transmit Pause Packet Counter Interrupt Mask is enabled
22	RO	0x0	reserved
21	RW	0x0	TXGPKTIM MMC Transmit Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Packet Counter Interrupt Mask is enabled
20	RW	0x0	TXGOCTIM MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_g counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Octet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Octet Counter Interrupt Mask is enabled

Bit	Attr	Reset Value	Description
19	RW	0x0	TXCARERPIM MMC Transmit Carrier Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txcarriererror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Carrier Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Carrier Error Packet Counter Interrupt Mask is enabled
18:14	RO	0x00	reserved
13	RW	0x0	TXUFLOWERPIM MMC Transmit Underflow Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txunderflowerror counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Underflow Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Underflow Error Packet Counter Interrupt Mask is enabled
12:2	RO	0x000	reserved
1	RW	0x0	TXGBPKTIM MMC Transmit Good Bad Packet Counter Interrupt Mask Setting this bit masks the interrupt when the txpacketcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Bad Packet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Bad Packet Counter Interrupt Mask is enabled
0	RW	0x0	TXGBOCTIM MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Transmit Good Bad Octet Counter Interrupt Mask is disabled 1'b1: MMC Transmit Good Bad Octet Counter Interrupt Mask is enabled

GMAC_Tx_Octet_Count_Good_Bad

Address: Operational Base + offset (0x0714)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXOCTGB Tx Octet Count Good Bad This field indicates the number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad packets.

GMAC_Tx_Packet_Count_Good_Bad

Address: Operational Base + offset (0x0718)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXPKTGB Tx Packet Count Good Bad This field indicates the number of good and bad packets transmitted, exclusive of retried packets.

GMAC_Tx_Underflow_Error_Packets

Address: Operational Base + offset (0x0748)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXUNDRFLW Tx Underflow Error Packets This field indicates the number of packets aborted because of packets underflow error.

GMAC_Tx_Carrier_Error_Packets

Address: Operational Base + offset (0x0760)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXCARR Tx Carrier Error Packets This field indicates the number of packets aborted because of carrier sense error (no carrier or loss of carrier).

GMAC_Tx_Octet_Count_Good

Address: Operational Base + offset (0x0764)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXOCTG Tx Octet Count Good This field indicates the number of bytes transmitted, exclusive of preamble, only in good packets.

GMAC_Tx_Packet_Count_Good

Address: Operational Base + offset (0x0768)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXPKTG Tx Packet Count Good This field indicates the number of good packets transmitted.

GMAC_Tx_Pause_Packets

Address: Operational Base + offset (0x0770)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXPAUSE Tx Pause Packets This field indicates the number of good Pause packets transmitted.

GMAC_Rx_Packets_Count_Good_Bad

Address: Operational Base + offset (0x0780)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXPKTGB Rx Packets Count Good Bad This field indicates the number of good and bad packets received.

GMAC_Rx_Octet_Count_Good_Bad

Address: Operational Base + offset (0x0784)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXOCTGB Rx Octet Count Good Bad This field indicates the number of bytes received, exclusive of preamble, in good and bad packets.

GMAC_Rx_Octet_Count_Good

Address: Operational Base + offset (0x0788)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXOCTG Rx Octet Count Good This field indicates the number of bytes received, exclusive of preamble, only in good packets.

GMAC_Rx_Multicast_Packets_Good

Address: Operational Base + offset (0x0790)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXMCASTG Rx Multicast Packets Good This field indicates the number of good multicast packets received.

GMAC_Rx_CRC_Error_Packets

Address: Operational Base + offset (0x0794)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXCRCERR Rx CRC Error Packets This field indicates the number of packets received with CRC error.

GMAC_Rx_Length_Error_Packets

Address: Operational Base + offset (0x07C8)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXLENERR Rx Length Error Packets This field indicates the number of packets received with length error (Length Type field not equal to packet size), for all packets with valid length field.

GMAC_Rx_Pause_Packets

Address: Operational Base + offset (0x07D0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXPAUSEPKT Rx Pause Packets This field indicates the number of good and valid Pause packets received.

GMAC_Rx_FIFO_Overflow_Packets

Address: Operational Base + offset (0x07D4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXFIFOOVFL Rx FIFO Overflow Packets This field indicates the number of missed received packets because of FIFO overflow.

GMAC_MMC_IPC_Rx_Interrupt_Mask

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	RXICMPEROIM MMC Receive ICMP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive ICMP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive ICMP Error Octet Counter Interrupt Mask is enabled
28	RO	0x0	reserved
27	RW	0x0	RXTCPEROIM MMC Receive TCP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive TCP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive TCP Error Octet Counter Interrupt Mask is enabled
26	RO	0x0	reserved
25	RW	0x0	RXUDPEROIM MMC Receive UDP Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive UDP Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive UDP Error Octet Counter Interrupt Mask is enabled
24:23	RO	0x0	reserved
22	RW	0x0	RXIPV6HEROIM MMC Receive IPV6 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_nopay_octets counter reaches half of the maximum value or the maximum value. Value: 1'b0: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Header Error Octet Counter Interrupt Mask is enabled
21:18	RO	0x0	reserved
17	RW	0x0	RXIPV4HEROIM MMC Receive IPV4 Header Error Octet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV4 Header Error Octet Counter Interrupt Mask is enabled

Bit	Attr	Reset Value	Description
16:14	RO	0x0	reserved
13	RW	0x0	RXICMPERPIM MMC Receive ICMP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive ICMP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive ICMP Error Packet Counter Interrupt Mask is enabled
12	RO	0x0	reserved
11	RW	0x0	RXTCPERPIM MMC Receive TCP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive TCP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive TCP Error Packet Counter Interrupt Mask is enabled
10	RO	0x0	reserved
9	RW	0x0	RXUDPERPIM MMC Receive UDP Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive UDP Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive UDP Error Packet Counter Interrupt Mask is enabled
8:7	RO	0x0	reserved
6	RW	0x0	RXIPV6HERPIM MMC Receive IPV6 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Header Error Packet Counter Interrupt Mask is enabled
5	RW	0x0	RXIPV6GPIM MMC Receive IPV6 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV6 Good Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV6 Good Packet Counter Interrupt Mask is enabled
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	RXIPV4HERPIM MMC Receive IPV4 Header Error Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is disabled 1'b1: MMC Receive IPV4 Header Error Packet Counter Interrupt Mask is enabled
0	RW	0x0	RXIPV4GPIM MMC Receive IPV4 Good Packet Counter Interrupt Mask Setting this bit masks the interrupt when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Values: 1'b0: MMC Receive IPV4 Good Packet Counter Interrupt Mask is disable 1'b1: MMC Receive IPV4 Good Packet Counter Interrupt Mask is enabled

GMAC_MMC_IPC_Rx_Interrupt

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RO	0x0	RXICMPEROIS MMC Receive ICMP Error Octet Counter Interrupt Status This bit is set when the rxicmp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive ICMP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive ICMP Error Octet Counter Interrupt Status detected
28	RO	0x0	reserved
27	RO	0x0	RXTCPEROIS MMC Receive TCP Error Octet Counter Interrupt Status This bit is set when the rxtcp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive TCP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive TCP Error Octet Counter Interrupt Status detected
26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25	RO	0x0	RXUDPEROIS MMC Receive UDP Error Octet Counter Interrupt Status This bit is set when the rxudp_err_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive UDP Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive UDP Error Octet Counter Interrupt Status detected
24:23	RO	0x0	reserved
22	RO	0x0	RXIPV6HEROIS MMC Receive IPV6 Header Error Octet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Header Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Header Error Octet Counter Interrupt Status detected
21:18	RO	0x0	reserved
17	RO	0x0	RXIPV4HEROIS MMC Receive IPV4 Header Error Octet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_octets counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Header Error Octet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Header Error Octet Counter Interrupt Status detected
16:14	RO	0x0	reserved
13	RO	0x0	RXICMPERPIS MMC Receive ICMP Error Packet Counter Interrupt Status This bit is set when the rxicmp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive ICMP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive ICMP Error Packet Counter Interrupt Status detected
12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RO	0x0	RXTCPERPIS MMC Receive TCP Error Packet Counter Interrupt Status This bit is set when the rxtcp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive TCP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive TCP Error Packet Counter Interrupt Status detected
10	RO	0x0	reserved
9	RO	0x0	RXUDPERPIS MMC Receive UDP Error Packet Counter Interrupt Status This bit is set when the rxudp_err_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive UDP Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive UDP Error Packet Counter Interrupt Status detected
8:7	RO	0x0	reserved
6	RO	0x0	RXIPV6HERPIS MMC Receive IPV6 Header Error Packet Counter Interrupt Status This bit is set when the rxipv6_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Header Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Header Error Packet Counter Interrupt Status detected
5	RO	0x0	RXIPV6GPIS MMC Receive IPV6 Good Packet Counter Interrupt Status This bit is set when the rxipv6_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV6 Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV6 Good Packet Counter Interrupt Status detected
4:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RO	0x0	RXIPV4HERPIS MMC Receive IPV4 Header Error Packet Counter Interrupt Status This bit is set when the rxipv4_hdrerr_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Header Error Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Header Error Packet Counter Interrupt Status detected
0	RO	0x0	RXIPV4GPIS MMC Receive IPV4 Good Packet Counter Interrupt Status This bit is set when the rxipv4_gd_pkts counter reaches half of the maximum value or the maximum value. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: MMC Receive IPV4 Good Packet Counter Interrupt Status not detected 1'b1: MMC Receive IPV4 Good Packet Counter Interrupt Status detected

GMAC_RxIPv4_Good_Packets

Address: Operational Base + offset (0x0810)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4GDPKT RxIPv4 Good Packets This field indicates the number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload.

GMAC_RxIPv4_Header_Error_Packets

Address: Operational Base + offset (0x0814)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4HDRERRPKT RxIPv4 Header Error Packets This field indicates the number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors.

GMAC_RxIPv6_Good_Packets

Address: Operational Base + offset (0x0824)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6GDPKT RxIPv6 Good Packets This field indicates the number of good IPv6 datagrams received with the TCP, UDP, or ICMP payload.

GMAC_RxIPv6_Header_Error_Packets

Address: Operational Base + offset (0x0828)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6HDRERRPKT RxIPv6 Header Error Packets This field indicates the number of IPv6 datagrams received with header (length or version mismatch) errors.

GMAC_RxUDP_Error_Packets

Address: Operational Base + offset (0x0834)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXUDPERRPKT RxUDP Error Packets This field indicates the number of good IP datagrams received whose UDP payload has a checksum error.

GMAC_RxTCP_Error_Packets

Address: Operational Base + offset (0x083C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXTCPERRPKT RxTCP Error Packets This field indicates the number of good IP datagrams received whose TCP payload has a checksum error.

GMAC_RxICMP_Error_Packets

Address: Operational Base + offset (0x0844)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXICMPERRPKT RxICMP Error Packets This field indicates the number of good IP datagrams received whose ICMP payload has a checksum error.

GMAC_RxIPv4_Header_Error_Octets

Address: Operational Base + offset (0x0854)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV4HDRERROCT RxIPv4 Header Error Octets This field indicates the 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. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

GMAC_RxIPv6_Header_Error_Octets

Address: Operational Base + offset (0x0868)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXIPV6HDRERROCT RxIPv6 Header Error Octets This field indicates the number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the Length field of IPv6 header is used to update this counter. (Ethernet header, FCS, pad, or IP pad bytes are not included in this counter.

GMAC_RxUDP_Error_Octets

Address: Operational Base + offset (0x0874)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXUDPERROCT RxUDP Error Octets This field indicates the number of bytes received in a UDP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_RxTCP_Error_Octets

Address: Operational Base + offset (0x087C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXTCPERROCT RxTCP Error Octets This field indicates the number of bytes received in a TCP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_RxICMP_Error_Octets

Address: Operational Base + offset (0x0884)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXICMPERROCT RxICMP Error Octets This field indicates the number of bytes received in a ICMP segment that had checksum errors. This counter does not count IP header bytes.

GMAC_MTL_Operation_Mode

Address: Operational Base + offset (0x0C00)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9	RW	0x0	CNTCLR Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 0x0 (DISABLE): Counters are not reset 0x1 (ENABLE): All counters are reset
8	RW	0x0	CNTPRST Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle. If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: iö 0x0 (DISABLE): Counters are not reset iö 0x1 (ENABLE): All counters are reset
7:2	RO	0x00	reserved
1	RW	0x0	DTXSTS Drop Transmit Status When this bit is set, the Tx packet status received from the MAC is dropped in the MTL. When this bit is reset, the Tx packet status received from the MAC is forwarded to the application. Values: iö 0x0 (DISABLE): Drop Transmit Status is disabled iö 0x1 (ENABLE): Drop Transmit Status is enabled
0	RO	0x0	reserved

GMAC_MTL_DBG_CTL

Address: Operational Base + offset (0x0C08)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	STSIE Transmit Status Available Interrupt Status Enable When this bit is set, an interrupt is generated when Transmit status is available in slave mode. Values: 1'b0: Transmit Packet Available Interrupt Status is disabled 1'b1: Transmit Packet Available Interrupt Status is enabled
14	RW	0x0	PKTIE Receive Packet Available Interrupt Status Enable When this bit is set, an interrupt is generated when EOP of received packet is written to the Rx FIFO. Values: 1'b0: Receive Packet Available Interrupt Status is disabled 1'b1: Receive Packet Available Interrupt Status is enabled
13:12	RW	0x0	FIFOSEL FIFO Selected for Access This field indicates the FIFO selected for debug access: Values: 2'b00: Tx FIFO 2'b01: Tx Status FIFO (only read access when SLVMOD is set) 2'b10: TSO FIFO (cannot be accessed when SLVMOD is set) 2'b11: Rx FIFO
11	RW	0x0	FIFOWREN FIFO Write Enable When this bit is set, it enables the Write operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: FIFO Write is disabled 1'b1: FIFO Write is enabled
10	RW	0x0	FIFORDEN FIFO Read Enable When this bit is set, it enables the Read operation on selected FIFO when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: FIFO Read is disabled 1'b1: FIFO Read is enabled
9	RW	0x0	RSTSEL Reset Pointers of Selected FIFO When this bit is set, the pointers of the currently-selected FIFO are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Reset Pointers of Selected FIFO is disabled 1'b1: Reset Pointers of Selected FIFO is enabled

Bit	Attr	Reset Value	Description
8	RW	0x0	RSTALL Reset All Pointers When this bit is set, the pointers of all FIFOs are reset when FIFO Debug Access is enabled. This bit must not be written to 1 when FIFO Debug Access is not enabled, that is FDBGEN bit is 0. Access restriction applies. Self-cleared. Setting 0 clears. Setting 1 sets. Values: 1'b0: Reset All Pointers is disabled 1'b1: Reset All Pointers is enabled
7	RO	0x0	reserved
6:5	RW	0x0	PKTSTATE Encoded Packet State This field is used to write the control information to the Tx FIFO or Rx FIFO. Tx FIFO: 2'b00: Packet Data 2'b01: Control Word 2'b10: SOP Data 2'b11: EOP Data Rx FIFO: 2'b00: Packet Data 2'b01: Normal Status 2'b10: Last Status 2'b11: EOP Values: 2'b00: Packet Data 2'b01: Control Word/Normal Status 2'b10: SOP Data/Last Status 2'b11: EOP Data/EOP
4	RO	0x0	reserved
3:2	RW	0x0	BYTEEN Byte Enables This field indicates the number of data bytes valid in the data register during Write operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. Values: 2'b00: Byte 0 valid 2'b01: Byte 0 and Byte 1 are valid 2'b10: Byte 0, Byte 1, and Byte 2 are valid 2'b11: All four bytes are valid
1	RW	0x0	DBGMOD Debug Mode Access to FIFO When this bit is set, it indicates that the current access to the FIFO is read, write, and debug access. In this mode, the following access types are allowed: 1. Read and Write access to Tx FIFO, TSO FIFO, and Rx FIFO 2. Read access is allowed to Tx Status FIFO. When this bit is reset, it indicates that the current access to the FIFO is slave access bypassing the DMA. In this mode, the following access are allowed: 1. Write access to the Tx FIFO 2. Read access to the Rx FIFO and Tx Status FIFO Values: 1'b0: Debug Mode Access to FIFO is disabled 1'b1: Debug Mode Access to FIFO is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	FDBGEN FIFO Debug Access Enable When this bit is set, it indicates that the debug mode access to the FIFO is enabled. When this bit is reset, it indicates that the FIFO can be accessed only through a master interface. Values: 1'b0: FIFO Debug Access is disabled 1'b1: FIFO Debug Access is enabled

GMAC_MTL_DBG_STS

Address: Operational Base + offset (0x0C0C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RO	0x320	LOCR Remaining Locations in the FIFO Slave Access Mode: This field indicates the space available in selected FIFO. Debug Access Mode: This field contains the Write or Read pointer value of the selected FIFO during Write or Read operation, respectively. Reset: In single Tx Queue configurations, (DWC_EQOS_TXFIFO_SIZE/(DWC_EQOS_DATAWIDTH/8)), Otherwise 0000H.
14:10	RO	0x00	reserved
9	RW	0x0	STSI Transmit Status Available Interrupt Status When set, this bit indicates that the Slave mode Tx packet is transmitted, and the status is available in Tx Status FIFO. This bit is reset when 1 is written to this bit. Values: 1'b0: Transmit Status Available Interrupt Status not detected 1'b1: Transmit Status Available Interrupt Status detected
8	RW	0x0	PKTI Receive Packet Available Interrupt Status When set, this bit indicates that MAC layer has written the EOP of received packet to the Rx FIFO. This bit is reset when 1 is written to this bit. Values: 1'b0: Receive Packet Available Interrupt Status not detected 1'b1: Receive Packet Available Interrupt Status detected
7:5	RO	0x0	reserved
4:3	RO	0x0	BYTEEN Byte Enables This field indicates the number of data bytes valid in the data register during Read operation. This is valid only when PKTSTATE is 2'b10 (EOP) and Tx FIFO or Rx FIFO is selected. Values: 2'b00: Byte 0 valid 2'b01: Byte 0 and Byte 1 are valid 2'b10: Byte 0, Byte 1, and Byte 2 are valid 2'b11: All four bytes are valid

Bit	Attr	Reset Value	Description
2:1	RO	0x0	PKTSTATE Encoded Packet State This field is used to get the control or status information of the selected FIFO. Tx FIFO: 2'b00: Packet Data 2'b01: Control Word 2'b10: SOP Data 2'b11: EOP Data Rx FIFO: 2'b00: Packet Data 2'b01: Normal Status 2'b10: Last Status 2'b11: EOP This field is applicable only for Tx FIFO and Rx FIFO during Read operation. Values: 2'b00: Packet Data 2'b01: Control Word/Normal Status 2'b10: SOP Data/Last Status 2'b11: EOP Data/EOP
0	RO	0x0	FIFOBUSY FIFO Busy When set, this bit indicates that a FIFO operation is in progress in the MAC and content of the following fields is not valid: 1. All other fields of this register 2. All fields of the MTL_FIFO_Debug_Data register Values: 1'b0: FIFO Busy not detected 1'b1: FIFO Busy detected

GMAC_MTL_FIFO_Debug_Data

Address: Operational Base + offset (0x0C10)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FDBGDATA FIFO Debug Data During debug or slave access write operation, this field contains the data to be written to the Tx FIFO, Rx FIFO, or TSO FIFO. During debug or slave access read operation, this field contains the data read from the Tx FIFO, Rx FIFO, TSO FIFO, or Tx Status FIFO.

GMAC_MTL_Interrupt_Status

Address: Operational Base + offset (0x0C20)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	DBGIS Debug Interrupt status This bit indicates an interrupt event during the slave access. To reset this bit, the application must read the FIFO Debug Access Status register to get the exact cause of the interrupt and clear its source. Values: 1'b0: Debug Interrupt status not detected 1'b1: Debug Interrupt status detected
16:1	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	Q0IS Queue 0 Interrupt status This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source. Values: 1'b0: Queue 0 Interrupt status not detected 1'b1: Queue 0 Interrupt status detected

GMAC_MTL_TxQ0_Operation_Mode

Address: Operational Base + offset (0x0D00)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	TTC Transmit Threshold Control These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset. Values: 3'b000: 32 bytes 3'b001: 64 bytes 3'b010: 96 bytes 3'b011: 128 bytes 3'b100: 192 bytes 3'b101: 256 bytes 3'b110: 384 bytes 3'b111: 512 bytes
3:2	RO	0x0	reserved
1	RW	0x0	TSF Transmit Store and Forward When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped. Values: 1'b0: Transmit Store and Forward is disabled 1'b1: Transmit Store and Forward is enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	FTQ Flush Transmit Queue When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TxQ1_Operation_Mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission. Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Flush Transmit Queue is disabled 1'b1: Flush Transmit Queue is enabled

GMAC_MTL_TxQ0_Underflow

Address: Operational Base + offset (0x0D04)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11	RO	0x0	UFCNTOVF Overflow Bit for Underflow Packet Counter This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Overflow not detected for Underflow Packet Counter 1'b1: Overflow detected for Underflow Packet Counter
10:0	RO	0x000	UFFRMCNT Underflow Packet Counter This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

GMAC_MTL_TxQ0_Debug

Address: Operational Base + offset (0x0D08)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22:20	RO	0x0	STXSTSF Number of Status Words in Tx Status FIFO of Queue This field indicates the current number of status in the Tx Status FIFO of this queue.When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.
19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:16	RO	0x0	PTXQ Number of Packets in the Transmit Queue This field indicates the current number of packets in the Tx Queue. When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.
15:6	RO	0x000	reserved
5	RO	0x0	TXSTSFSTS MTL Tx Status FIFO Full Status When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission. Values: 1'b0: MTL Tx Status FIFO Full status is not detected 1'b1: MTL Tx Status FIFO Full status is detected
4	RO	0x0	TXQSTS MTL Tx Queue Not Empty Status When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission. Values: 1'b0: MTL Tx Queue Not Empty status is not detected 1'b1: MTL Tx Queue Not Empty status is detected
3	RO	0x0	TWCSTS MTL Tx Queue Write Controller Status When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue. Values: 1'b0: MTL Tx Queue Write Controller status is not detected 1'b1: MTL Tx Queue Write Controller status is detected
2:1	RO	0x0	TRCSTS MTL Tx Queue Read Controller Status This field indicates the state of the Tx Queue Read Controller: Values: 2'b00: Idle state 2'b01: Read state (transferring data to the MAC transmitter) 2'b10: Waiting for pending Tx Status from the MAC transmitter 2'b11: Flushing the Tx queue because of the Packet Abort request from the MAC
0	RO	0x0	TXQPAUSED Transmit Queue in Pause When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following: 1. Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled 2. Reception of 802.3x Pause packet when PFC is disabled Values: 1'b0: Transmit Queue in Pause status is not detected 1'b1: Transmit Queue in Pause status is detected

GMAC_MTL_Q0_Interrupt_Ctrl_Status

Address: Operational Base + offset (0x0D2C)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved

Bit	Attr	Reset Value	Description
24	RW	0x0	RXOIE Receive Queue Overflow Interrupt Enable When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled. Values: 1'b0: Receive Queue Overflow Interrupt is disabled 1'b1: Receive Queue Overflow Interrupt is enabled
23:17	RO	0x00	reserved
16	RW	0x0	RXOVFIS Receive Queue Overflow Interrupt Status This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. Values: 1'b0: Receive Queue Overflow Interrupt Status not detected 1'b1: Receive Queue Overflow Interrupt Status detected
15:9	RO	0x00	reserved
8	RW	0x0	TXUIE Transmit Queue Underflow Interrupt Enable When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled. Values: 1'b0: Transmit Queue Underflow Interrupt Status is disabled 1'b1: Transmit Queue Underflow Interrupt Status is enabled
7:1	RO	0x00	reserved
0	RW	0x0	TXUNFIS Transmit Queue Underflow Interrupt Status This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. Values: 1'b0: Transmit Queue Underflow Interrupt Status not detected 1'b1: Transmit Queue Underflow Interrupt Status detected

GMAC_MTL_RxQ0_Operation_Mode

Address: Operational Base + offset (0x0D30)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
17:14	RW	0x0	RFD Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes) These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation: 0: Full minus 1 KB, that is, FULL 1 KB 1: Full minus 1.5 KB, that is, FULL 1.5 KB 2: Full minus 2 KB, that is, FULL 2 KB 3: Full minus 2.5 KB, that is, FULL 2.5 KB ... 62: Full minus 32 KB, that is, FULL 32 KB 63: Full minus 32.5 KB, that is, FULL 32.5 KB The de-assertion is effective only after flow control is asserted. Note: The value must be programmed in such a way to make sure that the threshold is a positive number. When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB. For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register. The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.
13:12	RO	0x0	reserved
11:8	RW	0x0	RFA Threshold for Activating Flow Control (in half-duplex and full-duplex) These bits control the threshold (fill-level of Rx queue) at which the flow control is activated:For more information on encoding for this field, see RFD.
7	RW	0x0	EHFC Enable Hardware Flow Control When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled. Values: 1'b0: Hardware Flow Control is disabled 1'b1: Hardware Flow Control is enabled
6	RW	0x0	DIS_TCP_EF Disable Dropping of TCP/IP Checksum Error Packets When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC. When this bit is reset, all error packets are dropped if the FEP bit is reset. Values: 1'b0: Dropping of TCP/IP Checksum Error Packets is enabled 1'b1: Dropping of TCP/IP Checksum Error Packets is disabled

Bit	Attr	Reset Value	Description
5	RW	0x0	RSF Receive Queue Store and Forward When this bit is set, the GMAC reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register. Values: 1'b0: Receive Queue Store and Forward is disabled 1'b1: Receive Queue Store and Forward is enabled
4	RW	0x0	FEP Forward Error Packets When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped. When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet may be forwarded to the application or DMA. Values: 1'b0: Forward Error Packets is disabled 1'b1: Forward Error Packets is enabled
3	RW	0x0	FUP Forward Undersized Good Packets When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01. Values: 1'b0: Forward Undersized Good Packets is disabled 1'b1: Forward Undersized Good Packets is enabled
2	RO	0x0	reserved
1:0	RW	0x0	RTC Receive Queue Threshold Control These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred. This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1.

GMAC_MTL_RxQ0_Miss_Pkt_Ovf_Cnt

Address: Operational Base + offset (0x0D34)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RO	0x0	MISCNTOVF Missed Packet Counter Overflow Bit When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Missed Packet Counter overflow not detected 1'b1: Missed Packet Counter overflow detected
26:16	RO	0x000	MISPKTCNT Missed Packet Counter This field indicates the number of packets missed by the GMAC because the application asserted ari_pkt_flush_i[] for this queue. This counter is incremented each time the application issues ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1. In EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations, This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability. Access restriction applies. Clears on read. Self-set to 1 on internal event.
15:12	RO	0x0	reserved
11	RO	0x0	OVFCNTOVF Overflow Counter Overflow Bit When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit. Access restriction applies. Clears on read. Self-set to 1 on internal event. Values: 1'b0: Overflow Counter overflow not detected 1'b1: Overflow Counter overflow detected
10:0	RO	0x000	OVFPKTCNT Overflow Packet Counter This field indicates the number of packets discarded by the GMAC because of Receive queue overflow. This counter is incremented each time the GMAC discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1. Access restriction applies. Clears on read. Self-set to 1 on internal event.

GMAC_MTL_RxQ0_Debug

Address: Operational Base + offset (0x0D38)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:16	RO	0x0	PRXQ Number of Packets in Receive Queue This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.
15:6	RO	0x000	reserved

Bit	Attr	Reset Value	Description
5:4	RO	0x0	RXQSTS MTL Rx Queue Fill-Level Status This field gives the status of the fill-level of the Rx Queue: Values: 2'b00: Rx Queue empty 2'b01: Rx Queue fill-level below flow-control deactivate threshold 2'b10: Rx Queue fill-level above flow-control activate threshold 2'b11: Rx Queue full
3	RO	0x0	reserved
2:1	RO	0x0	RRCSTS MTL Rx Queue Read Controller State This field gives the state of the Rx queue Read controller: Values: 2'b00: Idle state 2'b01: Reading packet data 2'b10: Reading packet status (or timestamp) 2'b11: Flushing the packet data and status
0	RO	0x0	RWCSTS MTL Rx Queue Write Controller Active Status When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue. Values: 1'b0: MTL Rx Queue Write Controller Active Status not detected 1'b1: MTL Rx Queue Write Controller Active Status detected

GMAC_DMA_Mode

Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
17:16	RW	0x0	INTM Interrupt Mode This field defines the interrupt mode of GMAC. The behavior of the following outputs changes depending on the following settings: 1. sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) 2. sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) 3. sbd_intr_o (Common Interrupt) It also changes the behavior of the RI/TI bits in the DMA_CH0_Status. 2'b00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits. 2'b01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. 2'b10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events. 2'b11: Reserved Values: 2'b00: See above description 2'b01: See above description 2'b10: See above description 2'b11: Reserved
15:9	RO	0x00	reserved
8	RW	0x0	DSPW Descriptor Posted Write When this bit is set to 0, the descriptor writes are always non-posted. When this bit is set to 1, the descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor, otherwise the descriptor writes are always posted. Values: 1'b0: Descriptor Posted Write is disabled 1'b1: Descriptor Posted Write is enabled
7:1	RO	0x00	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	SWR Software Reset When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all GMAC clock domains. Before reprogramming any GMAC register, a value of zero should be read in this bit. This bit must be read at least 4 CSR clock cycles after it is written to 1. Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock. Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect. Values: 1'b0: Software Reset is disabled 1'b1: Software Reset is enabled

GMAC_DMA_SysBus_Mode

Address: Operational Base + offset (0x1004)

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller. Values: 1'b0: Low Power Interface (LPI) is disabled 1'b1: Low Power Interface (LPI) is enabled
30	RW	0x0	LPI_XIT_PKT Unlock on Magic Packet or Remote Wake-Up Packet When set to 1, this bit enables the AXI master to come out of the LPI mode only when the magic packet or remote wake-up packet is received. When set to 0, this bit enables the AXI master to come out of the LPI mode when any packet is received. Values: 1'b0: Unlock on Magic Packet or Remote Wake-Up Packet is disabled 1'b1: Unlock on Magic Packet or Remote Wake-Up Packet is enabled
29:26	RO	0x0	reserved
25:24	RW	0x0	WR_OSR_LMT AXI Maximum Write Outstanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1 Note: 1. Bit 26 is reserved if GMAC_AXI_MAX-_WR_REQ = 4 2. Bit 27 is reserved if GMAC_AXI_MAX-_WR_REQ!= 16
23:19	RO	0x00	reserved

Bit	Attr	Reset Value	Description
18:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Outstanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1 Note: 1. Bit 18 is reserved if parameter GMAC_AXI_-MAX_RD_REQ = 4 2. Bit 19 is reserved if parameter GMAC_AXI_-MAX_RD_REQ!= 16
15:13	RO	0x0	reserved
12	RW	0x0	AAL Address-Aligned Beats When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels. Values: 1'b0: Address-Aligned Beats is disabled 1'b1: Address-Aligned Beats is enabled
11	RW	0x0	EAME Enhanced Address Mode Enable. When this bit is set to 1, the DMA master enables the enhanced address mode (40-bit or 48-bit addressing mode). In this mode, the DMA engine uses either the 40- or 48-bit address, depending on the configuration. Values: 0x0 (DISABLE): Enhanced Address Mode is disabled 0x1 (ENABLE): Enhanced Address Mode is enabled
10	RW	0x0	AALE Automatic AXI LPI enable When set to 1, enables the AXI master to enter into LPI state when there is no activity in the GMAC for number of system clock cycles programmed in the LPIEI field of AXI_LPI_Entry_Interval register. Values: 1'b0: Automatic AXI LPI is disabled 1'b1: Automatic AXI LPI is enabled
9:4	RO	0x00	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 16
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 8 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 8

Bit	Attr	Reset Value	Description
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 0, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 1'b0: No effect 1'b1: AXI Burst Length 4
0	RW	0x0	FB Fixed Burst Length When this bit is set to 1, the EQOS-AXI master initiates burst transfers of specified lengths as given below. 1. Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field 2. Burst transfers of length 1 When this bit is set to 0, the EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:1]. Values: 1'b0: Fixed Burst Length is disabled 1'b1: Fixed Burst Length is enabled

GMAC_DMA_Interrupt_Status

Address: Operational Base + offset (0x1008)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17	RO	0x0	MACIS MAC Interrupt Status This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source. Values: 1'b0: MAC Interrupt Status not detected 1'b1: MAC Interrupt Status detected
16	RO	0x0	MTLIS MTL Interrupt Status This bit indicates an interrupt event in the MTL. To reset this bit to 1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source. Values: 1'b0: MTL Interrupt Status not detected 1'b1: MTL Interrupt Status detected
15:1	RO	0x0000	reserved
0	RO	0x0	DC0IS DMA Channel 0 Interrupt Status This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source. Values: 1'b0: DMA Channel 0 Interrupt Status not detected 1'b1: DMA Channel 0 Interrupt Status detected

GMAC_DMA_Debug_Status0

Address: Operational Base + offset (0x100C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:12	RO	0x0	TPS0 DMA Channel 0 Transmit Process State This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. Values: 4'b0000: Stopped (Reset or Stop Transmit Command issued) 4'b0001: Running (Fetching Tx Transfer Descriptor) 4'b0010: Running (Waiting for status) 4'b0011: Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 4'b0100: Timestamp write state 4'b0101: Reserved for future use 4'b0110: Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 4'b0111: Running (Closing Tx Descriptor)
11:8	RO	0x0	RPS0 DMA Channel 0 Receive Process State This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt. Values: 4'b0000: Stopped (Reset or Stop Receive Command issued) 4'b0001: Running (Fetching Rx Transfer Descriptor) 4'b0010: Reserved for future use 4'b0011: Running (Waiting for Rx packet) 4'b0100: Suspended (Rx Descriptor Unavailable) 4'b0101: Running (Closing the Rx Descriptor) 4'b0110: Timestamp write state 4'b0111: Running (Transferring the received packet data from the Rx buffer to the system memory)
7:2	RO	0x00	reserved
1	RO	0x0	AXRHSTS AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data. Values: 1'b0: AXI Master Read Channel Status not detected 1'b1: AXI Master Read Channel Status detected
0	RO	0x0	AXWHSTS AXI Master Write Channel or AHB Master Status EQOS-AXI Configuration: When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data. EQOS-AHB Configuration: When high, this bit indicates that the AHB master FSMs are in the non-idle state. Values: 1'b0: AXI Master Write Channel or AHB Master Status not detected 1'b1: AXI Master Write Channel or AHB Master Status detected

GMAC_AXI_LPI_Entry_Interval

Address: Operational Base + offset (0x1040)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	LPIEI LPI Entry Interval Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the GMAC to enter into the AXI low power state. 0 indicates 64 clock cycles.

GMAC_DMA_CH0_Control

Address: Operational Base + offset (0x1100)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	SPH Split Headers When this bit is set, the DMA splits the header and payload in the Receive path. The DMA writes the header to the Buffer Address1 of RDES0. The DMA writes the payload to the buffer to which the Buffer Address2 is pointing. The software must ensure that the header fits into the Receive buffers. If the header length exceeds the receive buffer size, the DMA does not split the header and payload. This bit is available only if Enable Split Header Structure option is selected. Values: 0x0 (DISABLE): Split Headers feature is disabled 0x1 (ENABLE): Split Headers feature is enabled
23:21	RO	0x0	reserved
20:18	RW	0x0	DSL Descriptor Skip Length This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current descriptor to the start of the next descriptor. When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.
17	RO	0x0	reserved
16	RW	0x0	PBLx8 8xPBL mode When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH0_Tx_Control and Bits[21:16] in DMA_CH0_Rx_Control is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value. 0x0 (DISABLE): 8xPBL mode is disabled 0x1 (ENABLE): 8xPBL mode is enabled
15:14	RO	0x0	reserved
13:0	RW	0x0000	MSS Maximum Segment Size This field specifies the maximum segment size that should be used while segmenting the packet. This field is valid only if the TSE bit of DMA_CH0_Tx_Control register is set. The value programmed in this field must be more than the configured Datawidth in bytes. It is recommended to use a MSS value of 64 bytes or more.

GMAC_DMA_CH0_Tx_Control

Address: Operational Base + offset (0x1104)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21:16	RW	0x00	TxPBL Transmit Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in DMA_CH0_Control register. 2. Set the TxPBL. Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of MTL_TxQ[n]_Operation_Mode register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15	RW	0x0	IPBL Ignore PBL Requirement When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL may use handshaking to slow the DMA. Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it may block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer. 0x0 (DISABLE): Ignore PBL Requirement is disabled 0x1 (ENABLE): Ignore PBL Requirement is enabled
14:13	RW	0x0	TSE_MODE TSE Mode 00: TSO/USO (segmentation functionality is enabled). In this mode, the setting of TSE bit enables the TSO/USO segmentation. 01: UFO with Checksum (UDP Fragmentation over IPv4 with Checksum). In this mode, the setting of TSE bit enables the UDP fragmentation functionality with Checksum for all the UDP packets. 10: UFO without Checksum (UDP Fragmentation over IPv4 with Checksum). In this mode, the setting of TSE bit enables the UDP fragmentation functionality without Checksum for all the UDP packets. 11: Reserved 0x0 (TSO_USO): TSO/USO 0x1 (UFOWC): UFO with Checksum 0x2 (UFOWOC): UFO without Checksum 0x3 (RSVD): Reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	TSE TCP Segmentation Enabled When this bit is set, the DMA performs the TCP segmentation or UDP Segmentation/Fragmentation for packets in this channel. The TCP segmentation or UDP packet's segmentation/Fragmentation is done only for those packets for which the TSE bit (TDES0[19]) is set in the Tx Normal descriptor.When this bit is set, the TxPBL value must be greater than 4. 0x0 (DISABLE): TCP Segmentation is disabled 0x1 (ENABLE): TCP Segmentation is enabled
11:5	RO	0x00	reserved
4	RW	0x0	OSF Operate on Second Packet When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained. 0x0 (DISABLE): Operate on Second Packet disabled 0x1 (ENABLE): Operate on Second Packet enabled
3:1	RO	0x0	reserved
0	RW	0x0	ST Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted. The DMA tries to acquire descriptor from either of the following positions: 1.The current position in the list This is the base address of the Transmit list set by the DMA_CH0_TxDesc_List_Address register. 2.The position at which the transmission was previously stopped If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH0_Status register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TxDesc_List_Address register, 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 packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH0_TxDesc_List_Address register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state. 0x0 (STOP): Stop Transmission Command 0x1 (START): Start Transmission Command

GMAC_DMA_CH0_Rx_Control

Address: Operational Base + offset (0x1108)

Bit	Attr	Reset Value	Description
31	RW	0x0	RPF Rx Packet Flush When this bit is set to 1, then GMAC automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, get flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing happens on the Read side of the Rx Queue. When this bit is set to 0, the GMAC not flush the packet in the Rx Queue destined to this RxDMA Channel when it is STOP state. This may in turn cause head-of-line blocking in the corresponding RxQueue. 0x0 (DISABLE): Rx Packet Flush is disabled 0x1 (ENABLE): Rx Packet Flush is enabled
30:22	RO	0x000	reserved
21:16	RW	0x00	RxPBL Receive Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior. To transfer more than 32 beats, perform the following steps: 1. Set the 8xPBL mode in the DMA_CH0_Control register. 2. Set the RxPBL. Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of MTL_RxQ[n]_Operation_Mode register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.
15	RO	0x0	reserved
14:4	RW	0x000	RBSZ_13_y Receive Buffer size High RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_3_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled. Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_3_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).

Bit	Attr	Reset Value	Description
3:1	RO	0x0	RBSZ_3_0 Receive Buffer size Low RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_3_0. The RBSZ_3_0 is the lower field whose width is based on data bus width of the configuration. This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO)
0	RW	0x0	SR Start or Stop Receive When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets.The DMA tries to acquire descriptor from either of the following positions: 1.The current position in the list This is the address set by the DMA_CH0_RxDesc_List_Address register. 2.The position at which the Rx process was previously stopped If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_Status register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RxDesc_List_Address register, the DMA behavior is unpredictable. When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state. 0x0 (STOP): Stop Receive 0x1 (START): Start Receive

GMAC_DMA_CH0_TxDesc_List_HAddress

Address: Operational Base + offset (0x1110)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	TDESHA Start of Transmit List This field contains the most-significant 8 or 16 bits of the 40- or 48-bit base address of the first descriptor in the Transmit descriptor list.

GMAC_DMA_CH0_TxDesc_List_Address

Address: Operational Base + offset (0x1114)

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	TDESLA Start of Transmit List This field contains the base address of the first descriptor in the Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration
2:0	RO	0x0	reserved

GMAC_DMA_CH0_RxDesc_List_HAaddress

Address: Operational Base + offset (0x1118)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	RDESHA Start of Receive List This field contains the most-significant 8 or 16 bits of the 40-bit or 48-bit base address of the first descriptor in the Rx Descriptor list.

GMAC_DMA_CH0_RxDesc_List_Address

Address: Operational Base + offset (0x111C)

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	RDESLA Start of Receive List This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO). The width of this field depends on the configuration: 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration
2:0	RO	0x0	reserved

GMAC_DMA_CH0_TxDesc_Tail_Pointer

Address: Operational Base + offset (0x1120)

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	TDTP Transmit Descriptor Tail Pointer This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers. The width of this field depends on the configuration: 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration
2:0	RO	0x0	reserved

GMAC_DMA_CH0_RxDesc_Tail_Pointer

Address: Operational Base + offset (0x1128)

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	RDRT Receive Descriptor Tail Pointer This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers. The width of this field depends on the configuration: 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration

Bit	Attr	Reset Value	Description
2:0	RO	0x0	reserved

GMAC_DMA_CH0_TxDesc_Ring_Length

Address: Operational Base + offset (0x112C)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:0	RW	0x000	TDRL Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. we recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

GMAC_DMA_CH0_RxDesc_Ring_Length

Address: Operational Base + offset (0x1130)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9:0	RW	0x000	RDRL Receive Descriptor Ring Length This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.

GMAC_DMA_CH0_Interrupt_Enable

Address: Operational Base + offset (0x1134)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_Status register: Bit 0: Transmit Interrupt Bit 2: Transmit Buffer Unavailable Bit 6: Receive Interrupt Bit 11: Early Receive Interrupt When this bit is reset, the normal interrupt summary is disabled. 0x0 (DISABLE): Normal Interrupt Summary is disabled 0x1 (ENABLE): Normal Interrupt Summary is enabled

Bit	Attr	Reset Value	Description
14	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_Status register: Bit 1: Transmit Process Stopped Bit 7: Rx Buffer Unavailable Bit 8: Receive Process Stopped Bit 9: Receive Watchdog Timeout Bit 10: Early Transmit Interrupt Bit 12: Fatal Bus Error Bit 13: Context Descriptor Error When this bit is reset, the abnormal interrupt summary is disabled. 0x0 (DISABLE): Abnormal Interrupt Summary is disabled 0x1 (ENABLE): Abnormal Interrupt Summary is enabled
13	RW	0x0	CDEE Context Descriptor Error Enable When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled. 0x0 (DISABLE): Context Descriptor Error is disabled 0x1 (ENABLE): Context Descriptor Error is enabled
12	RW	0x0	FBEE Fatal Bus Error Enable When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled. 0x0 (DISABLE): Fatal Bus Error is disabled 0x1 (ENABLE): Fatal Bus Error is enabled
11	RW	0x0	ERIE Early Receive Interrupt Enable When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled. 0x0 (DISABLE): Early Receive Interrupt is disabled 0x1 (ENABLE): Early Receive Interrupt is enabled
10	RW	0x0	ETIE Early Transmit Interrupt Enable When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled. 0x0 (DISABLE): Early Transmit Interrupt is disabled 0x1 (ENABLE): Early Transmit Interrupt is enabled
9	RW	0x0	RWTE Receive Watchdog Timeout Enable When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled. 0x0 (DISABLE): Receive Watchdog Timeout is disabled 0x1 (ENABLE): Receive Watchdog Timeout is enabled
8	RW	0x0	RSE Receive Stopped Enable When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled.

Bit	Attr	Reset Value	Description
7	RW	0x0	RBUE Receive Buffer Unavailable Enable When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled. 0x0 (DISABLE): Receive Buffer Unavailable is disabled 0x1 (ENABLE): Receive Buffer Unavailable is enabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled. 0x0 (DISABLE): Receive Interrupt is disabled 0x1 (ENABLE): Receive Interrupt is enabled
5:3	RO	0x0	reserved
2	RW	0x0	TBUE Transmit Buffer Unavailable Enable When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled. 0x0 (DISABLE): Transmit Buffer Unavailable is disabled 0x1 (ENABLE): Transmit Buffer Unavailable is enabled
1	RW	0x0	TXSE Transmit Stopped Enable When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled. 0x0 (DISABLE): Transmit Stopped is disabled 0x1 (ENABLE): Transmit Stopped is enabled
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled. 0x0 (DISABLE): Transmit Interrupt is disabled 0x1 (ENABLE): Transmit Interrupt is enabled

GMAC_DMA_CH0_Rx_Interrupt_Watchdog_Timer

Address: Operational Base + offset (0x1138)

Bit	Attr	Reset Value	Description
31:18	RO	0x0000	reserved
17:16	RW	0x0	RWTU Receive Interrupt Watchdog Timer Count Units This fields indicates the number of system clock cycles corresponding to one unit in RWT field. 2'b00: 256 2'b01: 512 2'b10: 1024 2'b11: 2048 For example, when RWT=2 and RWTU=1, the watchdog timer is set for 2*512=1024 system clock cycles.
15:8	RO	0x00	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	RWT Receive Interrupt Watchdog Timer Count This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set. The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CH0_Status register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.

GMAC_DMA_CH0_Current_App_TxDesc

Address: Operational Base + offset (0x1144)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CURTDESAPTR Application Transmit Descriptor Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Current_App_RxDesc

Address: Operational Base + offset (0x114C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CURRDESAPTR Application Receive Descriptor Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Curren_App_TxBuffer_H

Address: Operational Base + offset (0x1150)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RO	0x00	CURTBUFAPTRH Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Current_App_TxBuffer

Address: Operational Base + offset (0x1154)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CURTBUFAPTR Application Transmit Buffer Address Pointer The DMA updates this pointer during Tx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Current_App_RxBuffer_H

Address: Operational Base + offset (0x1158)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	CURRBUFAPTRH Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Current_App_RxBuffer

Address: Operational Base + offset (0x115C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CURRBUFAPTR Application Receive Buffer Address Pointer The DMA updates this pointer during Rx operation. This pointer is cleared on reset.

GMAC_DMA_CH0_Status

Address: Operational Base + offset (0x1160)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21:19	RO	0x0	REB Rx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. 1.Bit 21 1'b1: Error during data transfer by Rx DMA 1'b0: No Error during data transfer by Rx DMA 2.Bit 20 1'b1: Error during descriptor access 1'b0: Error during data buffer access 3.Bit 19 1'b1: Error during read transfer 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.
18:16	RO	0x0	TEB Tx DMA Error Bits This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface. 1.Bit 18 1'b1: Error during data transfer by Tx DMA 1'b0: No Error during data transfer by Tx DMA 2.Bit 17 1'b1: Error during descriptor access 1'b0: Error during data buffer access 3.Bit 16 1'b1: Error during read transfer 1'b0: Error during write transfer This field is valid only when the FBE bit is set. This field does not generate an interrupt.

Bit	Attr	Reset Value	Description
15	RW	0x0	NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_Interrupt_Enable register: Bit 0: Transmit Interrupt Bit 2: Transmit Buffer Unavailable Bit 6: Receive Interrupt Bit 11: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH0_Interrupt_Enable register) affect the Normal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Normal Interrupt Summary status not detected 0x1 (ACTIVE): Normal Interrupt Summary status detected
14	RW	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 the DMA_CH0_Interrupt_Enable register: Bit 1: Transmit Process Stopped Bit 7: Receive Buffer Unavailable Bit 8: Receive Process Stopped Bit 10: Early Transmit Interrupt Bit 12: Fatal Bus Error Bit 13: Context Descriptor Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Abnormal Interrupt Summary status not detected 0x1 (ACTIVE): Abnormal Interrupt Summary status
13	RW	0x0	CDE Context Descriptor Error This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Context Descriptor Error status not detected 0x1 (ACTIVE): Context Descriptor Error status detected

Bit	Attr	Reset Value	Description
12	RW	0x0	FBE Fatal Bus Error This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Fatal Bus Error status not detected 0x1 (ACTIVE): Fatal Bus Error status detected
11	RW	0x0	ERI Early Receive Interrupt This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory. In configs supporting ERIC, When ERIC=0, this bit is set only after the Rx DMA has filled up a complete receive buffer with packet data. When ERIC=1, this bit is set after every burst transfer of data from the Rx DMA to the buffer.The setting of RI bit automatically clears this bit. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Early Receive Interrupt status not detected 0x1 (ACTIVE): Early Receive Interrupt status detected
10	RW	0x0	ETI Early Transmit Interrupt This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory. In configs supporting ERIC: When ETIC=0, this bit is set only after the Tx DMA has transferred a complete packet to MTL. When ETIC=1, this bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC=1. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Early Transmit Interrupt status not detected 0x1 (ACTIVE): Early Transmit Interrupt status detected
9	RO	0x0	reserved
8	RW	0x0	RPS Receive Process Stopped This bit is asserted when the Rx process enters the Stopped state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Receive Process Stopped status not detected 0x1 (ACTIVE): Receive Process Stopped status detected

Bit	Attr	Reset Value	Description
7	RW	0x0	RBU Receive Buffer Unavailable This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Receive Buffer Unavailable status not detected 0x1 (ACTIVE): Receive Buffer Unavailable status detected
6	RW	0x0	RI Receive Interrupt This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. The reception remains in the Running state. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Receive Interrupt status not detected 0x1 (ACTIVE): Receive Interrupt status detected
5:3	RO	0x0	reserved
2	RW	0x0	TBU Transmit Buffer Unavailable This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_Debug_Status0 register explains the Transmit Process state transitions. To resume processing the Transmit descriptors, the application should do the following: 1. Change the ownership of the descriptor by setting Bit 31 of TDES3. 2. Issue a Transmit Poll Demand command. For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Transmit Buffer Unavailable status not detected 0x1 (ACTIVE): Transmit Buffer Unavailable status

Bit	Attr	Reset Value	Description
1	RW	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Transmit Process Stopped status not detected 0x1 (ACTIVE): Transmit Process Stopped status detected
0	RW	0x0	TI Transmit Interrupt This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor. Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect. 0x0 (INACTIVE): Transmit Interrupt status not detected 0x1 (ACTIVE): Transmit Interrupt status detected

GMAC_DMA_CH0_Miss_Frame_Cnt

Address: Operational Base + offset (0x1164)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RO	0x0	MFC0 Overflow status of the MFC Counter When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event. 0x0 (INACTIVE): Miss Frame Counter overflow not occurred 0x1 (ACTIVE): Miss Frame Counter overflow occurred
14:11	RO	0x0	reserved
10:0	RO	0x000	MFC Dropped Packet Counters This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH0_Rx_Control register. The counter gets cleared when this register is read. Access restriction applies. Clears on read. Self-set to 1 on internal event.

GMAC_DMA_CH0_RX_ERI_Cnt

Address: Operational Base + offset (0x1168)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	ECNT ERI Counter When ERIC bit of DMA_CH0_RX_Control register is set, this counter increments for burst transfer completed by the Rx DMA from the start of packet transfer. This counter will get reset at the start of new packet.

19.5 Application Note

19.5.1 Descriptors

The DMA engine uses descriptors to efficiently move data from source to destination with minimal application CPU intervention. The DMA is designed for packet-oriented data transfers such as packets in Ethernet. The DMA controller can be programmed to interrupt the application CPU for situations such as Packet Transmit and Receive Transfer completion, and other normal or error conditions.

The DMA and the application communicate through the following two data structures:

- Control and Status registers (CSR)
- Descriptor lists and data buffers

The base address of each list is written to the respective Tx Descriptor List Address register and Rx Descriptor List Address register. The descriptor list is forward linked and the next descriptor is always considered at a fixed offset to the current one. The offset is controlled by the DSL field of DMA_Ch[n]_Control register. The number of descriptors in the list is programmed in the respective Tx (or Rx) Descriptor Ring Length register. Once the DMA processes the last descriptor in the list, it automatically jumps back to the descriptor in the List Address register to create a descriptor ring.

The descriptor lists reside in the physical memory address space of the application. Each descriptor can point to a maximum of two buffers in the system memory. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the application physical memory space and consists of an entire packet or part of a packet but cannot exceed a single packet. Buffers contain only data. Buffer status is maintained in the descriptor. Data chaining refers to packets that span multiple data buffers. However, a single descriptor cannot span multiple packets. The DMA skips to the data buffer of next packet when EOP is detected.

The GMAC supports the following two types of descriptors:

- Normal Descriptor: Normal descriptors are used for packet data and to provide control information applicable to the packets to be transmitted
- Context Descriptor: Context descriptors are used to provide control information applicable to the packet to be transmitted

The GMAC supports the ring structure for the DMA descriptor.

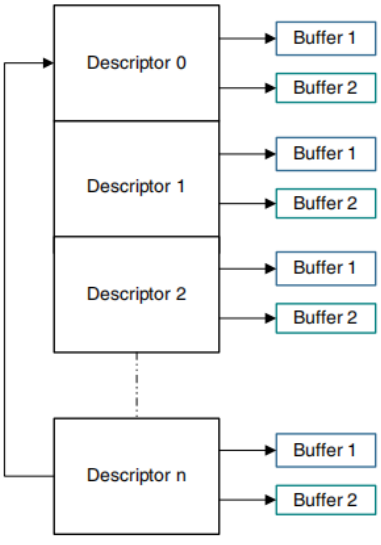


Fig. 19-10 Descriptor Ring Structure

In Ring structure, descriptors are separated by the Word, DWord, or LWord number programmed in the DSL field of the DMA_CH#_Control register. The application needs to program the total ring length, that is, the total number of descriptors in ring span in the following registers of a DMA channel:

- Transmit Descriptor Ring Length Register (DMA_CH#_TxDesc_Ring_Length)
- Receive Descriptor Ring Length Register (DMA_CH#_RxDesc_Ring_Length)

The Descriptor Tail Pointer Register contains the pointer to the descriptor address (N). The base address and the current descriptor pointer decide the address of the current descriptor that the DMA can process.
The descriptors up to one location less than the one indicated by the descriptor tail pointer (N – 1) are owned by the DMA. The DMA continues to process the descriptors until the following condition occurs:

Current Descriptor Pointer == Descriptor Tail Pointer

The DMA goes into the Suspend mode when this condition occurs. The application must perform a write to the Descriptor Tail pointer register and update the tail pointer so that the following condition is true:

Current Descriptor Pointer < Descriptor Tail Pointer

The DMA automatically wraps around the base address when the end of ring is reached. For descriptors owned by the application, the OWN bit of DES3 is reset to 0. For descriptors owned by the DMA, the OWN bit is set to 1. If the application has only one descriptor in the beginning, the application sets the last descriptor address (tail pointer) to Descriptor Base Address + 1. The DMA processes the first descriptor and then waits for the application to advance the tail pointer.

19.5.2 Transmit Descriptors

The DMA in GMAC requires at least one descriptor for a transmit packet. In addition to two buffers, two byte-count buffers, and two address pointers, the transmit descriptor has control fields which can be used to control the MAC operation on per-transmit packet basis. The Transmit Normal descriptor has two formats: Read format and Write-Back format.

19.5.3 Transmit Normal Descriptor (Read Format)

Table 19-3 TDES0 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF1AP	Buffer 1 Address Pointer or TSO Header Address Pointer These bits indicate the physical address of Buffer 1. These bits indicate the TSO Header Address pointer when the following bits are set: ■ TSE bit of TDES3 ■ FD bit of TDES3

Table 19-4 TDES1 Normal Descriptor (Read Format)

Bit	Name	Description
31:0	BUF2AP	Buffer 2 or Buffer 1 Address Pointer This bit indicates the physical address of Buffer 2 when a descriptor ring structure is used. There is no limitation for the buffer address alignment. In 40- or 48-bit addressing mode, these bits indicate the most-significant 8- or 16- bits of the Buffer 1 Address Pointer.

Table 19-5 TDES2 Normal Descriptor (Read Format)

Bit	Name	Description
31	IOC	Interrupt on Completion This bit controls the setting of TI and ETI status bits in the DMA_CH#_Status register. When ETIC = 1 and TDES2[LD] = 0, this bit sets the ETI bit. When TDES3[LD] = 1, this bit sets the TI status bit.
30	TTSE/T MWD	Transmit Timestamp Enable or External TSO Memory Write Enable This bit enables the IEEE1588 time stamping for Transmit packet referenced by the descriptor, if TSE bit is not set. If TSE bit is set and external TSO memory is enabled, setting this bit disables external TSO memory writing for

Bit	Name	Description
		this packet.
29:16	B2L	Buffer 2 Length The driver sets this field. When set, this field indicates Buffer 2 length
15:14	VTIR	VLAN Tag Insertion or Replacement These bits request the MAC to perform VLAN tagging or untagging before transmitting the packets. The application must set the CRC Pad Control bits appropriately when VLAN Tag Insertion, Replacement, or Deletion is enabled for the packet. The following list describes the values of these bits: ■ 2'b00: Do not add a VLAN tag. ■ 2'b01: Remove the VLAN tag from the packets before transmission. This option should be used only with the VLAN packets. ■ 2'b10: Insert a VLAN tag with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. ■ 2'b11: Replace the VLAN tag in packets with the tag value programmed in the MAC_VLAN_Incl register or context descriptor. This option should be used only with the VLAN packets. These bits are valid when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core
13:0	HL or B1L	Header Length or Buffer 1 Length For Header length only bits [9:0] are taken. The size 13:0 is applicable only when interpreting buffer 1 length. If the TCP Segmentation Offload feature is enabled through the TSE bit of TDES3, this field is equal to the header length. When the TSE bit is set in TDES3, the header length includes the length in bytes from Ethernet Source address till the end of the TCP header. The maximum header length supported for TSO feature is 1023 bytes. The maximum header length supported for TSO feature is 1023 bytes. If the TCP Segmentation Offload feature is not enabled, this field is equal to Buffer 1 length.

Table 19-6 TDES3 Normal Descriptor (Read Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:26	CPC	CRC Pad Control

Bit	Name	Description
		This field controls the CRC and Pad Insertion for Tx packet. This field is valid only when the first descriptor bit (TDES3[29]) is set. The following list describes the values of Bits[27:26]: ■ 2'b00: CRC and Pad Insertion. The MAC appends the cyclic redundancy check (CRC) at the end of the transmitted packet of length greater than or equal to 60 bytes. The MAC automatically appends padding and CRC to a packet with length less than 60 bytes. ■ 2'b01: CRC Insertion (Disable Pad Insertion). The MAC appends the CRC at the end of the transmitted packet but it does not append padding. The application should ensure that the padding bytes are present in the packet being transferred from the Transmit Buffer, that is, the packet being transferred from the Transmit Buffer is of length greater than or equal to 60 bytes. ■ 2'b10: Disable CRC Insertion. The MAC does not append the CRC at the end of the transmitted packet. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. ■ 2'b11: CRC Replacement. The MAC replaces the last four bytes of the transmitted packet with recalculated CRC bytes. The application should ensure that the padding and CRC bytes are present in the packet being transferred from the Transmit Buffer. This field is valid only for the first descriptor. Note: When the TSE bit is set, the MAC ignores this field because the CRC and pad insertion is always done for segmentation.
25:23	SAIC	SA Insertion Control These bits request the MAC to add or replace the Source Address field in the Ethernet packet with the value given in the MAC Address 0 register. The application must set the CRC Pad Control bits appropriately when SA Insertion Control is enabled for the packet. Bit 25 specifies the MAC Address Register (1 or 0) value that is used for Source Address insertion or replacement. The following list describes the values of Bits[24:23]: ■ 2'b00: Do not include the source address ■ 2'b01: Include or insert the source address. For reliable transmission, the application must provide frames without source addresses. ■ 2'b10: Replace the source address. For reliable transmission, the application must provide frames with source addresses. ■ 2'b11: Reserved These bits are valid in the EQOS-DMA, EQOS-AXI, and EQOS-AHB configurations when the Enable SA and VLAN Insertion on Tx option is selected while configuring the core and when the First Segment control bit (TDES3 [29]) is set. This field is valid only for the first descriptor.
22:19	SLOTNUM or THL	SLOTNUM: Slot Number Control Bits in AV Mode These bits indicate the slot interval in which the data should be fetched from the corresponding buffers

Bit	Name	Description
		addressed by TDES0 or TDES1. When the Transmit descriptor is fetched, the DMA compares the slot number value in this field with the slot interval maintained in the RSN field DMA_CH#_Slot_Function_Control_Status. It fetches the data from the buffers only if a value matches. These bits are valid only for the AV channels. THL: TCP/UDP Header Length If the TSE bit is set, this field contains the length of the TCP/UDP header. The minimum value of this field must be 5 for TCP header. The value must be equal to 2 for UDP header. This field is valid only for the first descriptor.
18	TSE	TCP Segmentation Enable When this bit is set, the DMA performs the TCP/UDP segmentation or UDP fragmentation for a packet depending on the TSE_MODE[1:0] bit of the DMA_CH(#i)_Tx_Control Register. This bit is valid only if the FD bit is set.
17:16	CIC/TPL	Checksum Insertion Control or TCP Payload Length These bits control the checksum calculation and insertion. The following list describes the bit encoding: ■ 2'b00: Checksum Insertion Disabled. ■ 2'b01: Only IP header checksum calculation and insertion are enabled. ■ 2'b10: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo-header checksum is not calculated in hardware. ■ 2'b11: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo-header checksum is calculated in hardware. This field is valid when the Enable Transmit TCP/IP Checksum Offload option is selected and the TSE bit is reset. When the TSE bit is set, this field contains the upper bits [17:16] of the TCP Payload (or IP Payload for UDP fragmentation). This allows the TCP/UDP packet length field to be spanned across TDES3[17:0] to provide 256 KB packet length support. This field is valid only for the first descriptor.
15	TPL	Reserved or TCP Payload Length When the TSE bit is reset, this bit is reserved. When the TSE bit is set, this is Bit 15 of the TCP payload length [17:0]. This field is valid only when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected while configuring the core.
14:0	FL/TPL	Frame Length or TCP Payload Length This field is equal to the length of the packet to be transmitted in bytes. When the TSE bit is not set, this field is equal to the total length of the packet to be transmitted: Ethernet Header Length + TCP /IP Header Length – Preamble Length – SFD Length + Ethernet Payload Length When the TSE bit is set, this field is equal to the lower 15 bits of the TCP payload length in case of segmentation and IP payload in case of UDP fragmentation. In case of segmentation, this length does not include Ethernet

Bit	Name	Description
		header or TCP/UDP/IP header length. In case of fragmentation, this length does not include Ethernet header and IP header. When DWRR/WFQ algorithm is NOT enabled, value written into this field is not used when TSE = 0.

19.5.4 Transmit Normal Descriptor (Write-back Format)

The write-back format of the Transmit Descriptor includes timestamp low, timestamp high, OWN, and Status bits. The write-back format is applicable only for the last descriptor of the corresponding packet. The LD bit (TDES3[28]) is set in the descriptor where the DMA writes back the status and timestamp information for the corresponding Transmit packet. this format is only applicable to the last descriptor of a packet.

Table 19-7 TDES0 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for the corresponding Transmit packet. The DMA writes the timestamp only if TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and the Timestamp status (TTSS) bit is set

Table 19-8 TDES1 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High The DMA updates this field with the most significant 32 bits of the timestamp captured for corresponding transmit packet. The DMA writes the timestamp only if the TTSE bit of TDES2 is set in the first descriptor of the packet. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 19-9 TDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31:0	Reserved	Reserved

Table 19-10 TDES2 Normal Descriptor (Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit after it completes the transfer of data given in the associated buffer(s).
30	CTXT	Context Type This bit should be set to 1'b0 for normal descriptor.
29	FD	First Descriptor When this bit is set, it indicates that the buffer contains the first segment of a packet.
28	LD	Last Descriptor When this bit is set, it indicates that the buffer contains the last segment of the packet. When this bit is set, the B1L or B2L field should have a non-zero value.
27:24	Reserved	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect. The DMA sets this bit during write-

Bit	Name	Description
		back while closing the descriptor. Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location after the first descriptor for a packet ■ All 1s ■ CTXT, LD, and FD bits set to 1 Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status register. Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:18	Reserved	Reserved
17	TTSS	Tx Timestamp Status This status bit indicates that a timestamp has been captured for the corresponding transmit packet. When this bit is set, TDES0 and TDES1 have timestamp values that were captured for the Transmit packet. This field is valid only when the Last Segment control bit (TDES3 [28]) in a descriptor is set. This bit is valid only when IEEE1588 timestamping feature is enabled; otherwise, it is reserved.
16	EUE	ECC Uncorrectable Error Status Indicates the ECC uncorrectable error in the TSO memory. Note: Uncorrectable error in Transmit FIFO memory is reported with (Bit 13) FF = 1. This is because, all such packets are flushed by GMAC.
15	ES	Error Summary This bit indicates the logical OR of the following bits: ■ TDES3[0]: IP Header Error ■ TDES3[14]: Jabber Timeout ■ TDES3[13]: Packet Flush ■ TDES3[12]: Payload Checksum Error ■ TDES3[11]: Loss of Carrier ■ TDES3[10]: No Carrier ■ TDES3[9]: Late Collision ■ TDES3[8]: Excessive Collision ■ TDES3[3]: Excessive Deferral ■ TDES3[2]: Underflow Error This bit is also set when EUE (bit 16) is set
14	JT	Jabber Timeout This bit indicates that the MAC transmitter has experienced a jabber time-out. This bit is set only when the JD bit of the MAC_Configuration register is not set.
13	PF	Packet Flushed This bit indicates that the DMA or MTL flushed the packet because of a software flush command given by the CPU.
12	PCE	Payload Checksum Error This bit indicates that the Checksum Offload engine had a failure and did not insert any checksum into the

Bit	Name	Description
		encapsulated TCP, UDP, or ICMP payload. This failure can be either because of insufficient bytes, as indicated by the Payload Length field of the IP Header or the MTL starting to forward the packet 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 packet being transmitted to avoid deadlock, the MTL starts forwarding the packet when the FIFO is full, even in the store-and-forward mode. This error can also occur when Bus Error is detected during packet transfer. When the Full Checksum Offload engine is not enabled, this bit is reserved.
11	LoC	Loss of Carrier This bit indicates that Loss of Carrier occurred during packet transmission (that is, the gmii_crs_i signal was inactive for one or more transmit clock periods during packet transmission). This is valid only for the packets transmitted without collision and when the MAC operates in the half-duplex mode.
10	NC	No Carrier This bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC	Late Collision This bit indicates that packet transmission was aborted because a collision occurred after the collision window (64 byte times including Preamble in MII mode and 512 byte times including Preamble and Carrier Extension in GMII mode). This bit is not valid if Underflow Error is set.
8	EC	Excessive Collision This bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after first collision and the transmission of the packet is aborted.
7:4	CC	Collision Count This 4-bit counter value indicates the number of collisions occurred before the packet was transmitted. The count is not valid when the EC bit is set.
3	ED	Excessive Deferral This bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000 Mbps mode or Jumbo Packet enabled mode) if DC bit is set in the MAC_Configuration register. When TBS is enabled in full duplex mode and this bit is set, it indicates that the frame has been dropped after the expiry time has reached.
2	UF	Underflow Error This bit indicates that the MAC aborted the packet because the data arrived late from the system memory. The underflow error can occur because of either of the following conditions: ■ The DMA encountered an empty Transmit Buffer while transmitting the packet ■ The application filled the MTL Tx FIFO slower than the MAC transmit rate

Bit	Name	Description
		The transmission process enter the suspended state and sets the underflow bit corresponding to a queue in the MTL_Interrupt_Status register.
1	DB	Deferred Bit This bit indicates that the MAC deferred before transmitting because of presence of carrier. This bit is valid only in the half-duplex mode.
0	IHE	IP Header Error When IP Header Error is set, this bit indicates that the Checksum Offload engine detected an IP header error. This bit is valid only when Tx Checksum Offload is enabled. Otherwise, it is reserved. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload. In full duplex mode, when EST/Qbv is enabled and this bit is set, it indicates the frame drop status due to Frame Size error or Schedule Error.

19.5.5 Programming Guide

19.5.5.1 Initializing DMA

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the MAC internal registers and logic (bit-0 of DMA_Mode).
2. Wait for the completion of the reset process (poll bit 0 of the DMA_Mode, which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the DMA_SysBus_Mode register:

a. AAL

b. Fixed burst or undefined burst

c. Burst mode values in case of AHB bus interface, OSR_LMT in case of AXI bus interface.

d. If fixed length value is enabled, select the maximum burst length possible on the AXI Bus (bits [7:1])

4. Create a descriptor list for transmit and receive. In addition, ensure that the descriptors are owned by DMA (set bit 31 of descriptor TDES3/RDES3). For more information about descriptors, see section "Descriptors".
5. Program the Transmit and Receive Ring length registers (DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Ring_Length (for i = 0; i <= GMAC_NUM_DMA_RX_CH-1)). The ring length programmed must be at least 4.
6. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1), DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= GMAC_NUM_DMA_RX_CH-1)). Also, program transmit and receive tail pointer registers indicating to the DMA about the available descriptors (DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1) and DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= GMAC_NUM_DMA_RX_CH-1)).
7. Program the settings of the following registers for the parameters like maximum burst-length (PBL) initiated by DMA, descriptor skip lengths, OSP in case of TxDMA, RBSZ in case of RxDMA, and so on:
   - DMA_CH(#i)_Control (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_TX_Control (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1)
   - DMA_CH(#i)_RX_Control (for i = 0; i <= GMAC_NUM_DMA_RX_CH-1)
8. Enable the interrupts by programming the DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1) register.
9. Start the Receive and Transmit DMAs by setting SR (bit 0) of the DMA_CH(#i)_RX_Control (for i = 0; i <= GMAC_NUM_DMA_RX_CH-1) and ST (bit 0) of the DMA_CH(#i)_TX_Control (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1) register.

10. Repeat steps 4 to 9 for all the Tx DMA and Rx DMA channels selected in the hardware.

19.5.5.2 Initializing MTL Registers

The Transaction Layer (MTL) registers must be initialized to establish the transmit and receive operating modes and commands.
Complete the following steps to initialize the MTL registers:
1. Program the Tx Scheduling (SCHALG) and Receive Arbitration Algorithm (RAA) fields in MTL_Operation_Mode to initialize the MTL operation in case of multiple Tx and Rx queues.
2. Program the Receive Queue to DMA mapping in MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1 registers.
3. Program the following fields to initialize the mode of operation in the MTL_TxQ0_Operation_Mode register.
a. Transmit Store And Forward (TSF) or Transmit Threshold Control (TTC) in case of threshold mode
b. Transmit Queue Enable (TXQEN) to value 2'b10 to enable Transmit Queue0
c. Transmit Queue Size (TQS)
4. Program the following fields to initialize the mode of operation in the MTL_RxQ0_Operation_Mode register:
a. Receive Store and Forward (RSF) or RTC in case of Threshold mode
b. Flow Control Activation and De-activation thresholds for MTL Receive FIFO (RFA and RFD)
c. Error Packet and undersized good Packet forwarding enable (FEP and FUP)
d. Receive Queue Size (RQS)
5. Repeat previous two steps for all MTL Tx and Rx queues selected in the configuration.

19.5.5.3 Initializing MAC

The MAC configuration registers establish the operating mode of the MAC. These registers must be initialized before initializing the DMA.
The following MAC Initialization operations can be performed after DMA initialization. If the MAC initialization is completed before the DMA is configured, enable the MAC receiver (last step in the following sequence) only after the DMA is active. Otherwise, received frames fill the Rx FIFO and overflow.
1. Provide the MAC address registers: MAC_Address0_High and MAC_Address0_Low. If more than one MAC address is enabled in your configuration, program the MAC addresses appropriately.
2. Program the following fields to set the appropriate filters for the incoming frames in the MAC_Packet_Filter register:
a. Receive All
b. Promiscuous mode
c. Hash or Perfect Filter
d. Unicast, multicast, broadcast, and control frames filter settings
3. Program the following fields for proper flow control in the MAC_Q0_Tx_Flow_Ctrl register:
a. Pause time and other Pause frame control bits
b. Transmit Flow control bits
c. Flow Control Busy
4. Program the MAC_Interrupt_Enable register, as required, and if applicable, for your configuration.
5. Program the appropriate fields in the MAC_Configuration register. For ex: Inter-packet gap while transmission and jabber disable.
6. Set bit 0 and 1 in MAC_Configuration registers to start the MAC transmitter and receiver.

19.5.5.4 Performing Normal Receive and Transmit Operation

During normal operation of the GMAC, normal and transmit interrupts are read, descriptors polled, the DMA is suspended (if it does not own descriptors), and values of current host transmitter or receiver descriptor pointers are read for debugging.
For normal operation, complete the following steps:

1. 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).
2. 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.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and writing the descriptor tail pointer to Tx/Rx tail pointer register (DMA_CH[n]_TxDesc_Tail_Pointer and DMA_CH[n]_RxDesc_Tail_Pointer).
4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (DMA_CH[n]_Current_App_TxDesc and DMA_CH[n]_Current_App_RxDesc).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Register DMA_CH[n]_Current_App_TxBuffer and DMA_CH[n]_Current_App_RxBuffer)

19.5.5.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time.
1. Disable the Transmit DMA (if applicable) by clearing Bit 0 (ST) of DMA_CH(#i)_TX_Control (for i = 0; i <= GMAC_NUM_DMA_TX_CH-1) Register.
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of MTL_TxQ0_Debug Register (TRCSTS is not 01 and TXQSTS=0).
3. Disable the MAC transmitter and MAC receiver by clearing Bit (RE) and Bit 1(TE) of the MAC_Configuration Register.
4. Disable the Receive DMA (if applicable), after making sure that the data in the Rx FIFO is transferred to the system memory (by reading the appropriate bits of MTL_TxQ0_Debug Register, PRXQ=0 and RXQSTS=00).
5. Make sure that both Tx Queue and Rx Queue are empty (TXQSTS is 0 in MTL_TxQ0_Debug Register and RXQSTS is 0 in MTL_RxQ0_Debug Register).
6. To restart the operation, first start the DMAs, and then enable the MAC Transmitter and Receiver.

19.5.5.6 Initialization Guidelines for System Time Generation

You can enable the timestamp feature by setting Bit 0 of the MAC_Timestamp_Control Register. However, it is essential that the timestamp counter be initialized after this bit is set. Complete the following steps during GMAC initialization:
1. Mask the Timestamp Trigger interrupt by clearing the bit 16 of MAC_Interrupt_Enable Register.
2. Set Bit 0 of MAC_Timestamp_Control Register to enable timestamping.
3. Program MAC_Sub_Second_Increment Register based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program MAC_Timestamp_Addend and set Bit 5 of MAC_Timestamp_Control Register.
5. Poll the MAC_Timestamp_Control Register until Bit 5 is cleared.
6. Program Bit 1 of MAC_Timestamp_Control Register to select the Fine Update method (if required).
7. Program MAC_System_Time_Seconds_Update Register and MAC_System_Time_Nanoseconds_Update Register with the appropriate time value.
8. Set Bit 2 in MAC_Timestamp_Control Register.
The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers. If one-step timestamping is enabled
a. To enable one-step timestamping, program Bit 27 of the TDES3 Context Descriptor.
b. Program registers MAC_Timestamp_Ingress_Asym_Corr and MAC_Timestamp_Egress_Asym_Corr to update the correction field in PDelay_Req PTP messages.
9. Enable the MAC receiver and transmitter for proper timestamping.

19.5.5.7 Coarse Correction Method

To synchronize or update the system time in one process (coarse correction method), complete the following steps:
1. Set the offset (positive or negative) in the Timestamp Update registers (MAC_System_Time_Seconds_Update and MAC_System_Time_Nanoseconds_Update).
2. Set Bit 3 (TSUPDT) of MAC_Timestamp_Control Register. The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

19.5.5.8 Programming Guidelines for TSO

The TCP Segmentation Offload (TSO) engine is used to offload the TCP segmentation functions to the hardware. To program the TSO, set the TSE bit to enable TCP packet segmentation, and program descriptor fields to enable TSO for the current packet. Complete the following steps to program TSO:
1. Program the TSE bit of corresponding DMA_CH[n]_Tx_Control register to enable TCP packet segmentation in that DMA.
2. In addition to the normal transfer descriptor setting, the following descriptor fields must be programmed to enable TSO for the current packet:
a. Enable TSE in Bit 18 of TDES3
b. Program the length of the un-segmented TCP/IP packet payload in bits [17:0] of TDES3 and the TCP header in bits [22:19] of TDES3.
c. Program the maximum size of the segment in MSS of DMA_CH[n]_Control register or MSS in the context descriptor. If MSS field is programmed in both DMA_CH[n]_Control register and in the context descriptor, the latest software programmed sequence is considered.
3. The header of the unsegmented TCP/IP packet should be in Buffer 1 of the first descriptor and this buffer must not hold any payload bytes. The payload is allocated to Buffer 2 and the buffers of the subsequent descriptors.

19.5.5.9 Programming Guidelines for Multi-Channel Multi-Queuing

Transmit

1. Program the Transmit queue size in the TQS field of MTL_TxQ[n]_Operation_Mode register. Based on the value programmed in the TQS field, the size of the queue is determined.
In the Transmit operation, the number of channels is equal to the number of the queues. Due to this reason, the Channel-to-Queue mapping is fixed.
2. For a queue to be used, the queue needs to be enabled in TXQEN in the corresponding MTL_TxQ[n]_Operation_Mode Register. In DMA configurations, the ST bit of DMA_CH[n]_Tx_Control Register and corresponding TXQEN in MTL_TxQ[n]_Operation Mode Register needs to be enabled.
3. The scheduling method needs to be programmed in SCHALG of MTL_Operation_Mode register.
4. Program the MTL_TxQ[n]_Quantum_Weight for DCB queue as per the selected algorithm. In case of CBS algorithm in AVB queues, the MTL_TxQ[n]_ETS_Control, MTL_TxQ[n]_SendSlopeCredit, MTL_TxQ[n]_HiCredit and MTL_TxQ[n]_LoCredit registers also need to be programmed as required.
5. If DCB is enabled and PFC function is required, program MAC_TxQ_Prty_Map0 Register to assign a fixed priority to the queue. This priority assigned is used for determining if the corresponding queue should stop transmitting packet based on the received PFC packet.

Receive

1. Program the Receive queue size in the RQS field of MTL_RxQ[n]_Operation_Mode Register. Based on the value programmed in RQS field, the size of the queue is determined.
2. Enable the Receive Queues 0 to 7 in the fields RXQ0EN to RXQ7EN in MAC_RxQ_Ctrl0 Register for AV or DCB. In DMA configurations, SR bit of statically or dynamically mapped DMA_CH[n]_Rx_Control Register and corresponding RXQ[n]_EN in MAC_RxQ_Ctrl0 Register needs to be enabled.

3. The MAC routes the Rx packets to the Rx Queues based on following packet types:
   a. AV PTP Packets: Based on the programming of AVPTPQ in MAC_RxQ_Ctrl1 Register.
   b. AV Untagged Control packets: Based on the programming of AVCPQ in MAC_RxQ_Ctrl1 Register.
   c. Data Center Bridging (DCB) related Link Layer Discovery Protocol (LLDP) packets. Program
   DCBCPQ in MAC_RxQ_Ctrl1 Register to indicate to MAC which queue should get the DCB packets.
   d. VLAN Tag Priority field in VLAN Tagged packets: Program PSRQ7-0 of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 Register for the routing of tagged packets based on the USP (user Priority) field of the received packets to the Rx Queues 0 to 7.
   e. The AV tagged control and data packets are also routed based on PSRQ field of the MAC_RxQ_Ctrl2 and MAC_RxQ_Ctrl3 registers.
4. If multiple RX DMA channels are enabled, the following programming should be done for proper arbitration and mapping:
   a. Program the RAA field of MTL_Operation_Mode register to select the arbitration algorithm to decide which RxQ is read out from the RxFIFO memory.
   b. Program the MTL_RxQ[n]_Control to decide the weights and the packet arbitration for each RxQ.
   c. If static mapping is programmed in MTL_RxQ_DMA_Map[n] register (RXQ[n]DADMACH is reset to 0), bits RXQx2DMA and others need to be programmed to select the channel for which each queue is mapped.
   d. Set RXQ[n]DADMACH bit in MTL_RxQ_DMA_Map0 Register to select dynamic mapping of
   packets in each RxQueue.
   e. In dynamic channel mapping, the routing of a packet to a specific RxDMA channel is decided by the value of DCS field in the lowest MAC Address Register.

19.5.6 Transmit Context Descriptor

The context descriptor is used to provide the timestamps for one-step timestamp correction, VLAN Tag ID for VLAN insertion feature. The Transmit Context descriptor can be provided any time before a packet descriptor. The context is valid for the current packet and subsequent packets. The context descriptor is used to provide the timestamps for one-step timestamp correction and VLAN Tag ID for VLAN insertion feature. Write back is done on a context descriptor only to reset the OWN bit.

Table 19-11 TDES0 Context Descriptor

Bit	Name	Description
31:0	TTSL	Transmit Packet Timestamp Low For one-step correction, the driver can provide the lower 32 bits of timestamp in this descriptor word. The DMA uses this value as the low word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 19-12 TDES1 Context Descriptor

Bit	Name	Description
31:0	TTSH	Transmit Packet Timestamp High For one-step correction, the driver can provide the upper 32 bits of timestamp in this descriptor. The DMA uses this value as the high word for doing one-step timestamp correction. This field is valid only if the OSTC and TCMSSV bits of TDES3 context descriptor are set.

Table 19-13 TDES2 Context Descriptor

Bit	Name	Description
31:16	IVT	Inner VLAN Tag When the IVLTV bit of TDES3 context descriptor is set and the TCMSSV and OSTC bits of TDES3 context descriptor are reset, TDES2[31:16] contains the inner VLAN Tag to be inserted in the subsequent Transmit packets.
15:14	Reserved	Reserved
13:0	MSS	Maximum Segment Size When the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected, the driver can provide maximum segment size in this field. This segment size is used while segmenting the TCP/IP payload. This field is valid only if the TCMSSV bit of TDES3 context descriptor is set and the OSTC bit of the TDES3 context descriptor is reset.

Table 19-14 TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit immediately after the read.
30	CTXT	Context Type This bit should be set to 1'b1 for Context descriptor.
29:28	Reserved	Reserved
27	OSTC	One-Step Timestamp Correction Enable When this bit is set, the DMA performs a one-step timestamp correction with reference to the timestamp values provided in TDES0 and TDES1.
26	TCMSSV	One-Step Timestamp Correction Input or MSS Valid When this bit and the OSTC bit are set, it indicates that the Timestamp Correction input provided in TDES0 and TDES1 is valid. When the OSTC bit is reset and this bit and the TSE bit of TDES3 are set in subsequent normal descriptor, it indicates that the MSS input in TDES2 is valid.

Bit	Name	Description
25:24	Reserved	Reserved
23	DE	Descriptor Error When this bit is set, it indicates that the descriptor content is incorrect.The DMA sets this bit during write-back while closing the context descriptor. Descriptor Errors can be: ■ Incorrect sequence from the context descriptor. For example, a location before the first descriptor for a packet ■ All 1s ■ CD, LD, and FD bits set to 1 Note 1: When Descriptor Error occurs due to All 1s or CTXT, LD, and FD bits set to 1, the Transmit DMA closes the transmit descriptor with DE and LD bits set to 1. When IOC bit in TDES2 of corresponding first descriptor is set to 1, Transmit DMA will set the TI bit in the DMA_CH#_Status Register. Note 2: Based on CTXT, LD, and FD bits of the transmit descriptor, the subsequent descriptor might be considered as the First Descriptor (even if FD bit is not set) and partial packet is sent.
22:20	Reserved	Reserved
19:18	IVTIR	Inner VLAN Tag Insert or Replace When this bit is set, these bits request the MAC to perform Inner VLAN tagging or un-tagging before transmitting the packets. If the packet is modified for VLAN tags, the MAC automatically recalculates and replaces the CRC bytes. The following list describes the values of these bits: ■ 2'b00: Do not add the inner VLAN tag. ■ 2'b01: Remove the inner VLAN tag from the packets before transmission. This option should be used only with the VLAN frames. ■ 2'b10: Insert an inner VLAN tag with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. ■ 2'b11: Replace the inner VLAN tag in packets with the tag value programmed in the MAC_Inner_VLAN_Incl register or context descriptor. This option should be used only with the VLAN frames. These bits are valid when the Enable SA and VLAN Insertion on Tx and Enable Double VLAN Processing options are selected.
17	IVLTV	Inner VLAN Tag Valid When this bit is set, it indicates that the IVT field of TDES2 is valid.
16	VLTV	VLAN Tag Valid When this bit is set, it indicates that the VT field of TDES3 is valid.
15:0	VT	VLAN Tag This field contains the VLAN Tag to be inserted or replaced in the packet. This field is used as VLAN Tag only when the VLTI bit of the MAC_VLAN_Incl register is reset.

19.5.7 Receive Descriptors

The DMA in GMAC attempts to read a descriptor only if the Tail Pointer is different from the Base Pointer or current pointer. It is recommended to have a descriptor ring with a length

that can accommodate at least two complete packets received by the MAC. Otherwise, the performance of the DMA is impacted greatly because of the unavailability of the descriptors. In such situations, the RxFIFO in MTL becomes full and starts dropping packets.

The following Receive Descriptors are present:

- Normal descriptors
- Context descriptors

All RX descriptors are prepared by the software and given to the DMA as "Normal" Descriptors with the content as shown in Receive Normal Descriptor (Read Format). The DMA reads this descriptor and after transferring a received packet (or part of) to the buffers indicated by the descriptor, the Rx DMA will close the descriptor with the corresponding packet status. The format of this status is given in the "Receive Normal Descriptor (Write-Back Format)". For some packets, the normal descriptor bits are not enough to write the complete status. For such packets, the RX DMA writes the extended status to the next descriptor (without processing or using the Buffers/ Pointers embedded in that descriptor). The format and content of the descriptor write back is described in "Receive Context Descriptor".

19.5.8 Receive Normal Descriptors(Read Format)

Table 19-15 TDES0 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	BUF1AP	Header or Buffer 1 Address Pointer When the SPH bit of Control register of a channel is reset, these bits indicate the physical address of Buffer 1. When the SPH bit is set, these bits indicate the physical address of Header Buffer where the Rx DMA writes the L2/L3/L4 header bytes of the received packet. The application can program a byte-aligned address for this buffer which means that the LS bits of this field can be non-zero. However, while transferring the start of packet, the DMA performs a Write operation with RDES0[1:0] (or RDES0[2:0]/[3:0] in case of 64-/128-bit configuration) as zero. However, the packet data is shifted as per actual offset as given by buffer address pointer. If the address pointer points to a buffer where the middle or last part of the packet is stored, the DMA ignores the offset address and writes to the full location as indicated by the data-width.

Table 19-16 TDES1 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

Table 19-17 TDES2 Normal Descriptor(Read Format)

Bit	Name	Description
31:0	Reserved or BUF1AP	In 64-bit addressing mode, this field contains the most-significant 32 bits of the Buffer 1 Address Pointer. Otherwise, this field is reserved.

Table 19-18 TDES3 Normal Descriptor(Read Format)

Bit	Name	Description
31	OWN	Own Bit

Bit	Name	Description
		When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	IOC	Interrupt Enabled on Completion When this bit is set, an interrupt is issued to the application when the DMA closes this descriptor.
29:26	Reserved	Reserved
25	BUF2V	Buffer 2 Address Valid When this bit is set, it indicates to the DMA that the buffer 2 address specified in RDES2 is valid. The application must set this bit so that the DMA can use the address, to which the Buffer 2 address in RDES2 is pointing, to write received packet data.
24	BUF1V	Buffer 1 Address Valid When set, this indicates to the DMA that the buffer 1 address specified in RDES1 is valid. The application must set this value if the address pointed to by Buffer 1 address in RDES1 can be used by the DMA to write received packet data.
23:0	Reserved	Reserved

19.5.9 Receive Normal Descriptors(Write-Back Format)

Table 19-19 TDES0 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:16	IVT	Inner VLAN Tag This field contains the Inner VLAN tag of the received packet if the RS0V bit of RDES3 is set. This is valid only when Double VLAN tag processing and VLAN tag stripping are enabled.
15:0	OVT	Outer VLAN Tag This field contains the Outer VLAN tag of the received packet if the RS0V bit of RDES3 is set.

Table 19-20 TDES1 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:16	OPC	OAM Sub-Type Code, or MAC Control Packet opcode OAM Sub-Type Code If Bits[18:16] of RDES3 are set to 3'b111, this field contains the OAM sub-type and code fields. MAC Control Packet opcode If Bits[18:16] of RDES3 are set to 3'b110, this field contains the MAC Control packet opcode field.
15	TD	Timestamp Dropped This bit indicates that the timestamp was captured for this packet but it got dropped in the MTL Rx FIFO because of overflow. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
14	TSA	Timestamp Available When Timestamp is present, this bit indicates that the timestamp value is available in a context descriptor word 2 (RDES2) and word 1(RDES1). This is valid only when the Last Descriptor bit (RDES3 [28]) is set.

Bit	Name	Description
		The context descriptor is written in the next descriptor just after the last normal descriptor for a packet.
13	PV	PTP Version This bit indicates that the received PTP message has the IEEE 1588 version 2 format. When this bit is reset, it indicates the IEEE 1588 version 1 format. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
12	PFT	PTP Packet Type This bit indicates that the PTP message is sent directly over Ethernet. This bit is available only when you select the Timestamp feature. Otherwise, this bit is reserved.
11:8	PMT	PTP Message Type These bits are encoded to give the type of the message received: ■ 0000: No PTP message received ■ 0001: SYNC (all clock types) ■ 0010: Follow_Up (all clock types) ■ 0011: Delay_Req (all clock types) ■ 0100: Delay_Resp (all clock types) ■ 0101: Pdelay_Req (in peer-to-peer transparent clock) ■ 0110: Pdelay_Resp (in peer-to-peer transparent clock) ■ 0111: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) ■ 1000: Announce ■ 1001: Management ■ 1010: Signaling ■ 1011–1110: Reserved ■ 1111: PTP packet with Reserved message type These bits are available only when you select the Timestamp feature.
7	IPCE	IP Payload Error When this bit is set, it indicates either of the following: ■ The 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) calculated by the MAC does not match the corresponding checksum field in the received segment. ■ The TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. ■ The TCP, UDP, or ICMP segment length is less than minimum allowed segment length for TCP, UDP, or ICMP. Bit 15 (ES) of RDES3 is not set when this bit is set.
6	IPCB	IP Checksum Bypassed This bit indicates that the checksum offload engine is bypassed. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
5	IPV6	IPv6 header Present This bit indicates that an IPV6 header is detected. When the Enable Split Header Feature option is selected and the SPH bit of Control Register of a channel is set, the IPV6 header is available in the header buffer area to which RDES0 is pointing.
4	IPV4	IPV4 Header Present

Bit	Name	Description
		This bit indicates that an IPV4 header is detected. When the SPH bit of RDES3 is set, the IPV4 header is available in the header buffer area to which RDES0 is pointing.
3	IPHE	IP Header Error When this bit is set, it indicates either of the following: ■ The 16-bit IPv4 header checksum calculated by the MAC does not match the received checksum bytes. ■ The IP datagram version is not consistent with the Ethernet Type value. ■ Ethernet packet does not have the expected number of IP header bytes. This bit is valid when either Bit 5 or Bit 4 is set. This bit is available when you select the Enable Receive TCP/IP Checksum Check feature.
2:0	PT	Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE): ■ 3'b000: Unknown type or IP/AV payload not processed ■ 3'b001: UDP ■ 3'b010: TCP ■ 3'b011: ICMP ■ 3'b110: AV Tagged Data Packet ■ 3'b111: AV Tagged Control Packet ■ 3'b101: AV Untagged Control Packet ■ 3'b100: IGMP if IPV4 Header Present bit is set else DCB (LLDP) Control Packet If the COE does not process the payload of an IP datagram because there is an IP header error or fragmented IP, it sets these bits to 3'b000.

Table 19-21 TDES2 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31:29	L3L4FM	Layer 3 and Layer 4 Filter Number Matched These bits indicate the number of the Layer 3 and Layer 4 Filter that matched the received packet: ■ 000: Filter 0 ■ 001: Filter 1 ■ 010: Filter 2 ■ 011: Filter 3 ■ 100: Filter 4 ■ 101: Filter 5 ■ 110: Filter 6 ■ 111: Filter 7 This field is valid only when Bit 28 or Bit 27 is set high. When more than one filter matches, these bits give the number of lowest filter. Note: This status is not available when Flexible RX Parser is enabled.
28	L4FM	Layer 4 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 4 Port Number fields. This status is given only when one of the following conditions is true: ■ Layer 3 fields are not enabled and all enabled Layer 4 fields match

Bit	Name	Description
		■ All enabled Layer 3 and Layer 4 filter fields match When more than one filter matches, this bit gives the layer 4 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
27	L3FM	Layer 3 Filter Match When this bit is set, it indicates that the received packet matches one of the enabled Layer 3 IP Address fields. This status is given only when one of the following conditions is true: ■ All enabled Layer 3 fields match and all enabled Layer 4 fields are bypassed ■ All enabled filter fields match When more than one filter matches, this bit gives the layer 3 filter status of filter indicated by Bits[31:29]. Note: This status is not available when Flexible RX Parser is enabled.
26:19	MADRM	MAC Address Match or Hash Value When the HF bit is reset, this field contains the MAC address register number that matched the Destination address of the received packet. This field is valid only if the DAF bit is reset. When the HF bit is set, this field contains the hash value computed by the MAC. A packet passes the hash filter when the bit corresponding to the hash value is set in the hash filter register. Note: This status is not available when Flexible RX Parser is enabled.
18	HF	Hash Filter Status When this bit is set, it indicates that the packet passed the MAC address hash filter. Bits[26:19] indicate the hash value. Note: This status is not available when Flexible RX Parser is enabled.
17	DAF/RXPI	Destination Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the DA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet parsing is incomplete (RXPI) due to ECC error. Note: When this bit is set, ES bit of RDES3 is also set.
16	SAF/RXPD	SA Address Filter Fail When Flexible RX Parser is disabled, and this bit is set, it indicates that the packet failed the SA Filter in the MAC. When Flexible RX Parser is enabled, this bit is set to indicate that the packet is dropped (RXPD) by the parser. Note: When this bit is set, ES bit of RDES3 is also set.
15	OTS	VLAN Filter Status When set, this bit indicates that the VLAN Tag of the received packed passed the VLAN filter. This bit is valid only when GMAC_ERVFE is not enabled. If GMAC_ERVFE is enabled, the bit is redefined as Outer VLAN Tag Filter Status (OTS). This bit is valid for both Single and Double VLAN Tagged frames.
14	ITS	Inner VLAN Tag Filter Status (ITS)

Bit	Name	Description
		This bit is valid only when GMAC_ERVFE is enabled. This bit is valid only for Double VLAN Tagged frames, when Double VLAN Processing is enabled. For more information, see the Filter Status topic.
13:11	Reserved	Reserved
10	ARPNR	ARP Reply Not Generated When this bit is set, it indicates that the MAC did not generate the ARP Reply for received ARP Request packet. This bit is set when the MAC is busy transmitting ARP reply to earlier ARP request (only one ARP request is processed at a time). This bit is reserved when the Enable IPv4 ARP Offload option is not selected.
9:0	HL	L3/L4 Header Length This field contains the length of the header of the packet split by the MAC at L3 or L4 header. boundary as identified by the MAC receiver. This field is valid only when the first descriptor bit is set (FD = 1). The header data is written to the Buffer 1 address of corresponding descriptor. If header length is zero, this field is not valid. It implies that the MAC did not identify and split the header. This field is valid when the Enable Split Header Feature option is selected.

Table 19-22 TDES3 Normal Descriptor(Write-Back Format)

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the GMAC DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context type descriptor. The DMA writes 1'b0 to this bit for normal receive descriptor. When CTXT and FD bits are used together, {CTXT, FD} ■ 2'b00: Intermediate Descriptor ■ 2'b01: First Descriptor ■ 2'b10: Reserved ■ 2'b11: Descriptor Error (due to all 1s) Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	FD	First Descriptor When this bit is set, it indicates that this descriptor contains the first buffer of the packet. If the size of the first buffer is 0, the second buffer contains the beginning of the packet. If the size of the second buffer is also 0, the next descriptor contains the beginning of the packet.

Bit	Name	Description
		See the CTXT bit description for details of using the CTXT bit and FD bit together.
28	LD	Last Descriptor When this bit is set, it indicates that the buffers to which this descriptor is pointing are the last buffers of the packet.
27	RS2V	Receive Status RDES2 Valid When this bit is set, it indicates that the status in RDES2 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
26	RS1V	Receive Status RDES1 Valid When this bit is set, it indicates that the status in RDES1 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
25	RS0V	Receive Status RDES0 Valid When this bit is set, it indicates that the status in RDES0 is valid and it is written by the DMA. This bit is valid only when the LD bit of RDES3 is set.
24	CE	CRC Error When this bit is set, it indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received packet. This field is valid only when the LD bit of RDES3 is set.
23	GP	Giant Packet When this bit is set, it indicates that the packet length exceeds the specified maximum Ethernet size of 1518, 1522, or 2000 bytes (9018 or 9022 bytes if jumbo packet enable is set). Note: Giant packet indicates only the packet length. It does not cause any packet truncation.
22	RWT	Receive Watchdog Timeout When this bit is set, it indicates that the Receive Watchdog Timer has expired while receiving the current packet. The current packet is truncated after watchdog timeout.
21	OE	Overflow Error When this bit is set, it indicates that the received packet is damaged because of buffer overflow in Rx FIFO. Note: This bit is set only when the DMA transfers a partial packet to the application. This happens only when the Rx FIFO is operating in the threshold mode. In the store-and-forward mode, all partial packets are dropped completely in Rx FIFO.
20	RE	Receive Error When this bit is set, it indicates that the gmii_rxer_i signal is asserted while the gmii_rxdv_i signal is asserted during packet reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd!= 0f) during extension.
19	DE	Dribble Bit Error When this bit is set, it indicates that the received packet has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
18:16	LT	Length/Type Field This field indicates if the packet received is a length packet or a type packet. The encoding of the 3 bits is as follows:

Bit	Name	Description
		■ 3'b000: The packet is a length packet ■ 3'b001: The packet is a type packet. ■ 3'b011: The packet is a ARP Request packet type ■ 3'b100: The packet is a type packet with VLAN Tag ■ 3'b101: The packet is a type packet with Double VLAN Tag ■ 3'b110: The packet is a MAC Control packet type ■ 3'b111: The packet is a OAM packet type ■ 3'b010: Reserved
15	ES	Error Summary When this bit is set, it indicates the logical OR of the following bits: ■ RDES3[24]: CRC Error ■ RDES3[19]: Dribble Error ■ RDES3[20]: Receive Error ■ RDES3[22]: Watchdog Timeout ■ RDES3[21]: Overflow Error ■ RDES3[23]: Giant Packet ■ RDES2[17]: Destination Address Filter Fail, when Flexible RX Parser is enabled ■ RDES2[16]: SA Address Filter Fail, when Flexible RX Parser is enabled This field is valid only when the LD bit of RDES3 is set.
14	PL	Packet Length These bits indicate the byte length of the received packet that was transferred to system memory (including CRC). This field is valid when the LD bit of RDES3 is set and Overflow Error bits are reset. The packet length also includes the two bytes appended to the Ethernet packet when IP checksum calculation is enabled and the received packet is not a MAC control packet. This field is valid when the LD bit of RDES3 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 packet.

19.5.10 Receive Context Descriptor

This descriptor is read-only for the application. Only the DMA can write to this descriptor. The context descriptor provides information about the extended status related to the last received packet. The Bit 30 of RDES3 indicates the context type descriptor.

Table 19-23 TDES0 Context Descriptor

Bit	Name	Description
31:0	RSTL	Receive Packet Timestamp Low The DMA updates this field with least significant 32 bits of the timestamp captured for corresponding Receive packet. When this field and the RTSH field of RDES1 show all-ones value, the timestamp must be considered as corrupt.

Table 19-24 TDES1 Context Descriptor

Bit	Name	Description
31:0	RSTH	Receive Packet Timestamp High The DMA updates this field with most significant 32 bits of the timestamp captured for corresponding receive packet. When this field and the RTSL field of RDES0 show all-ones value, the timestamp must be considered as corrupt.

Table 19-25 TDES2 Context Descriptor

Bit	Name	Description
31:0	Reserved	Reserved

Table 19-26 TDES3 Context Descriptor

Bit	Name	Description
31	OWN	Own Bit When this bit is set, it indicates that the DMA owns the descriptor. When this bit is reset, it indicates that the application owns the descriptor. The DMA clears this bit when either of the following conditions is true: ■ The DMA completes the packet reception ■ The buffers associated with the descriptor are full
30	CTXT	Receive Context Descriptor When this bit is set, it indicates that the current descriptor is a context descriptor. The DMA writes 1'b1 to this bit for context descriptor. DMA writes 2'b11 to indicate a descriptor error due to all 1s. When CTXT and DE bits are used together, {CTXT, DE} ■ 2'b00: Reserved ■ 2'b01: Reserved ■ 2'b10: Context Descriptor ■ 2'b11: Descriptor Error Note: When Descriptor Error occurs, the Receive DMA closes the receive descriptor indicating Descriptor Error. This receive descriptor is skipped and the buffer addresses are not used to write the packet data Receive DMA will set the CDE bit in DMA_CH#_Status register but not the RI bit even when IOC is set, as this is not marked as last receive descriptor for the packet. The subsequent valid receive descriptor is used to write the packet data.
29	DE	Descriptor Error See the CTXT bit description for details of using theDE bit along with CTXT bit.
28:0	Reserved	Reserved

19.5.11 Clock Architecture

In RMII mode, reference clock and TX/RX clock be from PHY as following figure. The mux selecting rmii_speed is GRF_GMAC_CLK_CON[2].

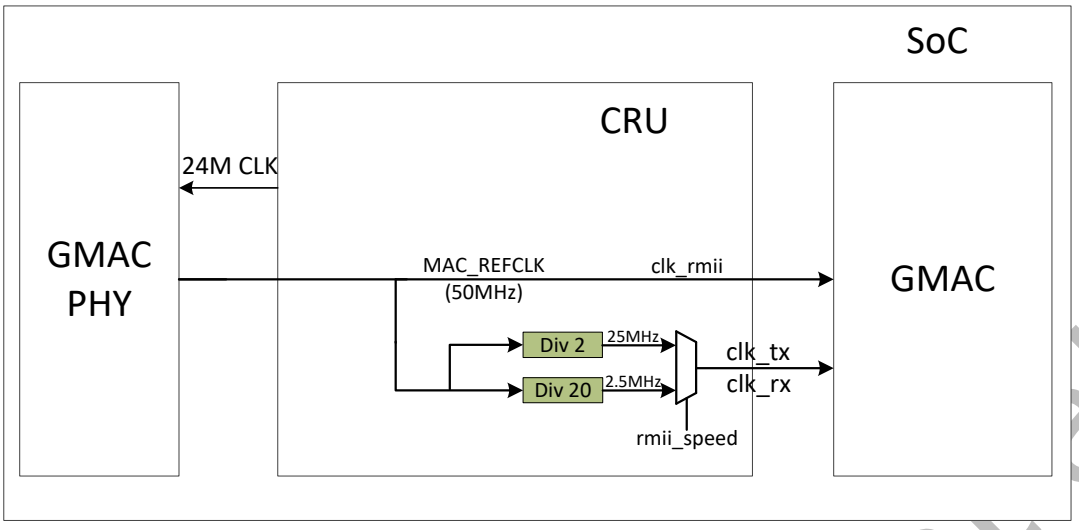


Fig. 19-11 RMII Clock Architecture When Clock Source From PHY

Chapter 20 SAR-ADC

20.1 Overview

The ADC is a 2-channel single-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as its reference which avoids the use of any external reference. The input range is typically 0V to 1.8V.
SAR-ADC controller supports the following features:

- Support single mode and series conversion mode.
- In single mode, the conversion operates once each software is accessed.
- In series conversion, the controller samples each channel then loops until the software stops conversion.
- High/low threshold can be set, higher/lower/between high-low-threshold interrupt can be enabled

20.2 Block Diagram

SAR-ADC block shows in Fig.1-1. This includes:

- APB Interface
- Control FSM
- SAR-ADC PHY

The software can configure the SAR-ADC controller by APB interface, then the inter FSM will communicate with SAR-ADC PHY for limited timing requests.

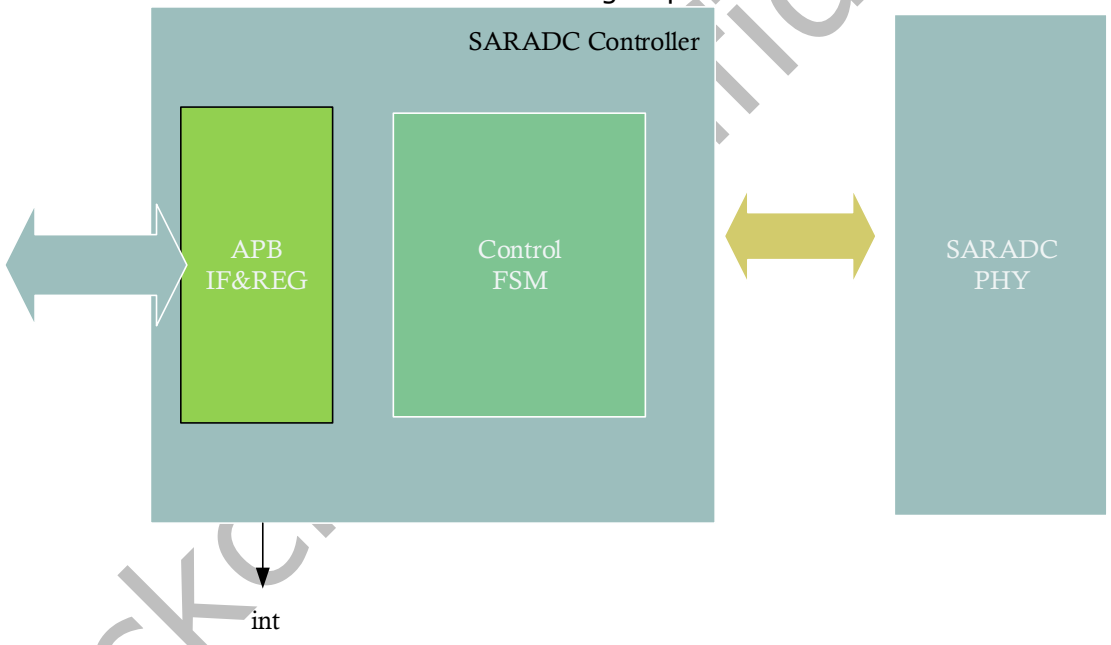


Fig. 20-1 SAR-ADC Block Diagram

20.3 Function Description

SAR-ADC block includes controller and PHY, user cannot directly access SAR-ADC PHY. Software access the SAR-ADC through SAR-ADC controller by APB interface. SAR-ADC controller will sample the conversion result from SAR-ADC PHY.

20.4 Register Description

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_CONV_CON	0x0000	W	0x00000000	Conversion control
SARADC_T_PD_SOC	0x0004	W	0x00026000	Timing control for PD to SOC
SARADC_T_AS_SOC	0x0008	W	0x00000000	Timing control for assert SOC

Name	Offset	Size	Reset Value	Description
SARADC_T_DAS_SOC	0x000C	W	0x00000007	Timing control from dis-assert SOC to change the channel
SARADC_T_SEL_SOC	0x0010	W	0x00000002	Timing control from change channel to assert SOC
SARADC_HIGH_COMP0	0x0014	W	0x00000000	High threshold for adc output data
SARADC_HIGH_COMP1	0x0018	W	0x00000000	High threshold for adc output data
SARADC_LOW_COMP0	0x0054	W	0x00000000	Low threshold for adc output data
SARADC_LOW_COMP1	0x0058	W	0x00000000	Low threshold for adc output data
SARADC_DEBOUNCE	0x0094	W	0x00000003	Threshold debounce
SARADC_HT_INT_EN	0x0098	W	0x00000000	High threshold int enable
SARADC_LT_INT_EN	0x009C	W	0x00000000	Low threshold int enable
SARADC_MT_INT_EN	0x0100	W	0x00000000	Middle threshold int enable
SARADC_END_INT_EN	0x0104	W	0x00000000	End conversion int enable
SARADC_STATUS	0x010C	W	0x00000002	Adc status
SARADC_END_INT_ST	0x0110	W	0x00000000	End conversion int state
SARADC_HT_INT_ST	0x0114	W	0x00000000	High threshold int state
SARADC_LT_INT_ST	0x0118	W	0x00000000	Low threshold int state
SARADC_MT_INT_ST	0x011C	W	0x00000000	Middle threshold int state
SARADC_DATA0	0x0120	W	0x00000000	ADC output data
SARADC_DATA1	0x0124	W	0x00000000	ADC output data
SARADC_AUTO_CH_EN	0x0160	W	0x00000000	Channel enable in auto channel mode

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

20.4.2 Detail Registers Description

SARADC_CONV_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x00	reserved
9	RW	0x0	int_lock This is used to lock the sample data when an interrupt happened. 1'b1: Enable lock. 1'b0: Disable lock.
8	RW	0x0	as_pd_mode If this bit is set to 1'b1, each time conversion ends, PD will be asserted. Then next time conversion starts, PD will be set to low auto automaticlly. This is not used in signal mode.
7	W1C	0x0	end_conv End conversion, this is not used when CONV_CON[5] is set to 1'b1. If this bit is set to 1'b1, PD will be set to 1'b1 after the last conversion and this bit will be cleared to 1'b0.
6	RW	0x0	auto_channel_mode Auto channel mode. If this is enabled, the channel will be round auto according to which is set in AUTO_CH_EN.
5	RW	0x0	single_pd_mode Single conversion mode. If this bit is set to 1, the conversion only operates once, then PD single will be set to 1.

Bit	Attr	Reset Value	Description
4	W1C	0x0	start_adc Enable ADC, if this bit is set to one, the conversion will start. Then this bit will be clear to 0.
3:0	RW	0x0	channel_sel Channel for SARADC, 2 channel is supported. This field is not used when CONV_CON[6] is set to 1. 4'd0: Select channel 0. 4'd1: Select channel 1.

SARADC_T_PD_SOC

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:21	RO	0x000	reserved
20:13	RW	0x13	t_pd_soc Timing control between power up to start-of-conversion.
12:0	RO	0x0000	reserved

SARADC_T_AS_SOC

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t_as_soc Timing control for assert SOC signal.

SARADC_T_DAS_SOC

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000007	t_das_soc Timing from dis-assret SOC to channel change.

SARADC_T_SEL_SOC

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0002	t_sel_soc Timing from channel load to SOC assert.

SARADC_HIGH_COMP0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	high_comp0 High threshold for ADC output data for channel 0.

SARADC_HIGH_COMP1

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	high_comp1 High threshold for ADC output data for channel 1.

SARADC_LOW_COMP0

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	low_comp0 Low threshold for ADC output data for channel 0.

SARADC_LOW_COMP1

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RW	0x000	low_comp1 Low threshold for ADC output data for channel 1.

SARADC_DEBOUNCE

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x03	debounce ADC controller will only generate interrupt data is higher/lower/between the setting threshold for "debounce" times.

SARADC_HT_INT_EN

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	ht_int_en1 High threshold interrupt for channel1. 1'b1: Enable ht intr. 1'b0: Disable ht intr.
0	RW	0x0	ht_int_en0 High threshold interrupt for channel0. 1'b1: Enable ht intr. 1'b0: Disable ht intr.

SARADC_LT_INT_EN

Address: Operational Base + offset (0x009C)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	lt_int_en1 Low threshold interrupt for channel1. 1'b1: Enable ht intr. 1'b0: Disable ht intr.
0	RW	0x0	lt_int_en0 Low threshold interrupt for channel0. 1'b1: Enable ht intr. 1'b0: Disable ht intr.

SARADC_MT_INT_EN

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	mt_int_en1 Middle threshold interrupt for channel1. 1'b1: Enable ht intr. 1'b0: Disable ht intr.
0	RW	0x0	mt_int_en0 Middle threshold interrupt for channel0. 1'b1: Enable ht intr. 1'b0: Disable ht intr.

SARADC_END_INT_EN

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0000	reserved
0	RW	0x0	end_int_en 1'b1: Enable end conversion intr. 1'b0: Disable end conversion intr.

SARADC_STATUS

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:2	RO	0x0	sel ADC channel.
1	RO	0x1	pd 1'b1: ADC is power down. 1'b0: ADC is power up and in conversion.
0	RO	0x0	conv_st Conversion status 1'b1: ADC controller FSM is busy. 1'b0: ADC controller FSM is idle.

SARADC_END_INT_ST

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	W1C	0x0	end_int_st ADC end conversion interrupt status. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.

SARADC_HT_INT_ST

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
1	W1C	0x0	ht_int_st1 High threshold interrupt state for channel1. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.
0	W1C	0x0	ht_int_st0 High threshold interrupt state for channel0. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.

SARADC_LT_INT_ST

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	W1C	0x0	lt_int_st1 Low threshold interrupt state for channel1. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.
0	W1C	0x0	lt_int_st0 Low threshold interrupt state for channel0. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.

SARADC_MT_INT_ST

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	W1C	0x0	mt_int_st1 Between high and low threshold interrupt state for channel1. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.
0	W1C	0x0	mt_int_st0 Between high and low threshold interrupt state for channel0. 1'b1: Interrupt happened. 1'b0: Interrupt not happened.

SARADC_DATA0

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RO	0x000	data0 ADC channel 0 data.

SARADC_DATA1

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:12	RO	0x00000	reserved
11:0	RO	0x000	data1 ADC channel 1 data.

SARADC_AUTO_CH_EN

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0000	reserved
1	RW	0x0	auto_ch1_en Enable channel 1 in auto channel mode.
0	RW	0x0	auto_ch0_en Enable channel 0 in auto channel mode.

20.5 Application Notes

Steps of ADC conversion in series conversion mode:

- Decide which channel should be used and whether auto_channel mode should be set. In auto_channel mode, the channel will be changed from 0-1 then to 0. Please refer to the SARADC_CONV_CON register.
- Set SARADC_CONV_CON[4] to 1’b1, then the conversion will start.
- If conversion wants to be ended, set SARADC_CONV_CON[7] to 1’b1.
- Conversion could be read from SARADC_DATAn (n is from 0-1).
- If threshold compare interrupt wants to be used, high/low threshold could be set, and interrupt should be set as an application.
- If auto_channel mode is not set, you could set the channel as an application.

Steps of ADC conversion in single mode:

- Decide which channel should be used and set SARADC_CONV_CON[3:0].
- Set SARADC_CONV_CON[5] to 1’b1.
- Set SARADC_CONV_CON[4] to 1’b1, then the conversion will start.
- The conversion only operates once then ends. PD will be asserted and PD status could be got by reading SARADC_STATUS.
- Conversion could be read from SARADC_DATAn (n is from 0-1).

Chapter 21 Audio Codec

21.1 Overview

Audio Codec is a low power, high resolution, stereo CODEC solution which employs Sigma-Delta noise-shaping technique. The ADC, DAC and power amplifier are integrated to provide total solutions. With 24 bits resolution for DAC and 24 bits resolution for ADC, Audio Codec is suitable for applications in high end consumer digital audio systems, automobile audio, multimedia and digital systems. It supports following features:

- 24 bits DAC with 90dB SNR
- Support 16~32Ω headphone out and line output
- Low power: 5mA for stereo playback
- 24 bits ADC with 90dB SNR
- Support differential and single-ended microphone or line input
- Low power: 5mA for stereo recording
- Automatic Level Control (ALC) for smooth audio recording
- Low power: less than 0.05mA for standby
- Support Mono, Stereo, 5.1 and 7.1 HiFi channel performance
- Programmable input and output analog gains
- Digital interpolation and decimation filter integrated
- Sampling rate of
- 8kHz/12kHz/16kHz/24kHz/32kHz/44.1kHz/48kHz/64kHz/96kHz
- 1.8V supply for analog and 0.8V supply for digital

21.2 Block Diagram

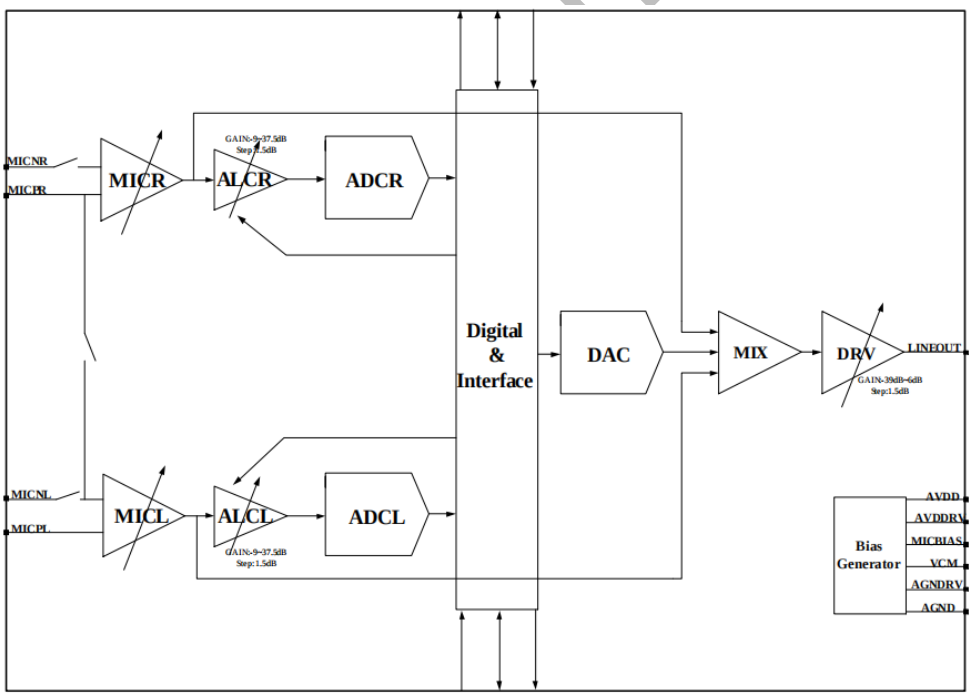


Fig. 21-1 Audio Codec Block Diagram

21.3 Function Description

21.3.1 Digital Interface

Acodec Codec provides the I2S PCM interface of audio data stream which gets into DAC and out from ADC, both of which can be configured in master or slave mode. Different audio data formats are available for different operating modes. This is demonstrated in following table.

Table 21-1 Supported Data Formats in Different Modes

Data Formats	ADC		DAC	
	Master	Slave	Master	Slave
Left Justified	√	×	√	√
Right Justified	√	√	√	√
I2S	√	√	√	√
DSP/PCM mode A	√	√	√	√
DSP/PCM mode B	√	×	√	√

I2S_PCM interface supports five audio data formats: Left Justified mode, Right Justified mode, I2S mode, DSP/PCM mode A and mode B. They are valid when the device operates as a master or slave.

For Left Justified mode, the data format is illustrated in Fig. 1-2.The MSB is valid at the first rising edge of sck after ws transition is done. The other valid bits up to the LSB are transmitted sequentially. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear before every ws transition, which means the data in this period is invalid.

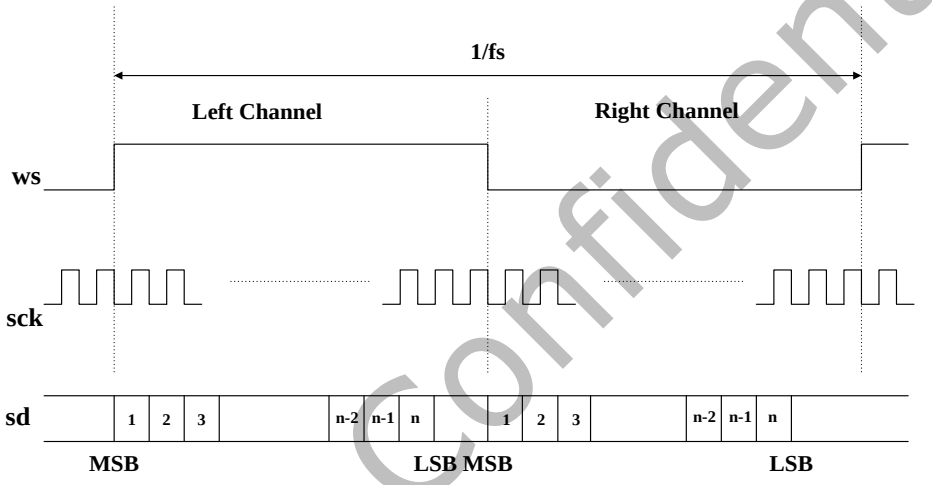


Fig. 21-2 Left Justified Mode (assuming n-bit word length)

For Right Justified mode, the data format is shown in Fig. 1-3. The LSB becomes valid at the last rising edge of sck before ws transition is done. As the MSB is transmitted first, the other valid bits up to the MSB are followed in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may exist after every ws transition, which means the data in this period is invalid.

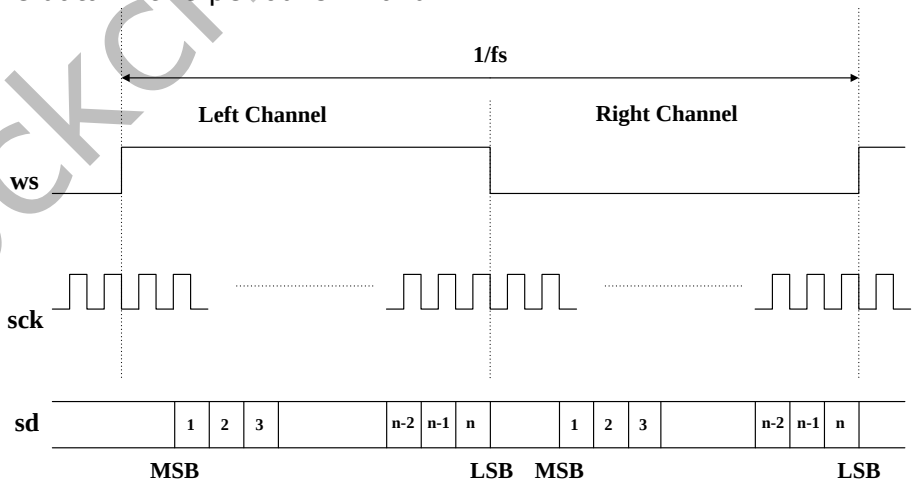


Fig. 21-3 Right Justified Mode (assuming n-bit word length)

For I2S mode, the data format is depicted in Fig. 1-4. The MSB becomes available at the second rising edge of sck when ws transition is done. The other valid bits up to the LSB are transmitted in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear between the LSB of the current sample and the MSB of the next one, which means the data in this period can be ignored.

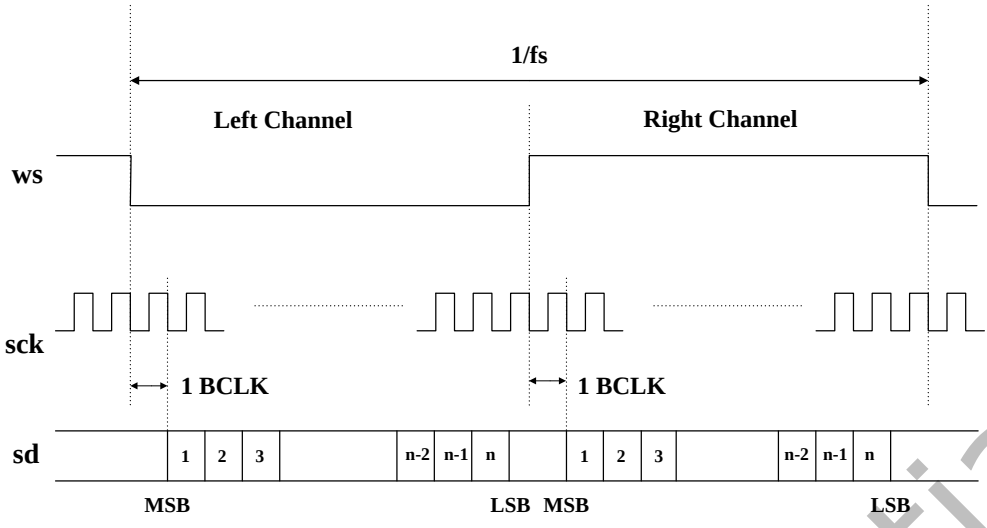


Fig. 21-4 I2S Mode (assuming n-bit word length)

For DSP/PCM mode, the left channel data is transmitted first, followed by right channel data. For DSP/PCM mode A/B, the MSB is available at the second and first rising edge of sck after the rising edge of ws respectively, as shown in Fig. 1-5 and Fig.1-6. Based on word length, sck frequency and sample rate, there may be some invalid data between the LSB of the right channel data and the next sample.

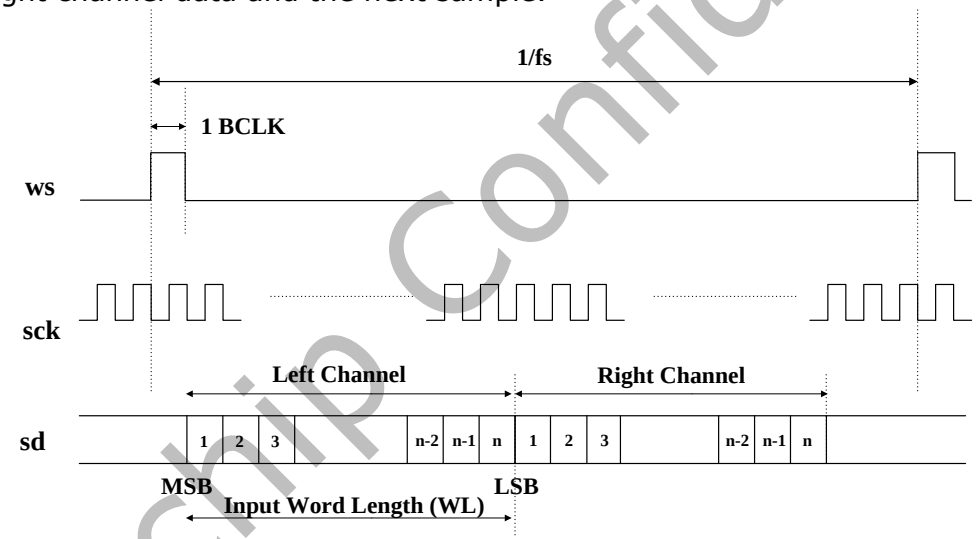


Fig. 21-5 DSP/PCM Mode A (assuming n-bit word length)

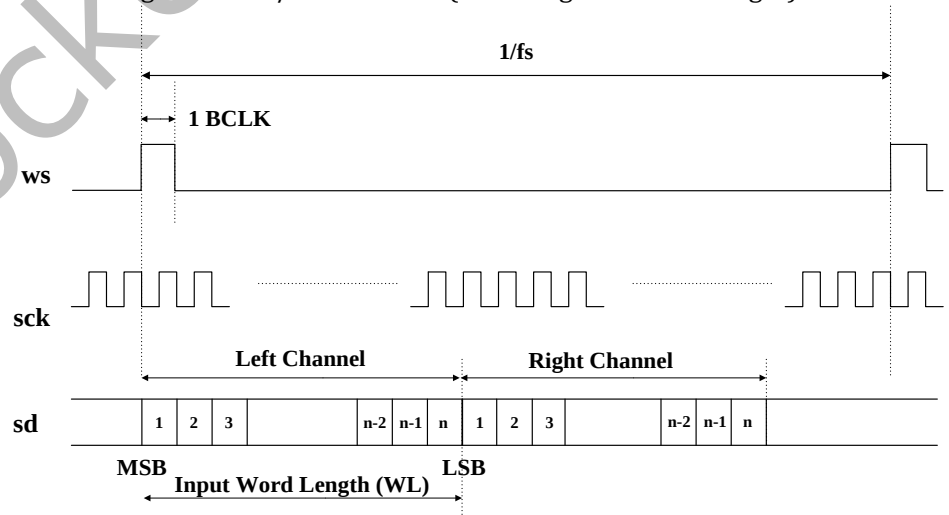


Fig. 21-6 DSP/PCM Mode B (assuming n-bit word length)

21.3.2 Analog Interface

Fig. 21-7 Microphone Input

There are two inputs channels named left ADC channel and right ADC channel. In the each channel, there are two inputs which are configured as differential input by the microphone PGA (MICL and MICR).

In the left channel, microphone inputs are MICPL and MICNL. In the right channel, microphone inputs are MICPR and MICRL.

Microphone PGA has a gain range from 0dB to 20dB.

Automatic Level Control (ALC) function is included to adjust the signal level, which is input into ADC. ALC will measure the signal magnitude and compare it to defined threshold. Then it will adjust the ALC controlled PAG (ALC_L and ALC_R) gain according to the comparison result. The programmable gain range of ALC controlled PAG is from -18dB to +28.5dB. The tuning step is 1.5dB.

Audio Codec supports headphone output or line output configurations. The output can drive load through DC-blocking capacitor.

In the configuration using DC-blocking capacitor, shown in the following figure, the headphone ground is connected to the real ground. The capacitance and the load resistance determine the lower cut-off frequency. For instance, if 600Ω load and 4.7uF DC-blocking capacitor are used, the lower cut-off frequency is:

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 600 \times 4.7 \times 10^{-6}} = 56.5Hz$$

The DC-blocking capacitor can be increased to lower the cut-off frequency for better bass response.

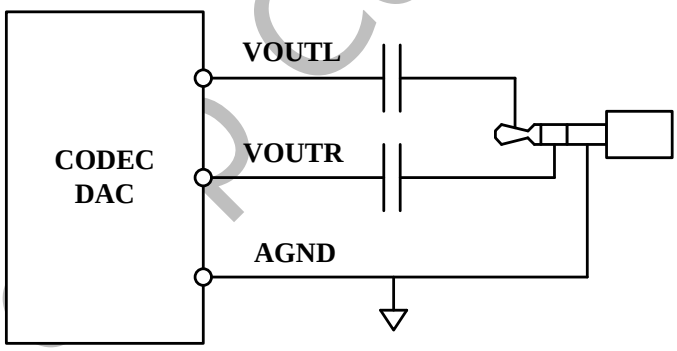


Fig. 21-8 Output DC-blocking capacitor

The out driver has a gain range from -39dB to +6dB with a tuning step of 1.5dB.

The output of the Microphone bias is used for bias external microphones. The bias voltage can vary from 0.5*AVDD to 0.85* AVDD with a step of 0.05*AVDD.

21.3.3 Interface Relationship

The relationships between I2S interface and parallel audio data for different ADC and DAC channels are shown in the following figure.

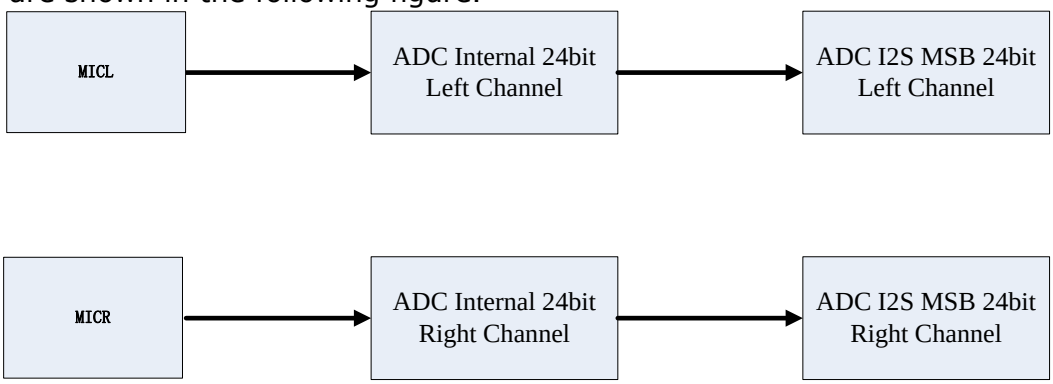
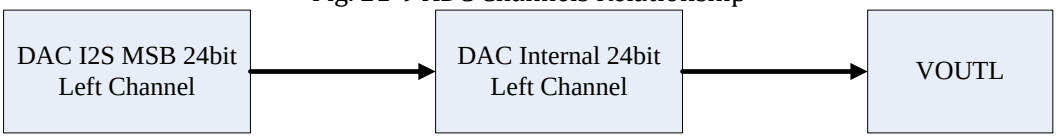


Fig. 21-9 ADC Channels Relationship



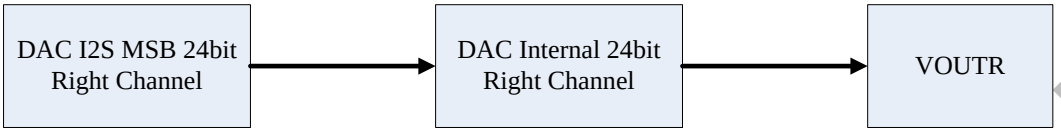


Fig. 21-10 DAC Channels Relationship

21.4 Register Description

21.4.1 Internal Address Mapping

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

21.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
ACODEC_REG0	0x0000	W	0x00000003	
ACODEC_REG1	0x0004	W	0x00000007	
ACODEC_REG2	0x0008	W	0x00000050	
ACODEC_REG3	0x000C	W	0x0000000E	
ACODEC_REG4	0x0010	W	0x00000050	
ACODEC_REG5	0x0014	W	0x0000000E	
ACODEC_REG6	0x0018	W	0x000000F1	
ACODEC_REG7	0x001C	W	0x00000000	
ACODEC_REG8	0x0020	W	0x000000C3	
ACODEC_REG9	0x0024	W	0x000000C3	
ACODEC_REG10	0x0028	W	0x00000084	
ACODEC_REG11	0x002C	W	0x000000F1	
ACODEC_REG12	0x0030	W	0x000000F1	
ACODEC_REG13	0x0034	W	0x00000003	
ACODEC_REG14	0x0038	W	0x000000C3	
ACODEC_REG32	0x0050	W	0x000000C7	
ACODEC_REG33	0x0054	W	0x00000000	
ACODEC_REG34	0x0058	W	0x00000000	
ACODEC_REG35	0x005C	W	0x00000000	
ACODEC_REG36	0x0060	W	0x00000007	
ACODEC_REG37	0x0064	W	0x00000000	
ACODEC_REG38	0x0068	W	0x0000000A	
ACODEC_REG39	0x006C	W	0x0000000A	
ACODEC_REG40	0x0070	W	0x00000000	
ACODEC_REG41	0x0074	W	0x00000010	
ACODEC_REG42	0x0078	W	0x00000002	
ACODEC_REG43	0x007C	W	0x00000000	

Name	Offset	Size	Reset Value	Description
ACODEC_REG47	0x0080	W	0x00000021	
ACODEC_REG64	0x00A0	W	0x00000000	
ACODEC_REG65	0x00A4	W	0x00000046	
ACODEC_REG66	0x00A8	W	0x00000041	
ACODEC_REG67	0x00AC	W	0x0000002C	
ACODEC_REG68	0x00B0	W	0x00000000	
ACODEC_REG69	0x00B4	W	0x00000026	
ACODEC_REG70	0x00B8	W	0x00000040	
ACODEC_REG71	0x00BC	W	0x00000036	
ACODEC_REG72	0x00C0	W	0x00000020	
ACODEC_REG73	0x00C4	W	0x00000038	

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

21.4.3 Detail Register Description

ACODEC_REG0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x1	codec digital core reset This reset only reset the codec data path. 0: Reset 1: Work
0	RW	0x1	codec system reset This signal will reset the registers which control all the digital and analog part. 0: Reset 1: Work

ACODEC_REG1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	dac_mute_en The enable signal of mute dac 0:Disable 1:Enable
6:4	RO	0x0	reserved
3	RW	0x0	dither_level Select dither data which has different high bits completion number.
2	RW	0x1	da_enable When the value is 1 activated DEM function.
1	RW	0x1	dither_enable When the value is 1 activated dither function.
0	RW	0x1	dither_Sign 0:High bits add ~lfsr 1:High bits add 0

ACODEC_REG2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	lrc polarity 0: Normal 1: Reversal
6:5	RW	0x2	frame_width ADC I2S configuration register[6:5] valid word Length in one 1/2frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
4:3	RW	0x2	adc_ode 11: PCM mode 10: I2S mode 01: Left justified mode 00: Right justified mode Note: Same word length in 1/2frame and valid data is not supported in right justified mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RW	0x0	adc_i2s output ADC_I2S output mix selection 0: ADC IIS interface output ADC data 1:ADC IIS interface output ADC left data and DAC left data received from DAC IIS interface
1:0	RW	0x0	adc_data_select I2s adc data select

ACODEC_REG3

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	mode3 DAC I2S mode select for IO pin 1: Master mode 0: Slave mode
6	RW	0x0	mode2 DAC I2S mode select for inner module 1: Master mode 0: Slave mode
5	RW	0x0	mode1 ADC I2S mode select for IO pin 1: Master mode 0: Slave mode
4	RW	0x0	mode0 ADC I2S mode select for inner module 1: Master mode 0: Slave mode
3:2	RW	0x3	adc frame length 01: Length is 24 11: Length is 32
1	RW	0x1	reset ADC I2S configuration register[9] reset 0: Reset 1: Work

Bit	Attr	Reset Value	Description
0	RW	0x0	bit clock polarity 0: Normal 1: Reversal

ACODEC_REG4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	lrc polarity DAC I2S Configuration Register[7] LRC Polarity 0: Normal 1: Reversal
6:5	RW	0x2	frame_width DAC I2S Configuration Register[6:5] Valid Word Length in one 1/2Frame 11: 32 bits 10: 24 bits 01: 20 bits 00: 16 bits
4:3	RW	0x2	mode DAC I2S configuration register[4:3] mode 11: PCM mode 10: I2S mode 01: Left justified mode 00: Right justified mode Note-1: Same word length in 1/2frame and valid data is not supported in right justified mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RW	0x0	swap DAC I2S configuration register[2] DAC left-right swap 0: Normal 1: Swap
1:0	RO	0x0	reserved

ACODEC_REG5

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	de_emphasis DAC de-emphasis filter selection 00: No de-emphasis filter work 01: 32K sample de-emphasis filter 10: 44.1K sample de-emphasis filter 11: 48K sample de-emphasis filter
3:2	RW	0x3	frame_width DAC I2S configuration register[11:10] 1/2frame word length 11: 32 bits(See Note 1) 10: 24 bits 01: 20 bits 00: 16 bits
1	RW	0x1	reset DAC I2S configuration register[9] reset 0: Reset 1: Work

Bit	Attr	Reset Value	Description
0	RW	0x0	bit clock polarity DAC I2S configuration register[8] bit clock polarity 0: Normal 1: Reversal

ACODEC_REG6

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xf1	digital_gain DAC digital gain selection every step 0.5dB 8hf1: 0dB 8hff: 7dB 8hfe: 6.5dB 8h01: -120dB 8h00: Digital mute

ACODEC_REG7

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	bist_mode DAC left channel bist mode select 01: Sine wave to the PCM bist mode enable and need to set the 0x00[7]) Other: Normal left channel data to the PCM normal mode
3:0	RO	0x0	reserved

ACODEC_REG8

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xc3	digital_volume Left ADC digital volume control 8'h0: Digital mute 8'h1: -97dB 8'h2: -96.5dB ... 0.5dB steps up to 8'hff: +30dB

ACODEC_REG9

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xc3	digital_volume Left ADC digital volume control 8'h0: Digital mute 8'h1: -97dB 8'h2: -96.5dB ... 0.5dB steps up to 8'hff: +30dB

ACODEC_REG10

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x1	filter enable 1: Enable 0: Disable
6	RW	0x0	agc select 1: Filter output 0: Sinc output
5	RW	0x0	gain1 0: Choose the register0x26[4:0] to control the gain of the PGA 1: Choose the ALCL module to control the gain of the PGA
4	RW	0x0	gain0 0: Choose the register0x27[4:0] to control the gain of the PGA 1: Choose the ALCR module to control the gain of the PGA
3:2	RW	0x1	hdc ADC HPF disable control 1: Disable HPF 0: Enable HPF
1	RW	0x0	rcdpi ADC right channel data polarity invert 1: Right channel data polarity invert 0: Right channel data polarity not inverted
0	RW	0x0	lxdpi ADC left channel data polarity invert 1: Left channel data polarity invert 0: Left channel data polarity not inverted

ACODEC_REG11

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xf1	delay_cnt1 Initial delay cnt1

ACODEC_REG12

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0xf1	delay_cnt2 Initial delay cnt2

ACODEC_REG13

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:5	RW	0x0	cbms ADC channel bist mode select 01: Sine wave to the PCM(bist mode enable and need to set the 0x00[7]) Other: Normal left channel data to the PCM
4:0	RW	0x03	dco DAC config output[11:8]

ACODEC_REG14

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0xc3	dco DAC config output[7:0]

ACODEC_REG32

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0xc	eh En_hpoutl[7] ini_hpoutl[6] en_hpoutr[5] ini_hpoutr[4]
3:0	RW	0x7	diss DAC ibias select signal

ACODEC_REG33

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	sctp The control signal to select current to precharge discharge. 7 0: Choose the Current I0 1: Dont choose the current I0 6 0: Choose the Current 2I0 1: Dont choose the current 2I0 5 0: Choose the Current 4I0 1: Dont choose the current 4I0 4 0: Choose the Current 8I0 1: Dont choose the current 8I0 3 0: Choose the Current 16I0 1: Dont choose the current 16I0 2 0: Choose the Current 32I0 1: Dont choose the current 32I0 1 0: Choose the Current 64 I0 1: Dont choose the current 64 I0 0 0: Choose the Current 128 I0 1: Dont choose the current 128 I0

ACODEC_REG34

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	srvm The enable signal of reference voltage module: 0: Disable 1: Enable
4	RW	0x0	scs The enable signal of current source for ADC. 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	micbias The enable signal of MIC bias voltage (MICBIAS) buffer 0: Stop woring 1: Work
2:0	RW	0x0	mv The level range control signal of MICBIAS voltage: 000: 0.8*AVDD 100: 0.9*AVDD 001: 0.825*AVDD 101: 0.925*AVDD 010: 0.85*AVDD 110: 0.95*AVDD 011: 0.875*AVDD 111: 0.975*AVDD

ACODEC_REG35

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	mmmal The mute signal of MIC module in the ADCL. 0: Mute 1: Work
6	RW	0x0	mmal The initial signal of MIC module in the ADCL. 0: Initialization 1: Work
5	RW	0x0	rbbad The enable signal of reference voltage buffer of ADCL. 0: Disable 1: Enable
4	RW	0x0	aizcd The enable signal of the ADCL input zero- crossing detection module: 0: Stop Working , output 0 electrical level 1: Work
3	RW	0x0	mmia The mute signal of MIC module in the ADCR. 0: Mute 1: Work
2	RW	0x0	mma The initial signal of MIC module in the ADCR. 0: Initialization 1: Work
1	RW	0x0	rvba The enable signal of reference voltage buffer of ADCR. 0: Disable 1: Enable
0	RW	0x0	aizc The enable signal of the ADCR input zero- crossing detection module: 0: Stop Working , output 0 electrical level 1: Work

ACODEC_REG36

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	gsmml The gain signal of MIC module in the ADC left channel. 00: Reserved 10: 20dB 01: 0dB 11: 20dB
5:4	RO	0x0	gsmmr The gain signal of MIC module in the ADC right channel. 00: Reserved 10: 20dB 01: 0dB 11: 20dB
3:0	RO	0x7	adcis ADC ibias select

ACODEC_REG37

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	ssadcl The mode select signal in the ADC left channel 00: Full differential 01: Single-ended 10: Single-ended 11: Reserved
5	RW	0x0	alcle ALCL enable 0: Disable 1: Enable
4	RW	0x0	micle MICL enable 0: Disable 1: Enable
3:2	RW	0x0	ssadcr The mode select signal in the ADC right channel 00: Full differential 01: Single-ended 10: Single-ended 11: Reserved
1	RW	0x0	alcre ALCR enable 0: Disable 1: Enable
0	RW	0x0	micre MICR enable 0: Disable 1: Enable

ACODEC_REG38

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x0a	gsalcl The gain signal of ALC module in the ADC left channel. 5'h00: -9dB 5'h0b: 7.5dB 5'h16: 24dB 5'h01: -7.5dB 5'h0c: 9dB 5'h17: 25.5dB 5'h02: -6dB 5'h0d: 10.5dB 5'h18: 27dB 5'h03: -4.5dB 5'h0e: 12dB 5'h19: 28.5dB 5'h04: -3dB 5'h0f: 13.5dB 5'h1a: 30dB 5'h05: -1.5dB 5'h10: 15dB 5'h1b: 31.5dB 5'h06: 0dB 5'h11: 16.5dB 5'h1c: 33dB 5'h07: 1.5dB 5'h12: 18dB 5'h1d: 34.5dB 5'h08: 3dB 5'h13: 19.5dB 5'h1e: 36dB 5'h09: 4.5dB 5'h14: 21dB 5'h1f: 37.5dB 5'h0a: 6dB 5'h15: 22.5dB

ACODEC_REG39

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0xa	gsalcr The gain signal of ALC module in the ADC left channel. 5'h00: -9dB 5'h0b: 7.5dB 5'h16: 24dB 5'h01: -7.5dB 5'h0c: 9dB 5'h17: 25.5dB 5'h02: -6dB 5'h0d: 10.5dB 5'h18: 27dB 5'h03: -4.5dB 5'h0e: 12dB 5'h19: 28.5dB 5'h04: -3dB 5'h0f: 13.5dB 5'h1a: 30dB 5'h05: -1.5dB 5'h10: 15dB 5'h1b: 31.5dB 5'h06: 0dB 5'h11: 16.5dB 5'h1c: 33dB 5'h07: 1.5dB 5'h12: 18dB 5'h1d: 34.5dB 5'h08: 3dB 5'h13: 19.5dB 5'h1e: 36dB 5'h09: 4.5dB 5'h14: 21dB 5'h1f: 37.5dB 5'h0a: 6dB 5'h15: 22.5dB

ACODEC_REG40

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	isalcl The initial signal of ALC module in the ADCL. 0: Initialization 1: Work
6	RW	0x0	esckl The enable signal of CLOCK for ADCL module 0: Set to logic 1 1: Work
5	RW	0x0	esadcl The enable signal of ADC module in the ADCL. 0: Disable 1: Enable
4	RW	0x0	isadcl The initial signal of ADC module in the ADCL. 0: Initialization 1: Work

Bit	Attr	Reset Value	Description
3	RW	0x0	isalcr The initial signal of ALC module in the ADCR. 0: Initialization 1: Work
2	RW	0x0	esckr The enable signal of CLOCK for ADCR module 0: Set to logic 1 1: Work
1	RW	0x0	esadcr The enable signal of ADC module in the ADCR. 0: Disable 1: Enable
0	RW	0x0	isadcr The initial signal of ADC module in the ADCR. 0: Initialization 1: Work

ACODEC_REG41

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	esibsdac The enable signal of ibias for DAC channel. 0: Disable 1: Enable
6	RW	0x0	esrvbdacr The enable signal of reference voltage buffer in the right DAC channel. 0: Disable 1: Enable
5:4	RW	0x1	cspdacl The control signal of POP sound in the left DAC channel. 01: Initialization 10: Work
3	RW	0x0	esrvdacl The enable signal of reference voltage of DACL module. 0: Disable 1: Enable
2	RW	0x0	esclkdacr The enable signal of clk in the right DAC channel. 0: Disable 1: Enable
1	RW	0x0	esssdac The enable signal of source signal for DAC channel. 0: Disable 1: Enable
0	RW	0x0	isdacl The initial signal of DAC module in the DACL. 0: Initialization 1: Work

ACODEC_REG42

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	mshpdacl The mute signal of HPOUTL module for DACL 0: Disable 1: Enable
5	RW	0x0	ishpdacl The Initial signal of HPOUTL for DACL module 0: Initialization 1: Work
4	RW	0x0	eshpdacl The enable signal of HPOUTL for DACL module 0: Disable 1: Enable
3:0	RW	0x2	cdrvd Used to change DRV driver strength 0001: 100% 0101: 29.7% 0010: 60.1% 0110: 25.7% 0011: 44.2% 0111: 22.7% 0100: 35.5% 1000: 20.4%

ACODEC_REG43

Address: Operational Base + offset (0x007C)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	gsdrvdac The gain signal of DRV module in the DAC channel. 5'h00: -39dB 5'h0b: -22.5dB 5'h16: -6dB 5'h01: -37.5dB 5'h0c: -21dB 5'h17: -4.5dB 5'h02: -36dB 5'h0d: -19.5dB 5'h18: -3dB 5'h03: -34.5dB 5'h0e: -18dB 5'h19: -1.5dB 5'h04: -33dB 5'h0f: -16.5dB 5'h1a: 0dB 5'h05: -31.5dB 5'h10: -15dB 5'h1b: 1.5dB 5'h06: -30dB 5'h11: -13.5dB 5'h1c: 3dB 5'h07: - 28.5dB 5'h12: -12dB 5'h1d: 4.5dB 5'h08: -27dB 5'h13: -10.5dB 5'h1e: 6dB 5'h09: -25.5dB 5'h14: -9dB 5'h1f: 6dB 5'h0a: -24dB 5'h15: -7.5dB

ACODEC_REG47

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	esmdac the enable signal of mix module in the DAC channel 0: Disable 1: Enable
6:5	RW	0x1	gsmdac the gain signal of mix module in the DAC channel 01: 0db 10: 6db
4	RW	0x0	ismdac the initial signal of mix module in the dac channel 0: Initialization 1: Work
3	RW	0x0	msmdac the mute signal of mix module in the dac channel 0: Mute 1:Work

Bit	Attr	Reset Value	Description
2:0	RW	0x1	ssmdac the select signal of mix module in the dac channel 001: Select input from i2s 010: Select line input from adcl 100: Select line input from adcr Others reserved

ACODEC_REG64

Address: Operational Base + offset (0x00A0)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	cmcga Choose the method to control the gain attack. 0 : Normal way 1 : Jack way
5:4	RW	0x0	fmgcs There are four methods to generate the control signals 00: Normal way 01: Jack way 1 10: Jack way 2 11: Jack way 3 This register is used to choose the method to generate the control signals according to the actual situation.
3:0	RW	0x0	agcht AGC hold time before gain is increased in normal mode. 0000: 0ms 0110: 64ms 0001: 2ms 0111: 128ms 0010: 4ms 1000: 256ms 0011: 8ms 1001: 512ms 0100: 16ms 1010: 1s 0101: 32ms 1011~1111: 0ms

ACODEC_REG65

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:4	RO	0x4	decay_time Decay (gain ramp-up) time normal mode(reg_agc_mode =0) 0000: 500us 0110: 32ms 0001: 1ms 0111: 64ms 0010: 2ms 1000: 128ms 0011: 4ms 1001: 256ms 0100: 8ms 1010: 512ms 0101: 16ms 1011~1111: 512ms Limter mode(reg_agc_mode =1) 0000: 125us 0110: 8ms 0001: 250us 0111: 16ms 0010: 500us 1000: 32ms 0011: 1ms 1001: 64ms 0100: 2ms 1010: 128ms 0101: 4ms 1011~1111: 128ms
3:0	RO	0x6	attack_time Attack (gain ramp-down) time normal mode(reg_agc_mode =0) 0000: 125us 0110: 8ms 0001: 250us 0111: 16ms 0010: 500us 1000: 32ms 0011: 1ms 1001: 64ms 0100: 2ms 1010: 128ms 0101: 4ms 1011~1111: 125us Limter mode(reg_agc_mode =1) 0000: 32us 0110: 2ms 0001: 64us 0111: 4ms 0010: 125us 1000: 8ms 0011: 250us 1001: 16ms 0100: 500us 1010: 32ms 0101: 1ms 1011~1111: 32us

ACODEC_REG66

Address: Operational Base + offset (0x00A8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	sagcm Determines the AGC mode of operation 0: AGC mode(normal mode) 1: Limiter mode
6	RW	0x1	agcuzc AGC users zero cross enable 0: Disabled 1: Enabled, the AGC gain will update at zero cross enable
5	RW	0x0	las When in the limiter mode, the low amplitude signal will recovery in two modes: 0: The gain will recovery to the value of the reg_pga_lvol 1: The gain will recovery to the gain at the moment when the mode changes from AGC to Limiter.
4	RW	0x0	scfd When the amplitude of the signal is more than 87.5% of the Full scale, use this signal to control the fast decrement: 0: Disabled 1: Enabled
3	RW	0x0	agcngfe AGC noise gate function enable 0: Disabled 1: Enabled
2:0	RW	0x1	agcng AGC noise gate threshold 000: -39dB 100: -63dB 001: -45dB 101: -69dB 010: -51dB 110: -75dB 011: -57dB 111: -81dB

ACODEC_REG67

Address: Operational Base + offset (0x00AC)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x1	lcipgazc Left channel input PGA zero cross enable: 0: Update gain when gain register changes. 1: Update gain on 1st zero cross after gain register write.
4:0	RW	0x0c	lcipgag Left channel input PGA gain 5'h00: -18dB 5'h0d: +1.5dB 5'h01: -16.5dB 5'h0e: +3dB ... ... 5'h0c: 0dB 5'h1f: +28.5dB

ACODEC_REG68

Address: Operational Base + offset (0x00B0)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	sclken Slow clock enabled, used for the zero cross timeout.
2:0	RW	0x0	asr Approximate sample rate Sample Rate 96kHz: 000 24kHz: 100 48kHz: 001 16kHz: 101 44.1kHz: 010 12kHz: 110 32kHz: 011 8kHz: 111 Note According to the sample rate to choose the right configuration

ACODEC_REG69

Address: Operational Base + offset (0x00B4)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x26	lagc_max The high 8 bits of the AGC maximum level

ACODEC_REG70

Address: Operational Base + offset (0x00B8)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x40	hagc_max Integer clock divider to provide 6.144/5.644/4.096MHz sample clock for internal filters. Make sure that int_div_con is an odd number between 7(8 times division) and 15(16 times division).

ACODEC_REG71

Address: Operational Base + offset (0x00BC)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x36	lagc_min The low 8 bits of the AGC minimum level

ACODEC_REG72

Address: Operational Base + offset (0x00C0)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x20	hagc_min The high 8 bits of the AGC minimum level

ACODEC_REG73

Address: Operational Base + offset (0x00C4)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	agcfs AGC function select 0: AGC function off 1: AGC function enable

Bit	Attr	Reset Value	Description
5:3	RW	0x7	smaxg_pga Set maximum gain of PGA 000: -13.5dB 100: +10.5dB 001: -7.5dB 101: +16.5dB 010: -1.5dB 110: +22.5dB 011: +4.5dB 111: +28.5dB
2:0	RW	0x0	sming_pga Set minimum gain of PGA Configurati on Min Gain 000: -18dB 100: +6dB 001: -12dB 101: +12dB 010: -6dB 110: +18dB 011: 0dB 111: +24dB

21.5 Interface Description

N/A

21.6 Application Notes

21.6.1 Enable DAC I2S Interface Flow

0. Provide stable clock to the pin_sys_clk and pin_mclk
1. Reset the IP by low the pin_rst_n.
2. Reset the IP by soft reset register as follow:
Set REG0=0x0
Set REG0=0x3
3. Configure the register related to the DAC I2S function to make the I2S RX work in different mode.
4. Begin to input the I2S data stream to the I2S interface of the DAC.

21.6.2 Enable DC I2S Interface Flow

0. Provide stable clock to the pin_sys_clk and pin_mclk.
1. Reset the IP by low the pin_rst_n.
2. Reset the IP by soft reset register as follow:
Set REG0=0x0
Set REG0=0x3
3. Configure the register related to the ADC I2S function to make the I2S TX work in different mode.
4. Begin to use the I2S interface of ADC to output the parallel data.

21.6.3 Power Up

0. Supply the power of digital part and reset the Audio Codec.
1. Configure the register 0x29[5:4] to 2'b01, to setup dc voltage of the DAC channel output.
2. Configure the register reg0x21[7:0] to 8'b000_0001.
3. Supply the power of the analog part .
4. Configure the register reg0x22[5] to 1 to setup reference voltage.
5. Change the register reg0x21[7:0] from the 8' b0000_0001 to 8'b1111_1111 step by step or configure the reg0x21[7:0] to 7'b1111_1111 directly. The suggestion slot time of the step is 20ms.

6. Wait until the voltage of VCM keeps stable at the AVDD/2.
7. Configure the register reg0x21[7:0] to the appropriate value(except 7'b0000_00000) for reducing power.

21.6.4 Power Off

0. Keep the power on and disable the DAC and ADC path according to the description of section 12.1.
1. Configure the register reg0x21[7:0] to 8'b0000_0001.
2. Configure the register reg0x22[5] to 1'b0.
3. Change the register reg0x21[7:0] from the 8'b0000_0001 to 8'b111_1111 step by step or configure the reg0x21[7:0] to 8'b111_1111 directly. The suggestion slot time of the step is 20ms.
4. Wait until the voltage of VCM keep stable at AGND.
5. Power off the analog power supply
6. Power off the digital power supply.

21.6.5 Enable DAC

0. Power up the Audio Codec and input the mute signal.
1. Set the register 0x20[3:0] to 1111, to enable the current source of DAC.
2. Set the register 0x29[6] to 1, to enable the reference voltage buffer of the DAC channel.
3. Set the register 0x29[5:4] to 2' b10, to enable POP sound in the DAC channel.
4. Set the register 0x2a[4] to 1, to enable the HPOUT module in the DAC channel.
5. Set the register 0x2a[5] to 1, to end the initialization of the HPOUT module in the DAC channel.
6. Set the register 0x29[3] to 1, to enable the reference voltage of DAC module.
7. Set the register 0x29[2] to 1, to enable the clock module of DAC module.
8. Set the register 0x29[1] to 1, to enable the DAC module.
9. Set the register 0x29[0] to 1, to end the initialization of the DAC module.
10. Set the register 0x2a[6] to 1, to end the mute station of the HPOUT module.
11. Set the register 0x2b[4:0], to select the gain of HPOUT module.
12. Play the music.

21.6.6 Disable DAC

0. Keep the DAC channel work and input the mute signal.
1. Set the register 0x2b[4:0] to 5' b0_0000, to select the gain of the HPOUT.
2. Set the register 0x2a[6] to 0, to mute the HPOUT module.
3. Set the register 0x2a[5] to 0, to initialize the DAC module.
4. Set the register 0x2a[4] to 0, to disable the HPOUT module .
5. Set the register 0x29[1] to 0 to disable the DAC module.
6. Set the register 0x29[2] to 0, to disable the clock module of DAC module.
7. Set the register 0x29[3] to 0, to disable the reference voltage of DAC module.
8. Set the register 0x29[1:0] to 2' b01 to initialize the POP sound.
9. Set the register 0x29[6] to 0, to disable the reference voltage buffer.
10. Set the register 0x20[3:0] to 0000 to disable the current source of DAC.
11. Set the register 0x29[0] to 0, to begin the initialization of the HPOUT module.

21.6.7 Enable ADC

0. Power up the Audio Codec.
1. Configure the register 0x26[7:6], to select single-ended or difference as input of the ADC left channel.
Configure the register 0x26[3:2], to select the single-ended or difference as input of the ADC right channel.
2. Configure the register 0x23[7] to 1, to end the mute station of the left ADC channel.
Configure the register 0x23[3] to 1, to end the mute station of the ADC right channel.
3. Configure the register 0x22[4] to 1, to enable the current source of audio.
4. Configure the register 0x23[5] to 1, to enable the reference voltage buffer in ADC left channel.
Configure the register 0x23[1] to 1, to enable the reference voltage buffer in ADC right channel.
5. Configure the register 0x25[4] to 1, to enable the MIC module in ADC left channel.
Configure the register 0x25[0] to 1, to enable the MIC module in ADC right channel.

6. Configure the register 0x25[5] to 1, to enable the ALC module in ADC left channel.
Configure the register 0x25[1] to 1, to enable the ALC module in ADC right channel.
7. Configure the register 0x28[2] to 1, to enable the clock module in ADC right channel.
Configure the register 0x28[6] to 1, to enable the clock module in ADC left channel.
8. Configure the register 0x28[1] to 1, to enable the ADC module in ADC right channel.
Configure the register 0x28[5] to 1, to enable the ADC module in ADC left channel.
9. Configure the register 0x28[0] to 1, to end the initialization of the ADCR module.
Configure the register 0x28[4] to 1, to end the initialization of the ADCL module.
10. Configure the register 0x28[3] to 1, to end the initialization of the ALC right module.
Configure the register 0x28[7] to 1, to end the initialization of the ALC left module.
11. Configure the register 0x23[2] to 1, to end the initialization of the MIC right module.
Configure the register 0x23[6] to 1, to end the initialization of the MIC right module.
12. Configure the register 0x24[5:4], to select the gain of the MIC right module.
Configure the register 0x24[7:6], to select the gain of the MIC left module.
13. Configure the register 0x26[4:0], to select the gain of the ALC left module.
Configure the register 0x27[4:0], to select the gain of the ALC right module.
14. Configure the register 0x23[0] to 1, to enable the zero-crossing detection function in ADC right channel.
Configure the register 0x23[4] to 1, to enable the zero-crossing detection function in ADC left channel.
Begin recording.

21.6.8 Disable ADC

0. Keep ADC channel work and stop recording.
1. Configure the register 0x23[0] to 0, to disable the zero-crossing detection function in ADC right channel
Configure the register 0x23[4] to 0, to disable the zero-crossing detection function in ADC left channel
2. Configure the register 0x28[1] to 0, to disable the ADC module in ADC right channel.
Configure the register 0x28[5] to 0, to disable the ADC module in ADC left channel.
3. Configure the register 0x28[2] to 0, to disable the clock module in ADC right channel.
Configure the register 0x28[6] to 0, to disable the clock module in ADC left channel.
4. Configure the register 0x25[1] to 0, to disable the ALC module in ADC right channel.
Configure the register 0x25[5] to 0, to disable the ALC module in ADC left channel.
5. Configure the register 0x25[0] to 0, to disable the MIC module in ADC right channel.
Configure the register 0x25[4] to 0, to disable the MIC module in ADC left channel
6. Configure the register 0x23[1] to 0, to disable the reference voltage buffer in ADC right channel.
Configure the register 0x23[5] to 0, to disable the reference voltage buffer in ADC left channel.
7. Configure the register 0x22[4] to 0, to disable the microphone bias voltage buffer.
8. Configure the register 0x28[0] to 0, to begin the initialization of the ADC right module.
Configure the register 0x28[4] to 0, to begin the initialization of the ADC left module.
9. Configure the register 0x28[3] to 0, to begin the initialization of the ALC right module.
Configure the register 0x28[7] to 0, to begin the initialization of the ALC left module.
10. Configure the register 0x23[2] to 0, to begin the initialization of the MIC right module.
Configure the register 0x23[6] to 0, to begin the initialization of the MIC left module.

21.6.9 Enable ALC

1.Set the Max level and Min level of the ALC need to control.
The relationship between the max_level and the signal amplitude (dB) is equal to: amplitude (dB) = 20log(max_level/16'h7fff), 16'7fff means the full scale amplitude. The register 0x47 and 0x48 control the min level of the ALC. min_level = {reg0x48, reg0x47};
The relationship between the min_level and the signal amplitude (dB) is equal to: amplitude(dB) = 20log(min_level/16'h7fff); 16'7fff means the full scale amplitude.
2. According to the sample rate to choose the configuration.
3. Enable the ALC module. Set the reg0x49 [6] to 1 to enable the ALC module.
4. Choose the ALC module to control the gain of the PGA. Set the reg0x0a [5] to 1 to choose the ALCL module to control the gain of the PGA.

5. Observe the current ALC output gain.

21.6.10 Disable ALC

1. Set the reg0x49[6] to 0 to disable the ALC module, then the ALC output gain will keep to the last value.
2. Choose to use the register to control the gain of the PGA. Set the reg0x0a[5] to 0 to choose to use the register to control the gain of the PGA, then we can use the register 0x27.

The ALC has two modes to control the gain of the PGA. If we use the register 0x49 to enable the automatically adjustment function of the ALC, the gain of the PGA will change automatically according to the amplitude of the input signal. If we don't use this function, we can use the register 0x43 to change the gain of the PGA manual. The prerequisite of these two modes is that we use the register 0x0a to choose to use the ALC module to control the gain of the PGA.

21.6.11 Inner-BIST FUNCTION

The codec has the bist function which can generate the sine or square digital wave. So the DAC path can choose the bist wave as the source, then the analog output of the DAC can output the sine or square wave without caring the digital I2S interface of the DAC. In the normal mode, the bist wave generate model is in the reset mode to save the power, you can use the register to enable this model.

The suggested way to use the Inner-Bist function lists below.

1.Configure the register 0x00 to enable the Inner-Bist wave generated model.
2.Configure the register 0x0d to choose the source of the digital input.

Note: Before you use this function, please enable the DAC firstly.

Chapter 22 Mobile Storage Host Controller

22.1 Overview

The Mobile Storage Host Controller is designed to support Secure Digital memory (SD-max version 3.01) with 1 bits or 4 bits data width, Multimedia Card(MMC-max version 4.51) with 1 bits or 4 bits or 8 bits data width.
The Host Controller is instantiated for SDMMC, SDIO and EMMC. The interface difference between these instances is shown in "Interface Description".
The Host Controller supports following features:

- Bus Interface Features:
  - Support AMBA AHB interface for master and slave
  - Supports internal DMA interface(IDMAC)
    - Supports 16/32-bit data transfers
    - Single engine used for Transmit and Receive, which are mutually exclusive
    - Dual-buffer and chained descriptor linked list
    - Each descriptor can transfer up to 4KB of data in chained mode and 8KB of data in dual-buffer mode
    - Programmable burst size for optimal host bus utilization
  - Support combined single FIFO for both transmit and receive operations
  - Support FIFO size of 256x32
  - Support FIFO over-run and under-run prevention by stopping card clock
- Card Interface Features:
  - Support Secure Digital memory protocol commands
  - Support Secure Digital I/O protocol commands
  - Support Multimedia Card protocol commands
  - Support Command Completion Signal and interrupts to host
  - Support CRC generation and error detection
  - Support programmable baud rate
  - Support power management and power switch
  - Support card detection
  - Support write protection
  - Support hardware reset
  - Support SDIO interrupts in 1-bit and 4-bit modes
  - Support 4-bit mode in SDIO3.0
  - Support SDIO suspend and resume operation
  - Support SDIO read wait
  - Support block size of 1 to 65,535 bytes
  - Support 1-bit, 4-bit and 8-bit SDR modes
  - Supports 4-bit and 8-bit DDR
  - Support boot in 1-bit, 4-bit and 8-bit SDR modes
- Clock Interface Features:
  - Support 0/90/180/270-degree phase shift operation for sample clock (cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock(cclk_in) respectively
  - Support phase tuning using delay line for sample clock(cclk_in_sample) and drive clock(cclk_in_drv) relative to function clock (cclk_in) respectively. The max number of delay element number is 256.

22.2 Block Diagram

The Host Controller consists of the following main functional blocks.

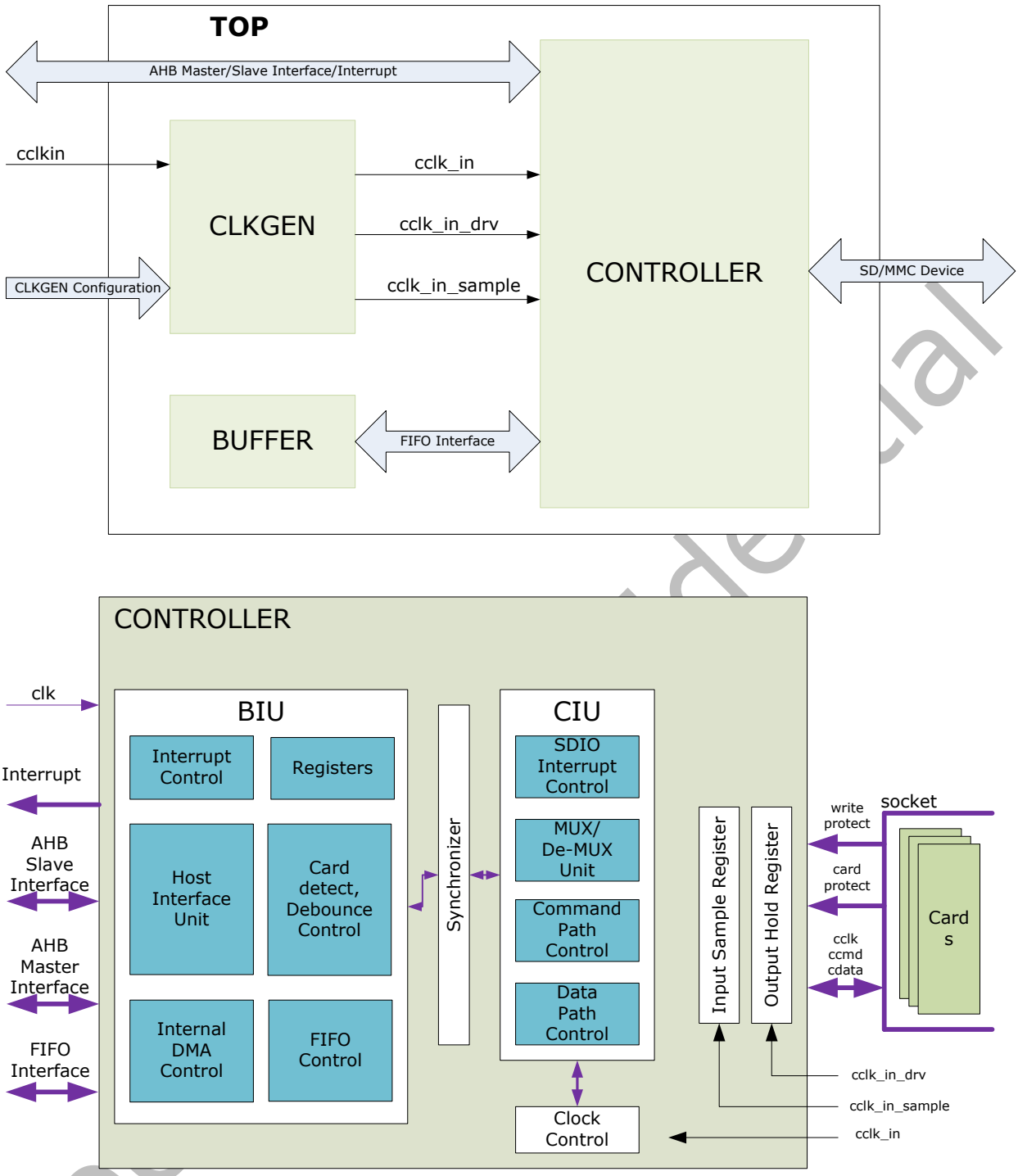


Fig. 22-1 Mobile Storage Host Control Block Diagram

- Clock Generate Unit (CLKGEN): Generates card interface clock cclk_in/ cclk_sample/cclk_drv based on cclkin and configuration information.
  - cclkin: original clock
  - cclk_in: functional clock
  - cclk_sample: sample clock
  - cclk_drv: driver clock
- Asynchronous dual-port memory (BUFFER): Use a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, and the other port is connected to the card clock.
- Bus Interface Unit (BIU): Provides AMBA AHB interfaces for register and data read/write.

Card Interface Unit (CIU): Takes care of the SD/MMC protocols and provides clock management.

22.3 Function Description

22.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- Interrupt control
- Register access
- External FIFO access
- Card detection

22.3.1.1 Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides the interface between the SD/MMC card and the host bus.

22.3.1.2 Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.

All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the SDMMC_CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.

Once a command start is issued by setting the start_bit of the SDMMC_CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- SDMMC_CMD
- SDMMC_CMDARG
- SDMMC_BYTCNT
- SDMMC_BLKSIZ
- SDMMC_CLKDIV
- SDMMC_CLKENA
- SDMMC_CLKSRC
- SDMMC_TMOUT
- SDMMC_CTYPE

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.

When the Card Interface Unit is in an idle state, it typically takes the following number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock:
3 (clk) + 3 (cclk_in)

Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the SDMMC_CMD register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.

If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

22.3.1.3 Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit.

Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register SDMMC_RINTSTS. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.
The interrupt port is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.
The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32'h0, which masks all the interrupts.
Notes:
Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal operation, ensure that you clear only the interrupt bits that you serviced.
The SDIO Interrupts, Receive FIFO Data Request (RXDR), and Transmit FIFO Data Request (TXDR) are set by level-sensitive interrupt sources. Therefore, the interrupt source should be first cleared before you can clear the interrupt bit of the Raw Interrupt register. For example, on seeing the Receive FIFO Data Request (RXDR) interrupt, the FIFO should be emptied so that the "FIFO count greater than the RX-Watermark" condition, which triggers the interrupt, becomes inactive. The rest of the interrupts are triggered by a single clock-pulse-width source.

Table 22-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	sdio_interrupt	Interrupt from SDIO card. In MMC-Ver3.3-only mode, these bits are always 0.
16	Card no-busy	If card exit busy status, the interrupt happened.
15	End Bit Error (read) / Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC received during write operation. For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt. Recommendation: Software typically need not enable this for non CE-ATA accesses; Data Transfer Over (DTO) interrupt that comes after this interrupt determines whether data transfer has correctly competed.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if DAT[0] line indicates start bit-that is, 0-and any of the other data bits do not have start bit, then this error is set. Busy Complete Interrupt when data is written to the card. This interrupt is generated after completion of busy driven by the card after the last data block is written into the card.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
11	FIFO Underrun/ Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty. If IDMAC is enabled, FIFO under-run/over-run can

Bits	Interrupt	Description
		occur due to a programming error on MSIZE and watermark values in SDMMC_FIFOTH register.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card, or does not read from FIFO during read from card before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts cclk_out and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on command signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	In Normal functioning mode: Data read timeout (DRTO) Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs. In Boot Mode: Boot Data Start (BDS) When set, indicates that Host Controller has started to receive boot data from the card. A write to this register with a value of 1 clears this interrupt.
8	Response Timeout (RTO)	In normal functioning mode: Response timeout (RTO) Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by Host Controller. In Boot Mode: Boot Ack Received (BAR) When expect_boot_ack is set, on reception of a boot acknowledge pattern—0-1-0—this interrupt is asserted. A write to this register with a value of 1 clears this interrupt.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU. Can also occur if the Write CRC status is incorrectly sampled by the Host.
6	Response CRC Error (RCRC)	Response CRC does not match with locally-generated CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than Receive-Threshold level. Recommendation: In DMA modes, this interrupt should not be enabled. In non-DMA mode: pop RX_WMark + 1 data from FIFO.
4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level. Recommendation: In DMA modes, this interrupt should not be enabled. In non-DMA mode:

Bits	Interrupt	Description

		if (pending_bytes > (FIFO_DEPTH - TX_WMark))
		push (FIFO_DEPTH - TX_WMark) data into FIFO
		else
		push pending_bytes data into FIFO

3	Data Transfer Over (DTO)	Indicates Data transfer completed. Though on detection of errors-Start Bit Error, Data CRC error, and so on, DTO may or may not be set; the application must issue CMD12, which ensures that DTO is set. Recommendation: In non-DMA mode, when data is read from card, on seeing interrupt, host should read any pending data from FIFO. In DMA mode, DMA controllers guarantee FIFO is flushed before interrupt. DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs.
1	Response Error (RE)	Error in received response set if one of following occurs: ● Transmission bit != 0 ● Command index mismatch ● End-bit != 1
0	Card-Detect (CDT)	When one or more cards inserted or removed, this interrupt occurs. Software should read card-detect register to determine current card status. Recommendation: After power-on and before enabling interrupts, software should read card detect register and store it in memory. When interrupt occurs, it should read card detect register and compare it with value stored in memory to determine which card(s) were removed/inserted. Before exiting ISR, software should update memory with new card-detect value.

22.3.1.4 FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host interface and the card controller unit. When FIFO overrun and under-run conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock(clk), and the second port is connected to the card clock(cclk_in).

22.3.1.5 Card Detection Unit

The register unit has registers that control the power. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounce associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

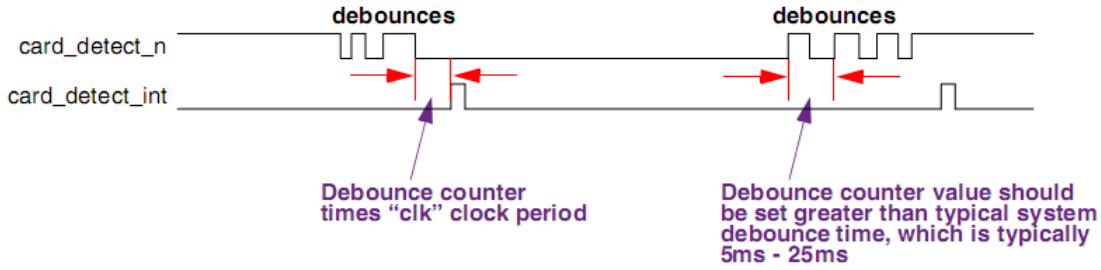


Fig. 22-2 SD/MMC Card-Detect Signal

22.3.2 Card Interface Unit

The Card Interface Unit (CIU) interfaces with the Bus Interface Unit (BIU) and the external devices. The host writes command parameters to the BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The Host Controller accordingly controls the command and data path.

The following software restrictions should be met for proper CIU operation:

- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the stop_abort_cmd bit in the SDMMC_CMD register so that the Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the SDMMC_RINTSTS register, the Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.
- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:

- Command path
- Data path
- SDIO interrupt control
- Clock control
- Error detection

22.3.2.1 Command Path

The command path performs the following functions:

- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU
- Drives the P-bit on command line

A new command is issued to the Host Controller by programming the BIU registers and setting the start_cmd bit in the SDMMC_CMD register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC device. The command path loads this new command (command, command argument, timeout) and sends acknowledge to the BIU by asserting cmd_taken.

Once the new command is loaded, the command path state machine sends a command to the device bus-including the internally generated CRC7-and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.

Load Command Parameters

One of the following commands or responses is loaded in the command path:

- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in the SDMMC_CMD register.

- Internally generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a "send irq response" request is signaled by the BIU, then the send_irq_response bit is set in the SDMMC_CTRL register.

Loading a new command from the BIU in the command path depends on the following SDMMC_CMD register bit settings:

- update_clock_registers_only – If this bit is set in the SDMMC_CMD register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command, command argument, and timeout registers; it then starts processing the new command.
- wait_prvdata_complete – If this bit is set, the command path loads the new command under one of the following conditions:
  - Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (byte_count = 0).
  - After completion of the current data transfer, if a predefined data transfer is in progress.

Send Command and Receive Response

Once a new command is loaded in the command path, update_clock_registers_only bit is unset – the command path state machine sends out a command on the device bus; the command path state machine is illustrated in following figure.

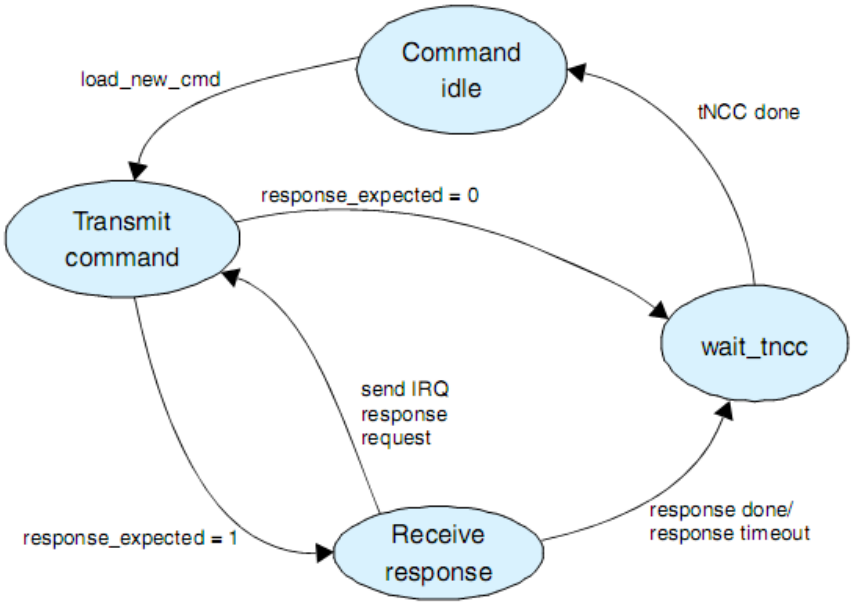


Fig. 22-3 Host Controller Command Path State Machine

The command path state machine performs the following functions, according to SDMMC_CMD register bit values:

- send_initialization – Initialization sequence of 80 clocks is sent before sending the command.
- response_expected – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
- response_length – If this bit is set, a 136-bit response is received; if it is not set, a 48-bit response is received.
- check_response_crc – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the response CRC error is signaled to the BIU; that is, the response CRC error bit is set in

the Raw Interrupt Status register SDMMC_RINTSTS.

Send Response to BIU

If the response_expected bit is set in the SDMMC_CMD register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set.

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command
- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved-that is, 111111.

Polling Command Completion Signal

The device generates the Command Completion Signal in order to notify the host controller of the normal command completion or command termination.

Command Completion Signal Detection and Interrupt to Host Processor

If the ccs_expected bit is set in the SDMMC_CMD register, the Command Completion Signal (CCS) from the device is indicated by setting the Data Transfer Over (DTO) bit in the SDMMC_RINTSTS register. The Host Controller generates a Data Transfer Over (DTO) interrupt if this interrupt is not masked.

Command Completion Signal Timeout

If the command expects a CCS from the device—if the ccs_expected bit is set in the SDMMC_CMD register—the command state machine waits for the CCS and remains in a wait_CCSS state. If the device fails to send out the CCS, the host software should implement a timeout mechanism to free the command and data path. The host controller does not implement a hardware timer; it is the responsibility of the host software to maintain a software timer.

In the event of a CCS timeout, the host should issue a CCSD by setting the send_ccsd bit in the CTRL register. The host controller command state machine sends the CCSD to the device and exits to an idle state. After sending the CCSD, the host should also send a CMD12 to the device in order to abort the outstanding command.

Send Command Completion Signal Disable

If the send_ccsd bit is set in the SDMMC_CTRL register, the host sends a Command Completion Signal Disable (CCSD) pattern on the CMD line. The host can send the CCSD while waiting for the CCS or after a CCS timeout happens.

After sending the CCSD pattern, the host sets the Command Done (CD) bit in SDMMC_RINTSTS and also generates an interrupt to the host if the Command Done interrupt is not masked.

22.3.2.2 Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the SDMMC_CMD register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in following figure, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the SDMMC_CMD register, the data transmit

state machine puts data on the card data bus in a stream or in block(s).

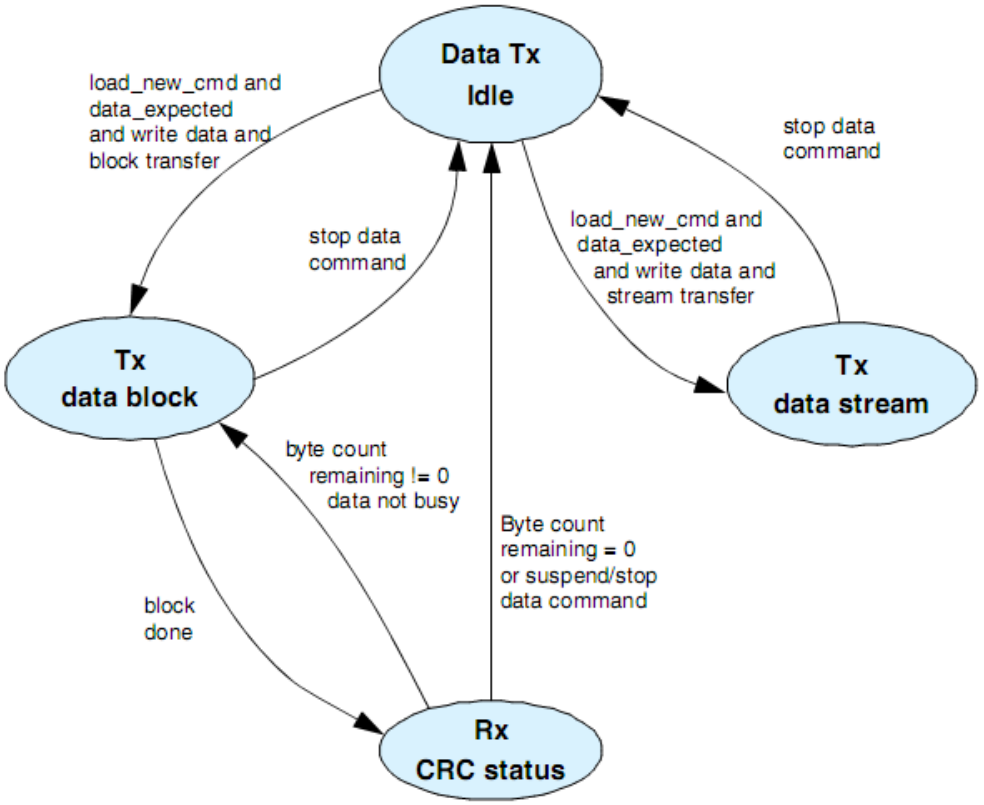


Fig. 22-4 Host Controller Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the SDMMC_CMD register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the SDMMC_CMD register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-

CRC bit in the SDMMC_RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16. If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the SDMMC_RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the SDMMC_RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in following figure, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the SDMMC_CMD register, the data-receive state machine gets data from the card data bus in a stream or block(s).

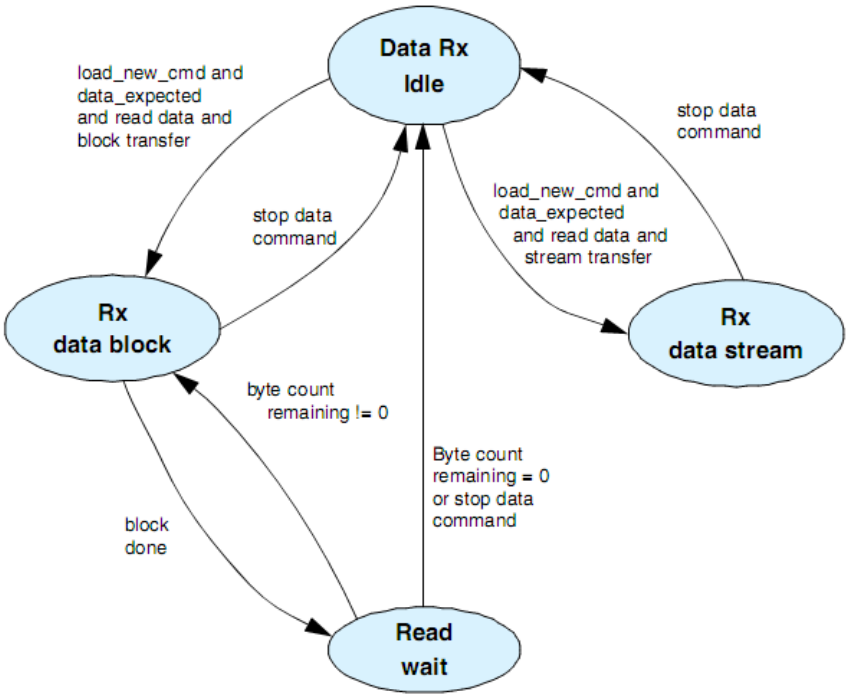


Fig. 22-5 Host Controller Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the SDMMC_CMD register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full.

An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the SDMMC_CMD register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16. If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the SDMMC_CMD register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the SDMMC_CTYPE register bit for the selected card – indicated by the card_num value in the SDMMC_CMD register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals

a data transfer to the BIU.
If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.
Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.
If the send_auto_stop bit is set in the SDMMC_CMD register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.
If the requested block size for data transfers to cards is less than 4, 16, or 32 bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.
If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the SDMMC_CMD register.
The software should set the send_auto_stop bit according to details listed in following table.

Table 22-2 Auto-Stop Generation

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
MMC	Stream read	0	No	Open-ended stream
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stream
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block	0	No	Open-ended multiple block

Card type	Transfer type	Byte Count	send_auto_stop bit set	Comments
	read			
SDIO	Multiple-block read	>0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	>0	No	Pre-defined multiple block

*①:The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2, 3, …), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.*

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32 (8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.
- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.

Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC device until the auto-stop is sent by the Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the Host Controller does not generate an auto-stop command.

22.3.2.3 Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Following table lists the commands and register programming requirements for them.

Table 22-3 Non-data Transfer Commands and Requirements

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
SDMMC_CMD register programming						
cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
response_expect	1	1	1	1	1	1

Base Address [12:8]	CMD 27	CMD 30	CMD 42	ACMD 13	ACMD 22	ACMD 51
rResponse_length	0	0	0	0	0	0
check_response_crc	1	1	1	1	1	1
data_expected	1	1	1	1	1	1
read/write	1	0	1	0	0	0
transfer_mode	0	0	0	0	0	0
send_auto_stop	0	0	0	0	0	0
wait_prevdata_complete	0	0	0	0	0	0
stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	stuff bits	32-bit write protect data address	stuff bits	stuff bits	stuff bits	stuff bits
Block Size register programming						
	16	4	Num_bytes①	64	4	8
Byte Count register programming						
	16	4	Num_bytes①	64	4	8

①: Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)

22.3.2.4 SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

- Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands. According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.

Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

22.3.2.5 Clock Control

The clock control block provides different clock frequencies required for SD/MMC cards. The cclk_in signal is the source clock (cclk_in >= card max operating frequency) for clock divider of the clock control block. This source clock (cclk_in) is used to generate different card clock frequencies (cclk_out). The card clock can have different clock frequencies, since the card can be a low-speed card or a full-speed card. The Host Controller provides one clock signal (cclk_out).

The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. A value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2.
- Clock Control register – cclk_out can be enabled or disabled for each card under the following conditions:
  - clk_enable – cclk_out for a card is enabled if the clk_enable bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
  - Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the cclk_out is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, cclk_out is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions. It is used by the command and data path to qualify cclk_in for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.

Under the following conditions, the card clock is stopped or disabled, along with the active clk_en, for the selected card:

- Clock can be disabled by writing to Clock Enable register (clk_en bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.

FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

22.3.2.6 Error Detection

- Response
  - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
  - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.
  - Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
  - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
    - Signals no CRC status error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to the BIU
  - Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
  - Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.
- Data receive
  - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data path does the following:
    - Signals data-timeout error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to BIU

- Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
- Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.
- Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.

Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in SDMMC_RINTSTS register) and the data path continues to wait for the FIFO to start to empty.

22.3.3 Internal Direct Memory Access Controller (IDMAC)

The Internal Direct Memory Access Controller (IDMAC) has a Control and Status Register (CSR) and a single Transmit/Receive engine, which transfers data from host memory to the device port and vice versa. The controller utilizes a descriptor to efficiently move data from source to destination with minimal Host CPU intervention. You can program the controller to interrupt the Host CPU in situations such as data Transmit and Receive transfer completion from the card, as well as other normal or error conditions.

The IDMAC and the Host driver communicate through a single data structure. CSR addresses 0x80 to 0x98 are reserved for host programming.

The IDMAC transfers the data received from the card to the Data Buffer in the Host memory, and it transfers Transmit data from the Data Buffer in the Host memory to the FIFO. Descriptors that reside in the Host memory act as pointers to these buffers.

A data buffer resides in physical memory space of the Host and consists of complete data or partial data. Buffers contain only data, while buffer status is maintained in the descriptor. Data chaining refers to data that spans multiple data buffers. However, a single descriptor cannot span multiple data.

A single descriptor is used for both reception and transmission. The base address of the list is written into Descriptor List Base Address Register (SDMMC_DBADDR @0x88). A descriptor list is forward linked. The Last Descriptor can point back to the first entry in order to create a ring structure. The descriptor list resides in the physical memory address space of the Host. Each descriptor can point to a maximum of two data buffers.

22.3.3.1 IDMAC CSR Access

When an IDMAC is introduced, an additional CSR space resides in the IDMAC that controls the IDMAC functionality. The host accesses the new CSR space in addition to the existing control register set in the BIU. The IDMAC CSR primarily contains descriptor information. For a write operation to the CSR, the respective CSR logic of the IDMAC and BIU decodes the address before accepting. For a read operation from the CSR, the appropriate CSR read path is enabled.

You can enable or disable the IDMAC operation by programming bit[25] in the SDMMC_CTRL register of the BIU. This allows the data transfer by accessing the slave interface on the AMBA bus if the IDMAC is present but disabled. When IDMAC is enabled, the FIFO cannot be accessed through the slave interface.

22.3.3.2 Descriptors

- Descriptor structures

The IDMAC uses these types of descriptor structures:

- Dual-Buffer Structure – The distance between two descriptors is determined by the Skip Length value programmed in the Descriptor Skip Length (DSL) field of the Bus Mode Register (SDMMC_BMOD @0x80).

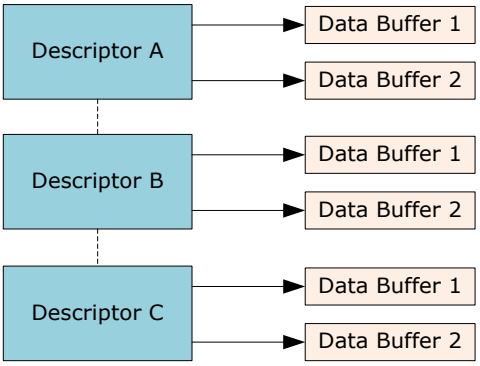


Fig. 22-6 Dual-Buffer Descriptor Structure

- Chain Structure – Each descriptor points to a unique buffer and the next descriptor.

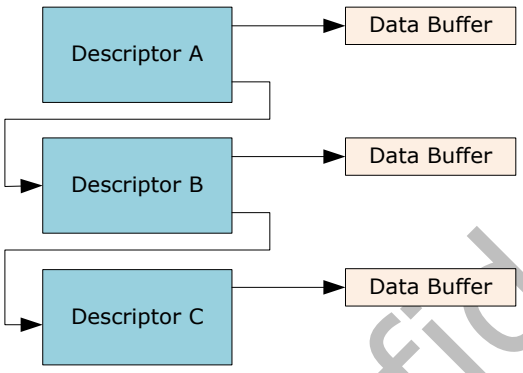


Fig. 22-7 Chain Descriptor Structure

- Descriptor formats

Following figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit AHB data buses. Each descriptor contains 16 bytes of control and status information. DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, DES3 to denote [127:96] bits.

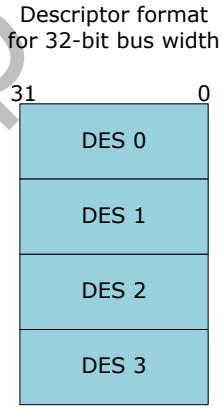


Fig. 22-8 Descriptor Formats for 32-bit AHB Address Bus Width

- The DES0 element in the IDMAC contains control and status information.

Table 22-4 Bits in IDMAC DES0 Element

Bit	Name	Description
31	OWN	When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the Host. The IDMAC clears this bit when it completes the data transfer.
30	Card Error Summary (CES)	These error bits indicate the status of the transaction to or from the card. These bits are also present in SDMMC_RINTSTS Indicates the logical OR of the following bits: ● EBE: End Bit Error

Bit	Name	Description
		● RTO: Response Time out ● RCRC: Response CRC ● SBE: Start Bit Error ● DRTO: Data Read Timeout ● DCRC: Data CRC for Receive ● RE: Response Error
29:6	Reserved	-
5	End of Ring (ER)	When set, this bit indicates that the descriptor list reached its final descriptor. The IDMAC returns to the base address of the list, creating a Descriptor Ring. This is meaningful for only a dual-buffer descriptor structure.
4	Second Address Chained (CH)	When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When this bit is set, BS2 (DES1[25:13]) should be all zeros.
3	First Descriptor (FS)	When set, this bit indicates that this descriptor contains the first buffer of the data. If the size of the first buffer is 0, next Descriptor contains the beginning of the data.
2	Last Descriptor (LD)	This bit is associated with the last block of a DMA transfer. When set, the bit indicates that the buffers pointed to by this descriptor are the last buffers of the data. After this descriptor is completed, the remaining byte count is 0. In other words, after the descriptor with the LD bit set is completed, the remaining byte count should be 0.
1	Disable Interrupt on Completion (DIC)	When set, this bit will prevent the setting of the TI/RI bit of the IDMAC Status Register (IDSTS) for the data that ends in the buffer pointed to by this descriptor.
0	Reserved	-

■ The DES1 element contains the buffer size.

Table 22-5 Bits in IDMAC DES1 Element

Bit	Name	Description
31:26	Reserved	-
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

■ The DES2 element contains the address pointer to the data buffer.

Table 22-6 Bits in IDMAC DES2 Element

Bit	Name	Description
31:26	Reserved	
25:13	Buffer 2 Size (BS2)	These bits indicate the second data buffer byte size. The buffer size must be a multiple of 2, 4, or 8, depending upon the bus widths—16, 32, and 64 respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior

Bit	Name	Description
		is undefined. If this field is 0, the DMA ignores this buffer and proceeds to the next buffer in case of a dual-buffer structure. This field is not valid for chain structure; that is, if DES0[4] is set.
12:0	Buffer 1 Size (BS1)	Indicates the data buffer byte size, which must be a multiple of 2, 4, or 8 bytes, depending upon the bus widths—16, 32, and 64, respectively. In the case where the buffer size is not a multiple of 2, 4, or 8, the resulting behavior is undefined. This field should not be zero. Note: If there is only one buffer to be programmed, you need to use only the Buffer 1, and not Buffer 2.

- The DES3 element contains the address pointer to the next descriptor if the present descriptor is not the last descriptor in a chained descriptor structure or the second buffer address for a dual-buffer structure.

Table 22-7 Bits in IDMAC DES3 Element

Bit	Name	Description
31:0	Buffer Address Pointer 2/ Next Descriptor Address (BAP2)	These bits indicate the physical address of the second buffer when the dual-buffer structure is used. If the Second Address Chained (DES0[4]) bit is set, then this address contains the pointer to the physical memory where the Next Descriptor is present. If this is not the last descriptor, then the Next Descriptor address pointer must be bus-width aligned.

22.3.3.3 Initialization

IDMAC initialization occurs as follows:

1) Write to IDMAC Bus Mode Register—SDMMC_BMOD to set Host bus access parameters.
2) Write to IDMAC Interrupt Enable Register—SDMMC_IDINTEN to mask unnecessary interrupt causes.
3) The software driver creates either the Transmit or the Receive descriptor list. Then it writes to IDMAC Descriptor List Base Address Register (SDMMC_DBADDR), providing the IDMAC with the starting address of the list.
4) The IDMAC engine attempts to acquire descriptors from the descriptor lists.

- Host Bus Burst Access

The IDMAC attempts to execute fixed-length burst transfers on the AHB Master interface if configured using the FB bit of the IDMAC Bus Mode register. The maximum burst length is indicated and limited by the PBL field. The descriptors are always accessed in the maximum possible burst-size for the 16-bytes to be read— 16*8/bus-width.
The IDMAC initiates a data transfer only when sufficient space to accommodate the configured burst is available in the FIFO or the number of bytes to the end of data, when less than the configured burst-length.
The IDMAC indicates the start address and the number of transfers required to the AHB Master Interface. When the AHB Interface is configured for fixed-length bursts, then it transfers data using the best combination of INCR4/8/16 and SINGLE transactions. Otherwise, in no fixed-length bursts, it transfers data using INCR (undefined length) and SINGLE transactions.

- Host Data Buffer Alignment

The Transmit and Receive data buffers in host memory must be aligned, depending on the data width.

- Buffer Size Calculations

The driver knows the amount of data to transmit or receive. For transmitting to the card, the IDMAC transfers the exact number of bytes to the FIFO, indicated by the buffer size field of DES1.
If a descriptor is not marked as last-LS bit of DES0-then the corresponding buffer(s) of the descriptor are full, and the amount of valid data in a buffer is accurately indicated by its buffer size field. If a descriptor is marked as last, then the buffer cannot be full, as indicated by the buffer size in DES1. The driver is aware of the number of locations that are

valid in this case.

- Transmission

IDMAC transmission occurs as follows:

1) The Host sets up the elements (DES0-DES3) for transmission and sets the OWN bit (DES0[31]). The Host also prepares the data buffer.
2) The Host programs the write data command in the SDMMC_CMD register in BIU.
3) The Host will also program the required transmit threshold level (TX_WMark field in SDMMC_FIFOTH register).
4) The IDMAC determines that a write data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the IDMAC enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed transmit threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB Master Interface.
8) The IDMAC fetches the Transmit data from the data buffer in the Host memory and transfers to the FIFO for transmission to card.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data transmission is complete, status information is updated in SDMMC_IDSTS register by setting Transmit Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Reception

IDMAC reception occurs as follows:

1) The Host sets up the element (DES0-DES3) for reception, sets the OWN (DES0[31]).
2) The Host programs the read data command in the SDMMC_CMD register in BIU.
3) The Host will program the required receive threshold level (RX_WMark field in FIFOTH register).
4) The IDMAC determines that a read data transfer needs to be done as a consequence of step 2.
5) The IDMAC engine fetches the descriptor and checks the OWN bit. If the OWN bit is not set, it means that the host owns the descriptor. In this case the DMA enters suspend state and asserts the Descriptor Unable interrupt in the SDMMC_IDSTS register. In such a case, the host needs to release the IDMAC by writing any value to the poll demand register.
6) It will then wait for Command Done (CD) bit and no errors from BIU which indicates that a transfer can be done.
7) The IDMAC engine will now wait for a DMA interface request from BIU. This request will be generated based on the programmed receive threshold value. For the last bytes of data which can't be accessed using a burst, SINGLE transfers are performed on AHB.
8) The IDMAC fetches the data from the FIFO and transfer to Host memory.
9) When data spans across multiple descriptors, the IDMAC will fetch the next descriptor and continue with its operation with the next descriptor. The Last Descriptor bit in the descriptor indicates whether the data spans multiple descriptors or not.
10) When data reception is complete, status information is updated in SDMMC_IDSTS register by setting Receive Interrupt, if enabled. Also, the OWN bit is cleared by the IDMAC by performing a write transaction to DES0.

- Interrupts

Interrupts can be generated as a result of various events. SDMMC_IDSTS register contains all the bits that might cause an interrupt. SDMMC_IDINTEN register contains an Enable bit

for each of the events that can cause an interrupt.
There are two groups of summary interrupts-Normal and Abnormal-as outlined in SDMMC_IDSTS register. Interrupts are cleared by writing a 1 to the corresponding bit position. When all the enabled interrupts within a group are cleared, the corresponding summary bit is cleared. When both the summary bits are cleared, the interrupt signal dmac_intr_o is de-asserted.
Interrupts are not queued and if the interrupt event occurs before the driver has responded to it, no additional interrupts are generated. For example, Receive Interrupt—SDMMC_IDSTS[1] indicates that one or more data was transferred to the Host buffer.
An interrupt is generated only once for simultaneous, multiple events. The driver must scan SDMMC_IDSTS register for the interrupt cause.

22.3.3.4 Variable Delay/Clock Generation Unit

Variable delay mechanism for the cclk_in_drv is useful in order to meet a range of hold-time requirements across modes. Variable delay mechanism for the cclk_in_sample is mandatory and is required to achieve the correct sampling point for data.
The Clock Generation Unit (CLKGEN) includes Phase Shift Unit and Delay Line Unit.
The Phase Shift Unit can shift cclk_in_sample/cclk_in_drv by 0/90/180/270-degree relative to cclk_in. The Delay Line Unit can shift cclk_in_sample/cclk_in_drv step by step in the unit of delay element. The delay value range is 25ps~56ps for every delay element; the max delay element number is 256.
The architecture is as follows.

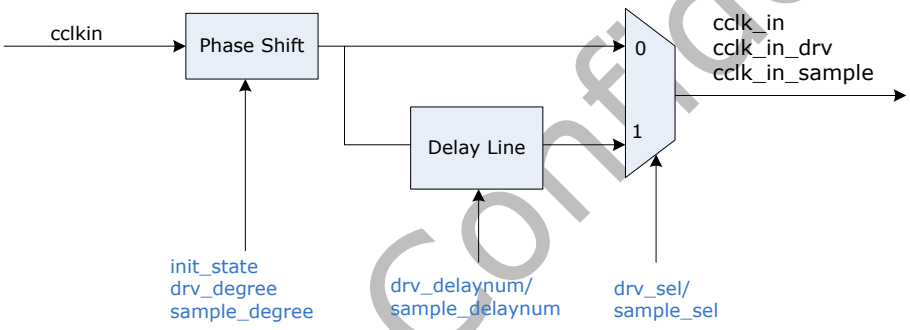


Fig. 22-9 Clock Generation Unit

22.4 Register Description

22.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x0000	W	0x00000000	Control register
SDMMC_PWREN	0x0004	W	0x00000000	Power enable register
SDMMC_CLKDIV	0x0008	W	0x00000000	Clock divider register
SDMMC_CLKSRC	0x000C	W	0x00000000	SD clock source register
SDMMC_CLKENA	0x0010	W	0x00000000	Clock enable register
SDMMC_TMOUT	0x0014	W	0xFFFFFF40	Timeout register
SDMMC_CTYPE	0x0018	W	0x00000000	Card type register
SDMMC_BLKSIZ	0x001C	W	0x00000200	Block size register
SDMMC_BYTCNT	0x0020	W	0x00000200	Byte count register
SDMMC_INTMASK	0x0024	W	0x00000000	Interrupt mask register
SDMMC_CMDARG	0x0028	W	0x00000000	Command argument register
SDMMC_CMD	0x002C	W	0x20000000	Command register
SDMMC_RESP0	0x0030	W	0x00000000	Response register 0
SDMMC_RESP1	0x0034	W	0x00000000	Response register 1
SDMMC_RESP2	0x0038	W	0x00000000	Response register 2
SDMMC_RESP3	0x003C	W	0x00000000	Response register 3
SDMMC_MINTSTS	0x0040	W	0x00000000	Masked interrupt status register
SDMMC_RINTSTS	0x0044	W	0x00000000	Raw interrupt status register
SDMMC_STATUS	0x0048	W	0x00000106	Status register

Name	Offset	Size	Reset Value	Description
SDMMC_FIFOTH	0x004C	W	0x00FF0000	FIFO threshold register
SDMMC_CDETECT	0x0050	W	0x00000001	Card detect register
SDMMC_WRTPRT	0x0054	W	0x00000000	Write protect register
SDMMC_TCBCNT	0x005C	W	0x00000000	Transferred card byte count register
SDMMC_TBBCNT	0x0060	W	0x00000000	Transferred host to FIFO byte count register
SDMMC_DEBNCE	0x0064	W	0x00FFFFFF	Debounce count register
SDMMC_USRID	0x0068	W	0x00000000	User ID register
SDMMC_VERID	0x006C	W	0x5342270A	Version ID register
SDMMC_HCON	0x0070	W	0x04C434C1	Hardware configuration register
SDMMC_UHSREG	0x0074	W	0x00000000	UHS-1 control register
SDMMC_RSTN	0x0078	W	0x00000001	Hardware reset register
SDMMC_BMOD	0x0080	W	0x00000000	Bus mode register
SDMMC_PLDMND	0x0084	W	0x00000000	Poll demand register
SDMMC_DBADDR	0x0088	W	0x00000000	Descriptor list base address register
SDMMC_IDSTS	0x008C	W	0x00000000	Internal DMAC status register
SDMMC_IDINTEN	0x0090	W	0x00000000	Internal DMAC interrupt enable register
SDMMC_DSCADDR	0x0094	W	0x00000000	Current host descriptor address register
SDMMC_BUFADDR	0x0098	W	0x00000000	Current buffer descriptor address register
SDMMC_CARDTHRCTL	0x0100	W	0x00000000	Card threshold control register
SDMMC_BACKEND_POWER	0x0104	W	0x00000000	Back-end power register
SDMMC_EMMCDDR_REG	0x010C	W	0x00000000	eMMC4.5 DDR start bit detection control register
SDMMC_RDYINT_GEN	0x0120	W	0x00FF0000	Card ready interrupt generation control register
SDMMC_FIFO_BASE	0x0200	W	0x00000000	FIFO base address register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

22.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25	RW	0x0	use_internal_dmac Present only for the Internal DMAC configuration; else, it is reserved. 1'b0: The host performs data transfers through the slave interface 1'b1: Internal DMAC used for data transfer
24:12	RO	0x0000	reserved
11	RW	0x0	ceata_device_interrupt_status 1'b0: Interrupts not enabled in CE-ATA device 1'b1: Interrupts are enabled in CE-ATA device Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled. If the host enables CE-ATA device interrupt, then software should set this bit.

Bit	Attr	Reset Value	Description
10	RW	0x0	send_auto_stop_ccsd 1'b0: Clear bit if Host Controller does not reset the bit 1'b1: Send internally generated STOP after sending CCSD to CE-ATA device NOTE: Always set send_auto_stop_ccsd and send_ccsd bits together send_auto_stop_ccsd should not be set independent of send_ccsd. When set, the Host Controller automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in SDMMC_RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, the Host Controller automatically clears send_auto_stop_ccsd bit.
9	RW	0x0	send_ccsd 1'b0: Clear bit if Host Controller does not reset the bit 1'b1: Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, the Host Controller sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, the Host Controller automatically clears send_ccsd bit. It also sets Command Done (CD) bit in SDMMC_RINTSTS register and generates interrupt to host if Command Done interrupt is not masked. NOTE: Once send_ccsd bit is set, it takes two card clock cycles to drive the CCSD on the CMD line. Due to this, during the boundary conditions it may happen that CCSD is sent to the CE-ATA device, even if the device signalled CCS.
8	RW	0x0	abort_read_data 1'b0: No change 1'b1: After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence.
7	RW	0x0	send_irq_response 1'b0: No change 1'b1: Send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, software issues CMD40, and the Host Controller waits for interrupt response from MMC card. In meantime, if software wants the Controller to exit waiting for interrupt state, it can set this bit, at which time the Host Controller command state-machine sends CMD40 response on bus and returns to idle state.
6	RW	0x0	read_wait 1'b0: Clear read wait 1'b1: Assert read wait For sending read-wait to SDIO cards.

Bit	Attr	Reset Value	Description
5	RW	0x0	dma_enable 1'b0: Disable DMA transfer mode 1'b1: Enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside the controller to prioritize simultaneous host/DMA access.
4	RW	0x0	int_enable Global interrupt enable/disable bit 1'b0: Disable interrupts 1'b1: Enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3	RO	0x0	reserved
2	W1C	0x0	dma_reset 1'b0: No change 1'b1: Reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.
1	W1C	0x0	fifo_reset 1'b0: No change 1'b1: Reset to data FIFO to reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation.
0	W1C	0x0	controller_reset 1'b0: No change 1'b1: Reset Host Controller To reset Host Controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: a. BIU/CIU interface b. CIU and state machines c. abort_read_data, send_irq_response, and read_wait bits of SDMMC_CTRL register d. start_cmd bit of SDMMC_CMD register Does not affect any registers or DMA interface, or FIFO or controller interrupts.

SDMMC_PWREN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 1'b0: Power off 1'b1: Power on Bit values output to card_power_en port.

SDMMC_CLKDIV

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	clk_divider0 Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, and so on. The recommended value is 0 or 1.

SDMMC_CLKSRC

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1:0	RW	0x0	clk_source Clock divider source. 2'b00: Clock divider 0 The cclk_out is always from clock divider 0, and this register is not implemented.

SDMMC_CLKENA

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	cclk_low_power Low-power control for SD card clock and MMC card clock supported. 1'b0: Non-low-power mode 1'b1: Low-power mode. Stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped).
15:1	RO	0x0000	reserved
0	RW	0x0	cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 1'b0: Clock disabled 1'b1: Clock enabled

SDMMC_TMOUT

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RW	0xffffff	data_timeout Value for card data read timeout; same value also used for data starvation by host timeout. Value is in number of card output clock. Note: The software timer should be used if the timeout value is in the order of 100 ms. In this case, read data timeout interrupt needs to be disabled.
7:0	RW	0x40	response_timeout Response timeout value. Value is in number of card output clock cclk_out.

SDMMC_CTYPE

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	card_width_8 Indicates if card is 8-bit. 1'b0: Non 8-bit mode 1'b1: 8-bit mode
15:1	RO	0x0000	reserved
0	RW	0x0	card_width Indicates if card is 1-bit or 4-bit. 1'b0: 1-bit mode 1'b1: 4-bit mode

SDMMC_BLKSIZ

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	RW	0x0200	block_size Block size

SDMMC_BYTCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000200	byte_count Number of bytes to be transferred; should be integer multiple of block size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

SDMMC_INTMASK

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RW	0x0	sdio_int_mask 1'b0: SDIO interrupt not masked 1'b1: SDIO interrupt masked
23:17	RO	0x00	reserved
16	RW	0x0	data_nobusy_int_mask 1'b0: Data no busy interrupt not masked 1'b1: Data no busy interrupt masked

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_CMDARG

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card

SDMMC_CMD

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD/MMC cards, Command Done bit is set in raw interrupt register.
30	RO	0x0	reserved
29	RW	0x1	use_hold_reg Use hold register. 1'b0: CMD and DATA sent to card bypassing hold register 1'b1: CMD and DATA sent to card through the hold register
28	RW	0x0	volt_switch Voltage switch bit. 1'b0: No voltage switching 1'b1: Voltage switching enabled; must be set for CMD11 only.
27	RW	0x0	boot_mode Boot mode selection. 1'b0: Mandatory boot operation 1'b1: Alternate boot operation
26	RW	0x0	disable_boot Disable boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do not set disable_boot and enable_boot together.

Bit	Attr	Reset Value	Description
25	RW	0x0	expect_boot_ack Expect boot acknowledge. When software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.
24	RW	0x0	enable_boot Enable boot. This bit should be set only for mandatory boot mode. When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do not set disable_boot and enable_boot together.
23	RW	0x0	ccs_expected 1'b0: Interrupts are not enabled in CE-ATA device or command does not expect CCS from device 1'b1: Interrupts are enabled in CE-ATA device and RW_BLK command expects command completion signal from CE-ATA device If the command expects command completion signal (CCS) from the CE-ATA device, the software should set this control bit. The Host Controller sets data transfer over (DTO) bit in SDMMC_RINTSTS register and generates interrupt to host if data transfer over interrupt is not masked.
22	RW	0x0	read_ceata_device 1'b0: Host is not performing read access towards CE-ATA device 1'b1: Host is performing read access towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. The Host Controller should not indicate read data timeout while waiting for data from CE-ATA device.
21	RW	0x0	update_clock_regs_only 1'b0: Normal command sequence 1'b1: Do not send commands, just update clock register value into card clock domain. Following register values transferred into card clock domain: SDMMC_CLKDIV, SDMMC_CLRSRC, SDMMC_CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_regs_only = 0, following control registers are transferred from BIU to CIU: SDMMC_CMD, SDMMC_CMDARG, SDMMC_TMOUT, SDMMC_CTYPE, SDMMC_BLKSIZ, SDMMC_BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC_CEATA cards.
20:16	RO	0x00	reserved

Bit	Attr	Reset Value	Description
15	RW	0x0	send_initialization 1'b0: Do not send initialization sequence (80 clocks of 1) before sending this command 1'b1: Send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory).
14	RW	0x0	stop_abort_cmd 1'b0: Neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1'b1: Stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with SDMMC_CMD[26]=disable_boot.
13	RW	0x0	wait_prvdata_complete 1'b0: Send command at once, even if previous data transfer has not completed 1'b1: Wait for previous data transfer completion before sending command The wait_prvdata_complete=0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.
12	RW	0x0	send_auto_stop 1'b0: No stop command sent at end of data transfer 1'b1: Send stop command at end of data transfer When set, the Host Controller sends stop command to card at end of data transfer. a. When send_auto_stop bit should be set, since some data transfers do not need explicit stop commands b. Open-ended transfers that software should explicitly send to stop command Additionally, when "resume" is sent to resume-suspended memory access of SD-Combo card, bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card.
11	RW	0x0	transfer_mode 1'b0: Block data transfer command 1'b1: Stream data transfer command Don't care if no data expected.
10	RW	0x0	wr 1'b0: Read from card 1'b1: Write to card Don't care if no data expected from card.
9	RW	0x0	data_expected 1'b0: No data transfer expected (read/write) 1'b1: Data transfer expected (read/write)

Bit	Attr	Reset Value	Description
8	RW	0x0	check_response_crc 1'b0: Do not check response CRC 1'b1: Check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller.
7	RW	0x0	response_length 1'b0: Short response expected from card 1'b1: Long response expected from card
6	RW	0x0	response_expect 1'b0: No response expected from card 1'b1: Response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response1 Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short" for them.

SDMMC_RESP2

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response

SDMMC_RESP3

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response3 Bit[127:96] of long response

SDMMC_MINTSTS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RO	0x0	sdio_interrupt SDIO interrupt status when sdio_int_mask is set
23:17	RO	0x00	reserved
16	RW	0x0	data_nobusy_int_status Data no busy interrupt status when data_nobusy_int_mask is set

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_RINTSTS

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	R/WSC	0x0	sdio_interrupt Raw SDIO interrupt status. Write value of 1 clears this bit, and value of 0 leaves bit intact.
23:17	RO	0x00	reserved
16	R/WSC	0x0	data_nobusy_int_status Raw data no busy interrupt status. Write value of 1 clears this bit, and value of 0 leaves bit intact.
15:0	R/WSC	0x0000	int_status Raw interrupt status. Writes to bits clear status bit. Write value of 1 clears status bit, and value of 0 leaves bit intact. bit 15: End-bit error (read)/Write no CRC (EBE) bit 14: Auto command done (ACD) bit 13: Start-bit error (SBE) bit 12: Hardware locked write error (HLE) bit 11: FIFO underrun/overrun error (FRUN) bit 10: Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9: Data read timeout (DRTO) bit 8: Response timeout (RTO) bit 7: Data CRC error (DCRC) bit 6: Response CRC error (RCRC) bit 5: Receive FIFO data request (RXDR) bit 4: Transmit FIFO data request (TXDR) bit 3: Data transfer over (DTO) bit 2: Command done (CD) bit 1: Response error (RE) bit 0: Card detect (CD)

SDMMC_STATUS

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	dma_req DMA request signal state
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core.
10	RO	0x0	data_state_mc_busy Data transmit or receive state-machine is busy
9	RO	0x0	data_busy Inverted version of raw selected card_data[0]. 1'b0: Card data not busy 1'b1: Card data busy
8	RO	0x1	data_3_status Raw selected card_data[3]; checks whether card is present. 1'b0: Card not present 1'b1: Card present
7:4	RO	0x0	command_fsm_states Command FSM states: 4'h0: Idle 4'h1: Send init sequence 4'h2: Tx cmd start bit 4'h3: Tx cmd tx bit 4'h4: Tx cmd index + arg 4'h5: Tx cmd crc7 4'h6: Tx cmd end bit 4'h7: Tx resp start bit 4'h8: Rx resp IRQ response 4'h9: Rx resp tx bit 4'ha: Rx resp cmd idx 4'hb: Rx resp data 4'hc: Rx resp crc7 4'hd: Rx resp end bit 4'he: Cmd path wait NCC 4'hf: Wait; CMD-to-response turnaround The command FSM state is represented using 19 bits. The SDMMC_STATUS register[7:4] has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the SDMMC_STATUS[7:4] register. The three states that are not represented in the SDMMC_STATUS register[7:4] are: Bit 16: Wait for CCS Bit 17: Send CCSD Bit 18: Boot Mode Due to this, while command FSM is in "Wait for CCS state" or "Send CCSD" or "Boot Mode", the SDMMC_STATUS register indicates status as 0 for the bit field [7:4].
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer.

Bit	Attr	Reset Value	Description
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer.

SDMMC_FIFOTH

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RW	0x0	dma_mutiple_transaction_size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 3'b000: 1 transfers 3'b001: 4 transfers 3'b010: 8 transfers 3'b011: 16 transfers 3'b100: 32 transfers 3'b101: 64 transfers 3'b110: 128 transfers 3'b111: 256 transfers The unit for transfer is 32bits.
27:16	RW	0x0ff	rx_wmark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits-1 bit less than FIFO-count of status register, which is 13 bits. Limitation: rx_wmark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	tx_wmark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits -1 bit less than FIFO-count of status register, which is 13 bits. Limitation: tx_wmark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)

SDMMC_CDETECT

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x1	card_detect_n Value on card_detect_n input port. 1'b0: Represents presence of card 1'b1: Represents absence of card

SDMMC_WRTPRT

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection.

SDMMC_TCBCNT

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_card_byte_count Number of bytes transferred by CIU unit to card

SDMMC_TBBCNT

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_fifo_byte_count Number of bytes transferred between host/DMA memory and BIU FIFO

SDMMC_DEBNCE

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0xffffff	debounce_count Number of host clock used by debounce filter logic; typical debounce time is 5-25 ms.

SDMMC_USRID

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	usrid User identification register

SDMMC_VERID

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:0	RO	0x5342270a	verid Version identification register

SDMMC_HCON

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	RO	0x1	area_optimized 1'b0: No area optimization 1'b1: Area optimization
25:24	RO	0x0	num_clk_div divider number-1.
23	RO	0x1	set_clk_false_path 1'b0: No false path 1'b1: False path set
22	RO	0x1	impl_hold_reg 1'b0: No hold register 1'b1: Hold register
21	RO	0x0	fifo_ram_inside 1'b0: Outside 1'b1: Inside
20:18	RO	0x1	ge_dma_data_width 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits Others: Reserved
17:16	RO	0x0	dma_interface 2'b00: None 2'b01: INT_DMA 2'b10: GENERIC_DMA 2'b11: NON_INT_DMA
15:10	RO	0x0d	h_addr_width 6'h8: 9 bits 6'h9: 10 bits ..... 6'h1f: 32 bits Others: Reserved
9:7	RO	0x1	h_data_width 3'b000: 16 bits 3'b001: 32 bits 3'b010: 64 bits Others: Reserved

Bit	Attr	Reset Value	Description
6	RO	0x1	h_bus_type 1'b0: APB 1'b1: AHB
5:1	RO	0x00	card_num Card number -1.
0	RO	0x1	card_type Card type. 1'b0: MMC_ONLY 1'b1: SD_MMC

SDMMC_UHSREG

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	ddr_reg DDR mode. These bits indicate DDR mode of operation to the core for the data transfer. 1'b0: Non-DDR mode 1'b1: DDR mode
15:0	RO	0x0000	reserved

SDMMC_RSTN

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x1	card_reset Hardware reset. 1'b0: Active mode 1'b1: Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized.

SDMMC_BMOD

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:11	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
10:8	RO	0x0	pbl Programmable burst length. These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of SDMMC_FIFOTH register. In order to change this value, write the required value to SDMMC_FIFOTH register. This is an encode value as follows. 3'b000: 1 transfers 3'b001: 4 transfers 3'b010: 8 transfers 3'b011: 16 transfers 3'b100: 32 transfers 3'b101: 64 transfers 3'b110: 128 transfers 3'b111: 256 transfers Transfer unit is 32 bits. PBL is a read-only value and is applicable only for data access; it does not apply to descriptor accesses.
7	RW	0x0	de IDMAC enable. When set, the IDMAC is enabled.
6:2	RW	0x00	dsl Descriptor skip length. Specifies the number of word to skip between two unchained descriptors. This is applicable only for dual buffer structure.
1	RW	0x0	fb Fixed burst. Controls whether the AHB 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 AHB will use SINGLE and INCR burst transfer operations.
0	RW	0x0	swr Software reset. When set, the DMA Controller resets all its internal registers. It is automatically cleared after 1 clock cycle.

SDMMC_PLDMND

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	pd Poll demand. If the OWN bit of a descriptor is not set, the FSM goes to the suspend state. The host needs to write any value into this register for the IDMAC FSM to resume normal descriptor fetch operation.

SDMMC_DBADDR

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sbl Start of descriptor list. Contains the base address of the first descriptor. The LSB bits[1:0] are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.

SDMMC_IDSTS

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16:13	RO	0x0	fsm DMAC FSM present state. 4'h0: DMA_IDLE 4'h1: DMA_SUSPEND 4'h2: DESC_RD 4'h3: DESC_CHK 4'h4: DMA_RD_REQ_WAI 4'h5: DMA_WR_REQ_WAI 4'h6: DMA_RD 4'h7: DMA_WR 4'h8: DESC_CLOSE Others: Reserved
12:10	RO	0x0	eb Error bits. Indicates the type of error that caused a bus error. Valid only with fatal bus. 3'h1: Host abort received during transmission 3'h2: Host abort received during reception Others: Reserved
9	RW	0x0	ais Abnormal interrupt summary. Logical OR of the following: SDMMC_IDSTS[2] fatal bus interrupt SDMMC_IDSTS[4] du bit interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes ais to be set is cleared. Writing a 1 clears this bit.
8	RW	0x0	nis Normal interrupt summary. Logical OR of the following: SDMMC_IDSTS[0] transmit interrupt SDMMC_IDSTS[1] receive interrupt Only unmasked bits affect this bit. This is a sticky bit and must be cleared each time a corresponding bit that causes nis to be set is cleared. Writing a 1 clears this bit.
7:6	RO	0x0	reserved
5	RW	0x0	ces Card error summary. Indicates the status of the transaction to/from the card; also present in SDMMC_RINTSTS. Indicates the logical OR of the following bits: EBE: End Bit Error RTO: Response Timeout/Boot Ack Timeout RCRC: Response CRC SBE: Start Bit Error DRTO: Data Read Timeout/BDS timeout DCRC: Data CRC for Receive RE: Response Error Writing a 1 clears this bit. The abort condition of the IDMAC depends on the setting of this CES bit. If the CES bit is enabled, then the IDMAC aborts on a "response error"; however, it will not abort if the CES bit is cleared.

Bit	Attr	Reset Value	Description
4	RW	0x0	dui Descriptor unavailable interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DES0[31] =0). Writing a 1 clears this bit.
3	RO	0x0	reserved
2	RW	0x0	fbe Fatal bus error interrupt. Indicates that a bus error occurred (SDMMC_IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.
1	RW	0x0	ri Receive interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit.
0	RW	0x0	ti Transmit interrupt. Indicates that data transmission is finished for a descriptor. Writing 1 clears this bit.

SDMMC_IDINTEN

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:10	RO	0x000000	reserved
9	RW	0x0	ai Abnormal interrupt summary enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: SDMMC_IDINTEN[2] fatal bus error interrupt SDMMC_IDINTEN[4] du interrupt
8	RW	0x0	ni Normal interrupt summary enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: SDMMC_IDINTEN[0] transmit interrupt SDMMC_IDINTEN[1] receive interrupt
7:6	RO	0x0	reserved
5	RW	0x0	ces Card error summary interrupt enable. When set, it enables the card interrupt summary.
4	RW	0x0	du Descriptor unavailable interrupt. When set along with abnormal interrupt summary enable, the du interrupt is enabled.
3	RO	0x0	reserved
2	RW	0x0	fbe Fatal bus error enable. When set with abnormal interrupt summary enable, the fatal bus error interrupt is enabled. When reset, fatal bus error enable interrupt is disabled.
1	RW	0x0	ri Receive interrupt enable. When set with normal interrupt summary enable, receive interrupt is enabled. When reset, receive interrupt is disabled.

Bit	Attr	Reset Value	Description
0	RW	0x0	ti Transmit interrupt enable. When set with normal interrupt summary enable, transmit interrupt is enabled. When reset, transmit interrupt is disabled.

SDMMC_DSCADDR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hda Host descriptor address pointer. This register points to the start address of the current descriptor read by the IDMAC. Cleared on reset. Pointer updated by IDMAC during operation.

SDMMC_BUFADDR

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hba Host buffer address pointer. This register points to the current data buffer address being accessed by the IDMAC. Cleared on Reset. Pointer updated by IDMAC during operation.

SDMMC_CARDTHRCTL

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	card_rd_thres Card read threshold size
15:2	RO	0x0000	reserved
1	RW	0x0	busy_clr_int_en Busy clear interrupt. 1'b0: Busy clear interrupt disabled 1'b1: Busy clear interrupt enabled Note: The application can disable this feature if it does not want to wait for a busy clear interrupt. For example, in a multi-card scenario, the application can switch to the other card without waiting for a busy to be completed. In such cases, the application can use the polling method to determine the status of busy. By default this feature is disabled and backward-compatible to the legacy drivers where polling is used.
0	RW	0x0	card_rd_thres_en Card read threshold enable. 1'b0: Card read threshold disabled 1'b1: Card read threshold enabled. The host initiates read transfer only if card_rd_thres amount of space is available in receive FIFO.

SDMMC_BACKEND_POWER

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	back_end_power Back end power. 1'b0: Off; Reset 1'b1: Back-end power supplied to card application

SDMMC_EMMCDDR_REG

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	half_start_bit Control for start bit detection mechanism inside Host Controller based on duration of start bit. For eMMC 4.5, start bit can be: 1'b0: Full cycle (half_start_bit=0) 1'b1: Less than one full cycle (half_start_bit=1) Set half_start_bit=1 for eMMC 4.5 and above; set to 0 for SD applications.

SDMMC_RDYINT_GEN

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:25	RO	0x00	reserved
24	RO	0x0	rdyint_cnt_finish Counter finish indication. When high, it indicates that the rdyint counter is finished.
23:16	RO	0xff	rdyint_cnt_status Couner status, reflect internal counter value.
15:9	RO	0x00	reserved
8	RW	0x0	rdyint_gen_working Working indication for rdyint generator. When high, Host Controller start to count and generate one rdyint trigger. After the rdyint trigger is generated, this bit will be set to 0 by Host Controller. So software should set it to 1 before detecting next interrupt.
7:0	RW	0x00	rdyint_gen_maxval Max counter value to detect cdata_in0 high value for generating rdyint, based on internal clock frequency.

SDMMC_FIFO_BASE

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fifo_base_addr FIFO base address

22.5 Interface Description

22.5.1 SDMMC Interface Description

Table 22-8 SDMMC Interface Description

Module Pin	Dir.	PAD Name	IOMUX Setting
sdmmc_cclk	O	PWM10_M0/LPMCU_JTAG_TCK_M1/I2C0_SCL_M2/UART5_RTSN_M0/SDMMC0_CLK/GPIO3_A4_d	VIIOC_GPIO3A_IOMUX_SEL_H[3:0]=4'h1
sdmmc_ccmd	I/O	PWM11_IR_M0/LPMCU_JTAG_TMS_M1/I2C0_SDA_M2/UART5_CTSN_M0/SDMMC0_CMD/GPIO3_A5_u	VIIOC_GPIO3A_IOMUX_SEL_H[11:8]=4'h1
sdmmc_cdata0	I/O	PWM8_M0/TEST_CLK1_OUT/UART2_RX_M0/SDMMC0_D0/GPIO3_A3_u	VIIOC_GPIO3A_IOMUX_SEL_L[15:12]=4'h1
sdmmc_cdata1	I/O	PWM9_M0/TEST_CLK0_OUT/UART2_TX_M0/SDMMC0_	VIIOC_GPIO3A_IOMUX_SEL_L[11:8]

Module Pin	Dir.	PAD Name	IOMUX Setting
		D1/GPIO3_A2_u	=4'h1
sdmmc_cdata2	I/O	HPMCU_JTAG_TCK_M1/A7_JTAG_TCK_M0/UART5_RX_M0/SDMMC0_D2/GPIO3_A7_u	VIIOC_GPIO3A_IOMUX_SEL_H[15:12] =4'h1
sdmmc_cdata3	I/O	HPMCU_JTAG_TMS_M1/A7_JTAG_TMS_M0/UART5_TX_M0/SDMMC0_D3/GPIO3_A6_u	VIIOC_GPIO3A_IOMUX_SEL_H[11:8] =4'h1
sdmmc_cdetn	I	SDMMC0_DET/GPIO3_A1_u	VIIOC_GPIO3A_IOMUX_SEL_L[7:4] =4'h1

Notes: I=input, O=output, I/O=input/output, bidirectional

22.5.2 SDIO Interface Description

Table 22-9 SDIO M0 Interface Description

Module Pin	Dir.	PAD Name	IOMUX Setting
sdio_cclk	O	LCD_D10/I2S0_MCLK/SDMMC1_CLK_M0/GPIO2_A2_d	VOIOC_GPIO2A_IOMUX_SEL_L[11:8] =4'h1
sdio_ccmd	I/O	LCD_D11/I2S0_SDO3_SDI1/SDMMC1_CMD_M0/GPIO2_A3_d	VOIOC_GPIO2A_IOMUX_SEL_L[15:12] =4'h1
sdio_cdata0	I/O	I2C4_SCL_M0/UART1_RTSN_M1/LCD_D9/I2S0_LRCK/SDMMC1_D0_M0/GPIO2_A1_d	VOIOC_GPIO2A_IOMUX_SEL_L[7:4] =4'h1
sdio_cdata1	I/O	I2C4_SDA_M0/UART1_CTSN_M1/LCD_D8/I2S0_SCLK/SDMMC1_D1_M0/GPIO2_A0_d	VOIOC_GPIO2A_IOMUX_SEL_L[3:0] =4'h1
sdio_cdata2	I/O	UART1_RX_M1/LCD_D13/I2S0_SDI0/SDMMC1_D2_M0/GPIO2_A5_d	VOIOC_GPIO2A_IOMUX_SEL_H[7:4] =4'h1
sdio_cdata3	I/O	UART1_TX_M1/LCD_D12/I2S0_SDO0/SDMMC1_D3_M0/GPIO2_A4_d	VOIOC_GPIO2A_IOMUX_SEL_H[3:0] =4'h1

Notes: I=input, O=output, I/O=input/output, bidirectional

Table 22-10 SDIO M1 Interface Description

Module Pin	Dir.	PAD Name	IOMUX Setting
sdio_cclk	O	SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VENCIOC_GPIO1C_IOMUX_SEL_L[11:8]=4'h5
sdio_ccmd	I/O	SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VENCIOC_GPIO1C_IOMUX_SEL_L[15:12]=4'h5
sdio_cdata0	I/O	SDMMC1_D0_M1/SPI0_CLK_M0/PWM4_M2/VICAP_D3_M1/LCD_D6/GPIO1_C1_d	VENCIOC_GPIO1C_IOMUX_SEL_L[7:4] =4'h5
sdio_cdata1	I/O	SDMMC1_D1_M1/SPI0_CS0n_M0/PWM2_M2/VICAP_D2_M1/LCD_D7/GPIO1_C0_d	VENCIOC_GPIO1C_IOMUX_SEL_L[3:0] =4'h5
sdio_cdata2	I/O	SDMMC1_D2_M1/UART4_TX_M1/PWM9_M1/VICAP_D7_M1/LCD_D2/GPIO1_C5_d	VENCIOC_GPIO1C_IOMUX_SEL_H[7:4] =4'h5
sdio_cdata3	I/O	SDMMC1_D3_M1/UART4_RX_M1/PWM8_M1/VICAP_D6_M1/LCD_D3/GPIO1_C4_d	VENCIOC_GPIO1C_IOMUX_SEL_H[3:0] =4'h5

Notes: I=input, O=output, I/O=input/output, bidirectional

22.5.3 EMMC Interface Description

Table 22-11 EMMC Interface Description

Module Pin	Dir.	PAD Name	IOMUX Setting
emmc_cclk	O	FSPI_CLK/EMMC_CLK/GPIO4_B1_d	PERIIOC_GPIO4B_IOMUX_SEL_L[7:4] =4'h1
emmc_ccmd	I/O	FSPI_CS0n/EMMC_CMD/GPIO4_B0_u	PERIIOC_GPIO4B_IOMUX_SEL_L[3:0] =4'h1
emmc_cdata0	I/O	FSPI_D0/EMMC_D0/GPIO4_A4_u	PERIIOC_GPIO4A_IOMUX_SEL_H[3:0] =4'h1
emmc_cdata1	I/O	FSPI_D1/EMMC_D1/GPIO4_A3_u	PERIIOC_GPIO4A_IOMUX_SEL_L[15:12] =4'h1
emmc_cdata2	I/O	FSPI_D2/EMMC_D2/GPIO4_A2_u	PERIIOC_GPIO4A_IOMUX_SEL_L[11:8] =4'h1
emmc_cdata3	I/O	FSPI_D3/EMMC_D3/GPIO4_A6_u	PERIIOC_GPIO4A_IOMUX_SEL_H[11:8] =4'h1
emmc_cdata4	I/O	I2C2_SDA_M1/UART1_TX_M2/SPI1_CS0n_M0/EMMC_D4/GPIO4_A5_u	PERIIOC_GPIO4A_IOMUX_SEL_H[7:4] =4'h1
emmc_cdata5	I/O	I2C2_SCL_M1/UART1_RX_M2/SPI1_CLK_M0/EMMC_D5/GPIO4_A7_u	PERIIOC_GPIO4A_IOMUX_SEL_H[15:12] =4'h1
emmc_cdata6	I/O	TEST_CLK3_OUT/I2C0_SCL_M1/UART0_TX_M2/SPI1_MOSI_M0/EMMC_D6/GPIO4_A1_u	PERIIOC_GPIO4A_IOMUX_SEL_L[7:4] =4'h1
emmc_cdata7	I/O	TEST_CLK2_OUT/I2C0_SDA_M1/UART0_RX_M2/SPI1_MISO_M0/EMMC_D7/GPIO4_A0_u	PERIIOC_GPIO4A_IOMUX_SEL_L[3:0] =4'h1

Notes: I=input, O=output, I/O=input/output, bidirectional

22.6 Application Notes

22.6.1 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.

To avoid glitches in the card clock outputs, the software should use the following steps when changing the card clock frequency:

1) Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of SDMMC_STATUS register.
2) Update the Clock Enable register to disable clock. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:

- start_cmd bit
- "update clock registers only" bits
- "wait_previous data complete" bit

  Wait for the CIU to take the command by polling for 0 on the start_cmd bit.

3) Set the start_cmd bit to update the Clock Divider and/or Clock Source register, and send a command to the CIU in order to update the clock registers; wait for the CIU to take the command.
4) Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.

In non-DMA mode, while reading from a card, the Data Transfer Over (SDMMC_RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA interface unit.

While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.

If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DMA is used, the DMA controller channel should also be reset and reprogrammed.

If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.

One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SD/MMC card (SDMMC_BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO. It is recommended that you not change the FIFO threshold register in the middle of data transfers.

22.6.2 Programming Sequence

22.6.2.1 Initialization

Following figure illustrates the initialization flow.

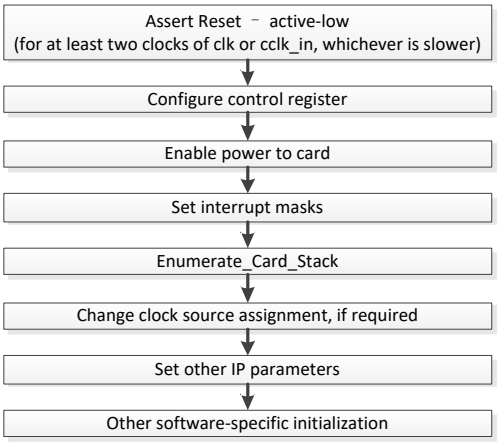


Fig. 22-10 Host Controller Initialization Sequence

Once the power and clocks are stable, reset_n should be asserted(active-low) for at least two cycles of clk or cclk_in, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:

1) Configure control register – For MMC mode, enable the open-drain pullup by setting enable_OD_pullup(bit24) in the SDMMC_CTRL register.
2) Enable power to cards – Before enabling the power, confirm that the voltage setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.
3) Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global int_enable bit of the SDMMC_CTRL register. It is recommended that you write 0xffff_ffff to the Raw Interrupt register in order to clear any pending interrupts before setting the int_enable bit.
4) Enumerate card stack – Each card is enumerated according to card type; for details, refer to "Enumerated Card Stack". For enumeration, you should restrict the clock frequency to 400KHz.
5) Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to "Clock Programming". MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
6) Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in cclk_out according to SD/MMC specifications.

- ResponseTimeOut = 0x64
- DataTimeOut = highest of one of the following:
- (10*((TAAC*Fop)+(100*NSAC))
- Host FIFO read/write latency from FIFO empty/full
- Set the debounce value to 25ms(default:0x0fffff) in host clock cycle units in the SDMMC_DEBNCE register.

FIFO threshold value in bytes in the SDMMC_FIFOTH register.

22.6.2.2 Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SD/MMC card; the card type is first identified and the appropriate card enumeration routine is called.

1) Check if the card is connected.
2) Clear the card type register to set the card width as a single bit. For the given card

number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.

3) Set clock frequency to FOD=400KHz, maximum – Program clock divider0 (bits 0-7 in the SDMMC_CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is 20, 000/(2*400)=25.
4) Identify the card type; that is, SD, MMC, or SDIO.
   a. Send CMD5 first. If a response is received, then the card is SDIO
   b. If not, send CMD8 with the following Argument
      Bit[31:12] = 20'h0 //reserved bits
      Bit[11:8] = 4'b0001 //VHS value
      Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
   c. If Response is received the card supports High Capacity SD2.0 then send ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b1; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
   d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
   e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument
      Bit[31] = 1'b0; //Reserved bits
      Bit[30] = 1'b0; //High Capacity Status
      Bit[29:24] = 6'h0; //Reserved bits
      Bit[23:0] = Supported Voltage Range
5) Enumerate the card according to the card type.
6) Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:

- SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
- MMC – Send CMD0, CMD1, CMD2, CMD3.

22.6.2.3 Clock Programming

The Host Controller supports one clock source. The clock to an individual card can be enabled or disabled. Registers that support this are:

- SDMMC_CLKDIV – Programs individual clock source frequency. SDMMC_CLKDIV limited to 0 or 1 is recommended.
- SDMMC_CLKSRC – Assign clock source for each card.
- SDMMC_CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The Host Controller loads each of these registers only when the start_cmd bit and the update_clk_regs_only bit in the SDMMC_CMD register are set. When a command is successfully loaded, the Host Controller clears this bit, unless the Host Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error). Software should look for the start_cmd and the update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the Host Controller does not raise a command_done signal upon command completion.
The following shows how to program these registers:

1) Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
2) Stop all clocks by writing 0 to the SDMMC_CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
3) Program the SDMMC_CLKDIV and SDMMC_CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the

command.

Re-enable all clocks by programming the SDMMC_CLKENA register. Set the start_cmd, update_clk_regs_only, and wait_prvdata_complete bits in the SDMMC_CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.

22.6.2.4 No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the SDMMC_CMD register @0x2C and the SDMMC_CMDARG register @0x28 with appropriate parameters. Using these two registers, the Host Controller forms the command and sends it to the command bus. The Host Controller reflects the errors in the command response through the error bits of the SDMMC_RINTSTS register.

When a response is received – either erroneous or valid – the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3 register bit 31 represents the MSB, and the Response0 register bit 0 represents the LSB of a long response.

For basic commands or non-data commands, follow these steps:

1) Program the Command register @0x28 with the appropriate command argument parameter.
2) Program the Command register @0x2C with the settings in following table.

Table 22-12 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line, host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC

Parameter	Value	Description
		of response received

3) Wait for command acceptance by host. The following happens when the command is loaded into the Host Controller:
- Host Controller accepts the command for execution and clears the start_cmd bit in the SDMMC_CMD register, unless one command is in process, at which point the Host Controller can load and keep the second command in the buffer.
- If the Host Controller is unable to load the command – that is, a command is already in progress, a second command is in the buffer, and a third command is attempted – then it generates an HLE (hardware-locked error).

4) Check if there is an HLE.
5) Wait for command execution to complete. After receiving either a response from a card or response timeout, the Host Controller sets the command_done bit in the SDMMC_RINTSTS register. Software can either poll for this bit or respond to a generated interrupt.
6) Check if response_timeout error, response_CRC error, or response error is set. This can be done either by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from the SDMMC_RINTSTS register @0x44. If no response error is received, then the response is valid. If required, the software can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

22.6.2.5 Data Transfer Commands

Data transfer commands transfer data between the memory card and the Host Controller. To send a data command, the Host Controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively. For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.

The Host Controller generates an interrupt for different conditions during data transfer, which are reflected in the SDMMC_RINTSTS register @0x44 as:

1) Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the Host Controller does not attempt any data transfer and the "Data Transfer Over" bit is never set.
2) Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.
3) Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
4) Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the Host Controller cannot continue with data transfer. The clock to the card has been stopped.
5) Data read timeout error (bit 9) – Card has not sent data within the timeout period.
6) Data CRC error (bit 7) – CRC error occurred during data reception.
7) Start bit error (bit 13) – Start bit was not received during data reception.
8) End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6), 7), and 8) indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

22.6.2.6 Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

1) Write the data size in bytes in the SDMMC_BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C. The Host Controller expects data from the card in blocks of size SDMMC_BLKSIZ each.
3) Program the SDMMC_CMDARG register @0x28 with the data address of the beginning of a data read.
4) Program the Command register with the parameters listed in following table. For SD

and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 22-13 Command Setting for Single or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, the Host Controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

- Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
- Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.

When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

22.6.2.7 Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

1) Write the data size in bytes in the SDMMC_BYTCNT register @0x20.
2) Write the block size in bytes in the SDMMC_BLKSIZ register @0x1C; the Host Controller sends data in blocks of size SDMMC_BLKSIZ each.

3) Program SDMMC_CMDARG register @0x28 with the data address to which data should be written.
4) Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
5) Program the Command register with the parameters listed in following table.

Table 22-14 Command Settings for Single or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1	Choose value based on speed mode being used;ref to "use_hold_reg" on SDMMC_CMD register
update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No. 0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0/1	-
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0: Sends command immediately 1: Sends command after previous data transfer ends
check_response_crc	1	0: Host Controller should not check response CRC 1: Host Controller should check response CRC

After writing to the SDMMC_CMD register, Host Controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

- Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the SDMMC_RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
- Software should look for Transmit_FIFO_Data_Request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.

When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the SDMMC_RINTSTS register. A response to AUTO_STOP is stored in SDMMC_RESP1 @0x34.

22.6.2.8 Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and the Controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data transfer is in progress; this command can be sent at any time during a data transfer.

You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the Controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the Controller send the command at once, even though there is a data transfer in progress.
Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.

22.6.2.9 Read_Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer—either from function or memory—and allow the host to send commands to any function within the SDIO device. The host can stall this transfer for as long as required. The Host Controller provides the facility to signal this stall transfer to the card. The steps for doing this are:

1) Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. If this bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read this bit.
2) If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the SDMMC_CTRL register @0x00.

Clear the read_wait bit in the SDMMC_CTRL register.

22.6.2.10 Controller/DMA/FIFO Reset Usage

- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the SDMMC_CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset - Resets the FIFO by setting the fifo_reset bit (bit 1) in the SDMMC_CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO under-run or overrun errors in the SDMMC_RAWINTS register caused by the DMA transfers after the FIFO was reset.

22.6.2.11 Card Read Threshold

When an application needs to perform a Single or Multiple Block Read command, the application must program the SDMMC_CARDTHRCTL register with the appropriate Card Read Threshold size (CardRdThreshold) and set the Card Read Threshold Enable (CardRdThrEnable) bit to 1'b1. This additional programming ensures that the Host controller sends a Read Command only if there is space equal to the Card Read Threshold available in the Rx FIFO. This in turn ensures that the card clock is not stopped in the middle a block of data being transmitted from the card. The Card Read Threshold can be set to the block size of the transfer, which guarantees that there is a minimum of one block size of space in the RxFIFO before the controller enables the card clock. The Card Read Threshold is required when the Round Trip Delay is greater than 0.5cclk_in period.

22.6.2.12 Error Handling

The Host Controller implements error checking; errors are reflected in the SDMMC_RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the SDMMC_CTRL register is 0), and all the interrupts are masked (bits 0-31 of the SDMMC_INTMASK register; default is 0).
Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. For data timeout, the Host Controller has not received the data start bit – either for the first block or the intermediate block – within the timeout period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the SDMMC_TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC, start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.
- Hardware locked error – Set when the Host Controller cannot load a command issued by software. When software sets the start_cmd bit in the SDMMC_CMD register, the Host Controller tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO under-run/overrun error – If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an under-run error is set. Before reading or writing data in the FIFO, the software should read the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the Host Controller is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated timeout period. Under this condition and when a read transfer is in process, the software should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.
- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does not send a response, and a response timeout is expected from the Host Controller. The ATA layer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from the device, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the SDMMC_RINTSTS register. It then continues further data transmission until all the bytes are transmitted.

22.6.2.13 Voltage Switching

The Host Controller supports SD 3.0 Ultra High Speed (UHS-1) and is capable of voltage switching in SD-mode, which can be applied to SD High-Capacity (SDHC) and SD Extended Capacity (SDXC) cards. UHS-1 supports only 4-bit mode.

However, whether the IO voltage of 1.8v supported or not is depended on the SoC design.

SD 3.0 UHS-1 supports the following transfer speed modes for UHS-50 and/or UHS-104 cards:

- DS – default-speed up to 25MHz, 3.3V signaling
- HS – high-speed up to 50MHz, 3.3V signaling
- SDR12 – SDR up to SDR 25MHz, 1.8V signaling
- SDR25 – SDR up to 50MHz, 1.8V signaling
- SDR50 – SDR up to 100MHz, 1.8V signaling
- DDR50 – DDR up to 50MHz, 1.8V signaling

Voltage selection can be done in only SD mode. The first CMD0 selects the bus mode-either SD mode or SPI mode. The card must be in SD mode in order for 1.8V signaling mode to apply, during which time the card cannot be switched to SPI mode or 3.3V signaling without a power cycle.

If the System BIOS in an embedded system already knows that it is connected to an SD 3.0 card, then the driver programs the Controller to initiate ACMD41. The software knows from the response of ACMD41 whether or not the card supports voltage switching to 1.8V.

- If bit 32 of ACMD41 response is 1'b1: card supports voltage switching and next command-CMD11-invokes voltage switching sequence. After CMD11 is started, the software must program the IO voltage selection register based on the soc architecture.

- If bit 32 of ACMD41 response is 1'b0: card does not support voltage switching and CMD11 should not be started.

If the card and host controller accept voltage switching, then they support UHS-1 modes of data transfer. After the voltage switch to 1.8V, SDR12 is the default speed.
Since the UHS-1 can be used in only 4-bit mode, the software must start ACMD6 and change the card data width to 4-bit mode; ACMD6 is driven in any of the UHS-1 speeds. If the host wants to select the DDR mode of data transfer, then the software must program the SDMMC_DDR_REG register in the CSR space with the appropriate card number.
To choose from any of the SDR or DDR modes, appropriate values should be programmed in the SDMMC_CLKDIV register.

22.6.2.14 Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

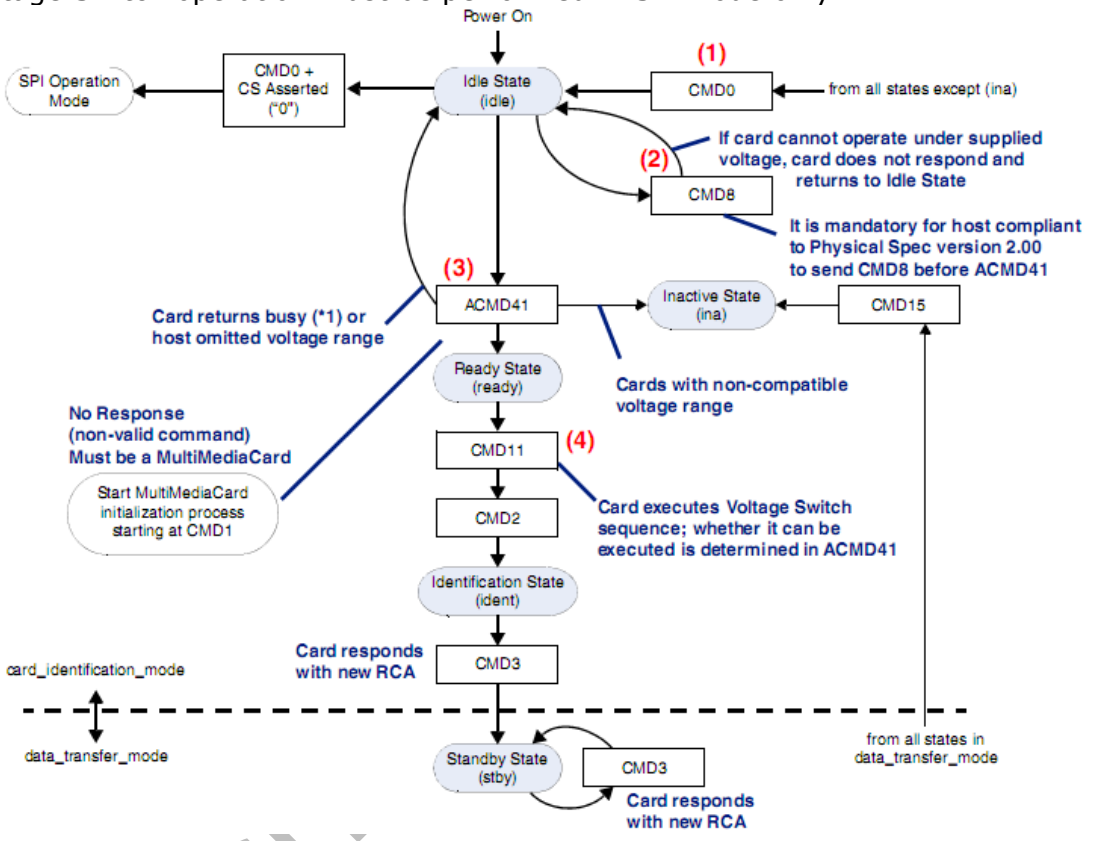


Fig. 22-11 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

1) Software Driver starts CMD0, which selects the bus mode as SD.
2) After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2. 00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
3) ACMD 41 is started. The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to following figure.

47	46	45-40	39	38	37	36	35-33	32	31-16	15-08	07-01	00
S	D	Index	Busy 31	HCS 30	(FB) 29	XPC 28	Reserved 27-25	S18R 24	OCR 23-08	Reserved 07-00	CRC7	E
0	1	101001	0	X	0	X	000	X	xxxxh	0000000	xxxxxxx	1

Host Capacity Support
0b: SDSC-only Host
1b: SDHC or SDXC supported

SCXC Power Control
0b: Power saving
1b: Maximum performance

S18R: Switching to 1.8V Request
0b: Use current signal voltage
1b: Switch to 1.8V signal voltage

Fig. 22-12 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to following figure.

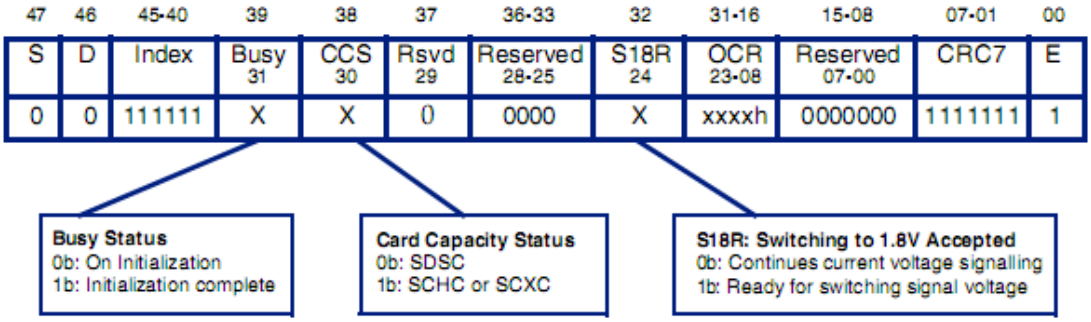


Fig. 22-13 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
- Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
- Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process

If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

22.6.2.15 Voltage Switch Normal Scenario

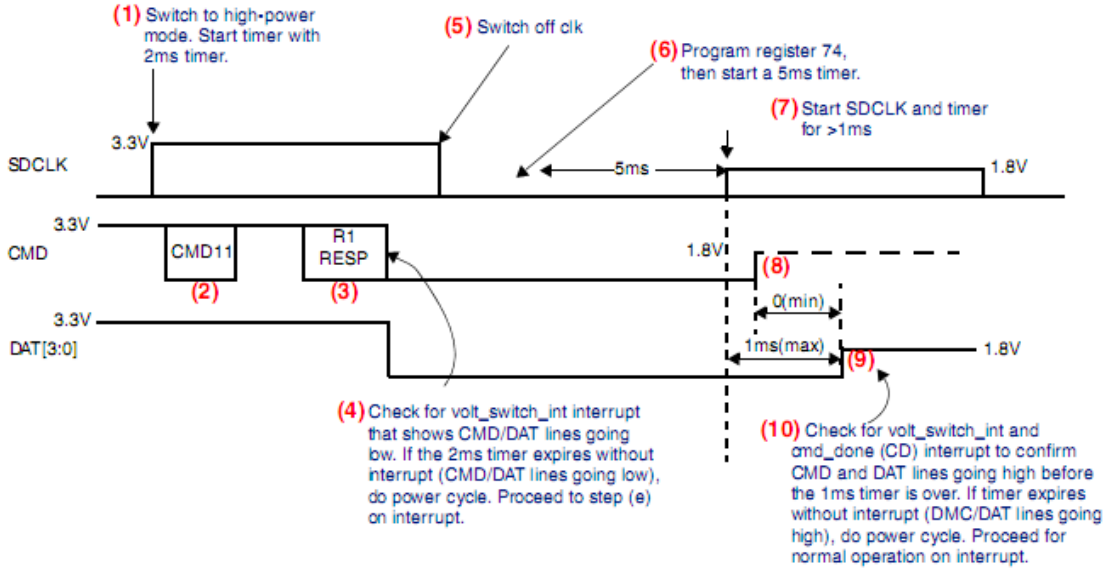


Fig. 22-14 Voltage Switch Normal Scenario

1) The host programs SDMMC_CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below: Total cycles required for CMD11 = 48 cycles
   Total cycles required for RESP R1 = 48 cycles
   Maximum clock delay between MCD11 end to start of RESP1 = 60 cycles
   Total = 48+48 + 60 = 160
   Minimum frequency during enumeration is 100 KHz; that is, 10us
   Total time = 160 * 10us = 1600us = 1. 6ms ~ 2ms
2) The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".
3) The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
4) The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires

without an interrupt (CMD/DAT lines going low), do a power cycle.

Note: Before doing a power cycle, switch off the card clock by programming SDMMC_CLKENA register

Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).

Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be masked during the voltage switch sequence.

If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle. Proceed to step (5) on interrupt.

1) Program the SDMMC_CLKENA register, with 0 for the corresponding card; the host stops supplying SDCLK.
2) Program Voltage register to the required values for the corresponding card. The application should start a timer > 5ms.
3) After the 5ms timer expires, the host voltage regulator is stable. Program SDMMC_CLKENA register, with 1 for the corresponding card; the host starts providing SDCLK at 1. 8V; this can be at zero time after Voltage register has been programmed. When the SDMMC_CLKENA register is programmed, the application should start another timer > 1ms.
4) By detecting SDCLK, the card drives CMD to high at 1. 8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
5) If switching to 1. 8V signaling is completed successfully, the card drives DAT [3:0] to high at 1. 8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
6) The host controller generates a voltage switch interrupt (VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.

If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the SDMMC_CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

22.6.2.16 Voltage Switch Error Scenario

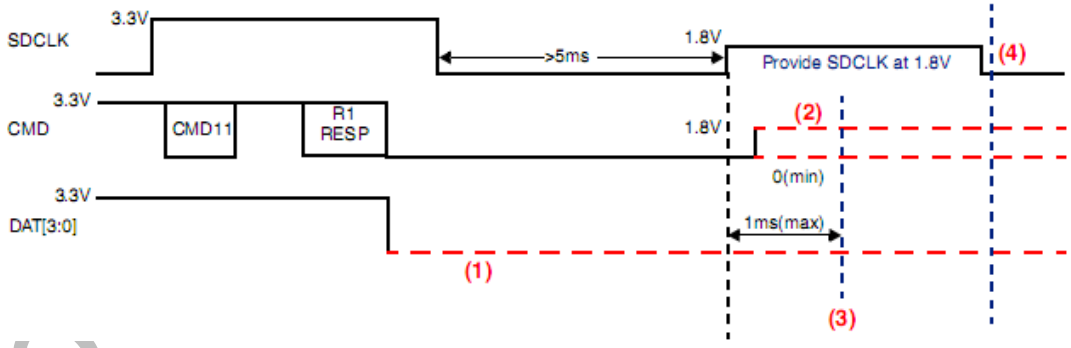


Fig. 22-15 Voltage Switch Error Scenario

1) If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.

Note: Before performing a power cycle, switch off the card clock by programming SDMMC_CLKENA register; no cmd_done (CD) interrupt is generated.

Additionally, if the card detects a voltage error at any point in between steps (5) and (7) in the card keeps driving DAT[3:0] to low until card power off.

2) CMD can be low or tri-state.
3) The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done.

If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the SDMMC_CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.

4) If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.

Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2. 5V. Errors are indicated by (1) and (2).

- If voltage switching is accepted by the card, the default speed is SDR12.
- Command Done is given:
  - If voltage switching is properly done, CMD and DAT line goes high.
  - If switching is not complete, the 1ms timer expires, and the card clock is switched off.

Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.

- The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected.

After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

22.6.3 DDR Operation

22.6.3.1 4-bit DDR Programming Sequence

DDR programming should be done only after the voltage switch operation has completed. The following outlines the steps for the DDR programming sequence:

1) Once the voltage switch operation is complete, the user must program voltage selection register to the required values for the corresponding card.

- To start a card to work in DDR mode, the application must program a bit of the newly defined SDMMC_UHSREG[16] register with a value of 1'b1.
- The bit that the user programs depends on which card is to be accessed in DDR mode.

To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should SDMMC_UHSREG[16]be set back to 1'b0 for the appropriate card.

22.6.3.2 8-bit DDR Programming Sequence

The following outlines the steps for the 8-bit DDR programming sequence:

1) The cclk_in signal should be twice the speed of the required cclk_out. Thus, if the cclk_out signal is required to be 50 MHz, the cclk_in signal should be 100 MHz.
2) The SDMMC_CLKDIV register should always be programmed with a value higher than zero (0); that is, a clock divider should always be used for 8-bit DDR mode.
3) The application must program the SDMMC_UHSREG[16] register (ddr_reg bits) by assigning it with a value of 1 for the bit corresponding to the card number; this causes the selected card to start working in DDR mode.

Depending on the card number, the SDMMC_CTYPE [16] bits should be set in order to make the host work in the 8-bit mode.

22.6.3.3 eMMC4.5 DDR START Bit

The eMMC4.5 changes the START bit definition in the following manner:

1) Receiver samples the START bit on the rising edge.
2) On the next rising edge after sampling the START bit, the receiver must sample the data.
3) Removes requirement of the START bit and END bit to be high for one full cycle.

Notes: The Host Controller does not support a START bit duration higher than one clock cycle. START bit durations of one or less than one clock cycle are supported and can be defined at the time of startup by programming the EMMC_DDR_REG register.

22.6.3.4 Reset Command/Moving From DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

1) Issue CMD0.
2) When CMD0 is received, the card changes from DDR50 to SDR12.
3) Program the SDMMC_CLKDIV register with an appropriate value.
4) Set ddr_reg to 0.

Note: The Voltage register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

22.6.4 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

The following outlines the steps for the H/W reset programming sequence:

1) Program CMD12 to end any transfer in process.
2) Wait for DTO, even if no response is sent back by the card.
3) Set the following resets:

- DMA reset–SDMMC_CTRL [2] bit
- FIFO reset–SDMMC_CTRL [1] bit

Note: The above steps are required only if a transfer is in process.

4) Program the SDMMC_RSTN register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n signal and resets the card.
5) Wait for minimum of 1 µs or cclk_in period, whichever is greater
6) After a minimum of 1 µs, the application should program a value of 0 into the SDMMC_RSTN register. This de-asserts the RST_n signal and takes the card out of reset.
7) The application can program a new CMD only after a minimum of 200 µs after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.

Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

22.6.5 FBE Scenarios

An FBE occurs due to an AHB error response on the AHB bus. This is a system error, so the software driver should not perform any further programming to the Host. The only recovery mechanism from such scenarios is to do one of the following:

- Issue a hard reset by asserting the reset_n signal
- Do a program controller reset by writing to the SDMMC_CTRL[0] bit

22.6.5.1 FIFO Overflow and Underflow

During normal data transfer conditions, FIFO overflow and underflow will not occur. However if there is a programming error, then FIFO overflow/underflow can result. For example, consider the following scenarios.

- For transmit: PBL=4, Tx watermark = 1. For the above programming values, if the FIFO has only one location empty, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pushes into the FIFO. This will result in a FIFO overflow interrupt.
- For receive: PBL=4, Rx watermark = 1. For the above programming values, if the FIFO has only one location filled, it issues a dma_req to IDMAC FSM. Due to PBL value=4, the IDMAC FSM performs 4 pops to the FIFO. This will result in a FIFO underflow interrupt.

The driver should ensure that the number of bytes to be transferred as indicated in the descriptor should be a multiple of 4bytes with respect to H_DATA_WIDTH=32. For example, if the SDMMC_BYTCNT=13, the number of bytes indicated in the descriptor should be 16 for H_DATA_WIDTH=32.

22.6.5.2 Programming of PBL and Watermark Levels

The DMAC performs data transfers depending on the programmed PBL and threshold values.

Table 22-15 PBL and Watermark Levels

PBL (Number of transfers)	Tx/Rx Watermark Value
1	Greater than or equal to 1
4	Greater than or equal to 4
8	Greater than or equal to 8
16	Greater than or equal to 16
32	Greater than or equal to 32
64	Greater than or equal to 64
128	Greater than or equal to 128
256	Greater than or equal to 256

22.6.6 Variable Delay Usage

The delay time of every element is in the range of 36ps~68ps, varying with different voltage and temperature.

The control signals for variable delay element usage are shown as follows.

22.6.6.1 SDMMC Variable Delay Usage

Table 22-16 Configuration for SDMMC Variable Delay Usage

Signal Name	Source	Default	Description
init_state	VI_GRF_SDMMC_CON1[0]	0	Enable initialization for clock source. 1'b0: Disable 1'b1: Enable When init_state=1, the host clocks including drive clock and sample clock are inactive.
drv_degree[1:0]	VI_GRF_SDMMC_CON1[2:1]	2	Phase selection for drive clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
drv_delaynum[7:0]	VI_GRF_SDMMC_CON1[10:3]	0	Delay element number configuration for drive clock. It can be modified when init_state=1 and should be stable when init_state=0.
drv_sel	VI_GRF_SDMMC_CON1[11]	0	Selection for drive clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.
sample_degree[1:0]	VI_GRF_SDMMC_CON2[2:1]	0	Phase selection for sample clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
sample_delaynum[7:0]	VI_GRF_SDMMC_CON2[10:3]	0	Delay element number configuration for sample clock. It can be modified when init_state=1 and should be stable when init_state=0.
sample_sel	VI_GRF_SDMMC_CON2[11]	0	Selection for sample clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.

22.6.6.2 SDIO Variable Delay Usage

Table 22-17 Configuration for SDIO Variable Delay Usage

Signal Name	Source	Default	Description
init_state	VO_GRF_SDIO_CON0[0]	0	Enable initialization for clock source. 1'b0: Disable 1'b1: Enable When init_state=1, the host clocks including drive clock and sample clock are inactive.
drv_degree[1:0]	VO_GRF_SDIO_CON0[2:1]	2	Phase selection for drive clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
drv_delaynum[7:0]	VO_GRF_SDIO_CON0[10:3]	0	Delay element number configuration for drive clock. It can be modified when init_state=1 and should be stable when init_state=0.
drv_sel	VO_GRF_SDIO_CON0[11]	0	Selection for drive clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.
sample_degree[1:0]	VO_GRF_SDIO_CON1[2:1]	0	Phase selection for sample clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
sample_delaynum[7:0]	VO_GRF_SDIO_CON1[10:3]	0	Delay element number configuration for sample clock. It can be modified when init_state=1 and should be stable when init_state=0.
sample_sel	VO_GRF_SDIO_CON1[11]	0	Selection for sample clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.

22.6.6.3 EMMC Variable Delay Usage

Table 22-18 Configuration for EMMC Variable Delay Usage

Signal Name	Source	Default	Description
init_state	PERI_GRF_EMMC_CON0[	0	Enable initialization for clock

Signal Name	Source	Default	Description
	0]		source. 1'b0: Disable 1'b1: Enable When init_state=1, the host clocks including drive clock and sample clock are inactive.
drv_degree[1:0]	PERI_GRF_EMMC_CON0[2:1]	2	Phase selection for drive clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
drv_delaynum[7:0]	PERI_GRF_EMMC_CON0[10:3]	0	Delay element number configuration for drive clock. It can be modified when init_state=1 and should be stable when init_state=0.
drv_sel	PERI_GRF_EMMC_CON0[11]	0	Selection for drive clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.
sample_degree[1:0]	PERI_GRF_EMMC_CON1[2:1]	0	Phase selection for sample clock. 2'h0: 0-degree 2'h1: 90-degree 2'h2: 180-degree 2'h3: 270-degree It can be modified when init_state=1 and should be stable when init_state=0.
sample_delaynum[7:0]	PERI_GRF_EMMC_CON1[10:3]	0	Delay element number configuration for sample clock. It can be modified when init_state=1 and should be stable when init_state=0.
sample_sel	PERI_GRF_EMMC_CON1[11]	0	Selection for sample clock: 1'b0: Select clock delayed by phase shift 1'b1: Select clock delayed by phase shift and delay line It can be modified when init_state=1 and should be stable when init_state=0.

The following outlines the steps for clock generation sequence:

1) Assert init_state to soft reset the CLKGEN.
2) Configure drv_degree/sample_degree.
3) If fine adjustment required, delay line can be used by configuring drv_delaynum/sample_delaynum and drv_sel/sample_sel.
4) Dis-assert init_state to start CLKGEN.

22.6.7 Variable Delay Tuning

Tuning is defined by SD and MMC cards to determine the correct sampling point required for the host, especially for the speed modes SDR104 and HS200 where the output delays from the cards can be up to 2 UI. Tuning is required for other speed modes-such as DDR50-even though the output delay from the card is less than one cycle.
Command for tuning is different for different cards.

- SD Memory Card:
  - CMD19 – SD card for SDR50 and SDR104 speed modes. Tuning data is defined by card specifications.
  - CMD6 – SD card for speed modes not supporting CMD19. Tuning data is the 64byte SD status.
- Multimedia Card:
  - CMD21 – MMC card for HS200 speed mode. Tuning data is defined by card specifications.
  - CMD8 – MMC card for speed modes not supporting CMD21. Tuning data is 512 byte ExtCSD data.

The following is the procedure for variable delay tuning:

1) Set a phase shift of 0-degree on cclk_in_sample.
2) Send the Tuning command to the card; the card in turn sends an R1 response on the CMD line and tuning data on the DAT line.
3) If the host sees any of the errors—start bit error, data crc error, end bit error, data read time-out, response CRC error, response error—then the sampling point is incorrect.
4) Send CMD12 to bring the host controller state machines to idle.
   - The card may treat CMD12 as an invalid command because the card has successfully sent the tuning data, and it cannot send a response.
   - The host controller may generate a response time-out interrupt that must be cleared by software.
5) Repeat steps 2) to 4) by increasing the phase shift value or delay element number on cclk_in_sample until the correct sampling point is received such that the host does not see any of the errors.
6) Mark this phase shift value as the starting point of the sampling window.
7) Repeat steps 2 to 4 by increasing the phase shift value or delay element number on cclk_in_sample until the host sees the errors starting to come again or the phase shift value reaches 360-degree.
8) Mark the last successful phase shift value as the ending point of the sampling window.

A window is established where the tuning block is matched. For example, for a scenario where the tuning block is received correctly for a phase shift window of 90-degree and 180-degree, then an appropriate sampling point is established as 135-degree. Once a sampling point is established, no errors should be visible in the tuning block.

22.6.8 Card Detection Method

There are many methods for SDMMC/SDIO device detection.

- Method1: Using SDMMC_CDETECT register, which is value on card_detect_n input port. 0 represents presence of card. This method is available only for SDMMC/SDIO.
- Method2: Using card detection unit in Host Controller, outputting host interrupt. The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value in SDMMC_DEBNCE [23:0]. Following figure illustrates the timing for card-detect signals. This method is available only for SDMMC/SDIO.

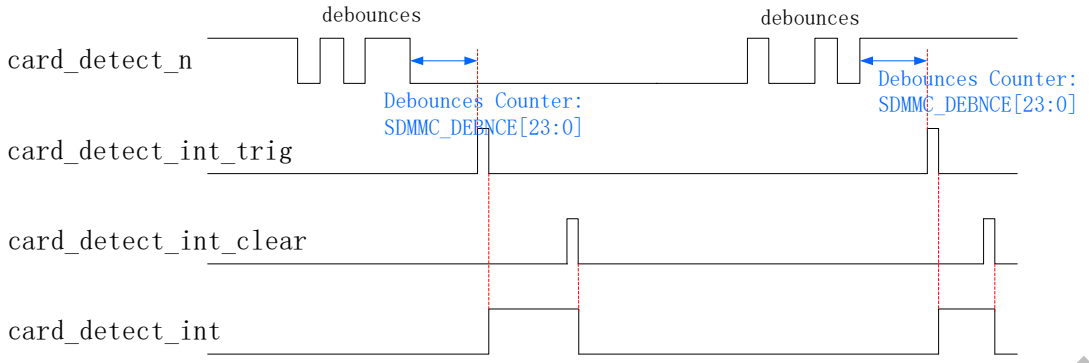


Fig. 22-16 Card Detection Method 2

- Method3: Using general card detection unit, outputting detect_dual_edge_int connecting to IRQ[308]. Similar to Method2, except that the debounce time is configurable by PMU1_GRF_SD_DETECT_CNT; and the insertion/removal detection interrupt can be enabled or cleared respectively.

The detailed register information is:

Table 22-19 Register for SDMMC Card Detection Method 3

Signal Name	Source	Default	Description
sd_detectn_rise_edge_irq_en	VI_GRF_SIG_DETECT_CON[0]	0	Enable SDMMC detect pin rising edge interrupt. 1'b0: Disable 1'b1: Enable
sd_detect_fall_edge_detect_en	VI_GRF_SIG_DETECT_CON[1]	0	Enable SDMMC detect pin falling edge interrupt. 1'b0: Disable 1'b1: Enable
sd_detect_rising_edge_dectect_status	VI_GRF_SIG_DETECT_STS[0]	0	SDMMC detect pin rising edge status. 1'b0: Inactive 1'b1: Active
sd_detect_fall_edge_detect_status	VI_GRF_SIG_DETECT_STS[1]	0	SDMMC detect pin falling edge status. 1'b0: Inactive 1'b1: Active
sd_detect_rising_edge_dectect_clr	VI_GRF_SIG_DETECT_CLR[0]	0	Enable clear for SDMMC detect pin rising edge interrupt. 1'b0: Disable 1'b1: Enable
sd_detect_fall_edge_detect_clr	VI_GRF_SIG_DETECT_CLR[1]	0	Enable clear for SDMMC detect pin falling edge interrupt. 1'b0: Disable 1'b1: Enable

22.6.9 SDMMC IOMUX With JTAG

The GPIO4d2/GPIO4d3 for sdmmc_cdata2/sdmmc_cdata3 is shared with jtag_tck/jtag_tms. The condition of usage for SDMMC or JTAG usage is as follows.

- If VIIOC_FORCE_JTAG_SDMMC[0] is equal to 1 and SD/MMC card is not detected within detection time(defined in VI_GRF_SDMMC_DET, in the unit of XIN24M clock), the GPIOs are used for JTAG.
- Otherwise, the GPIOs' usage is defined by IOMUX configuration.

Chapter 23 Serial Peripheral Interface (SPI)

23.1 Overview

The serial peripheral interface is an APB slave device. 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 64-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 combining interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one quarter 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
- Support timeout mechanism in slave mode
- Support BYPASS slave mode, in which RX and TX logic is drive by SCLK_IN directly instead of spi_clk

23.2 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

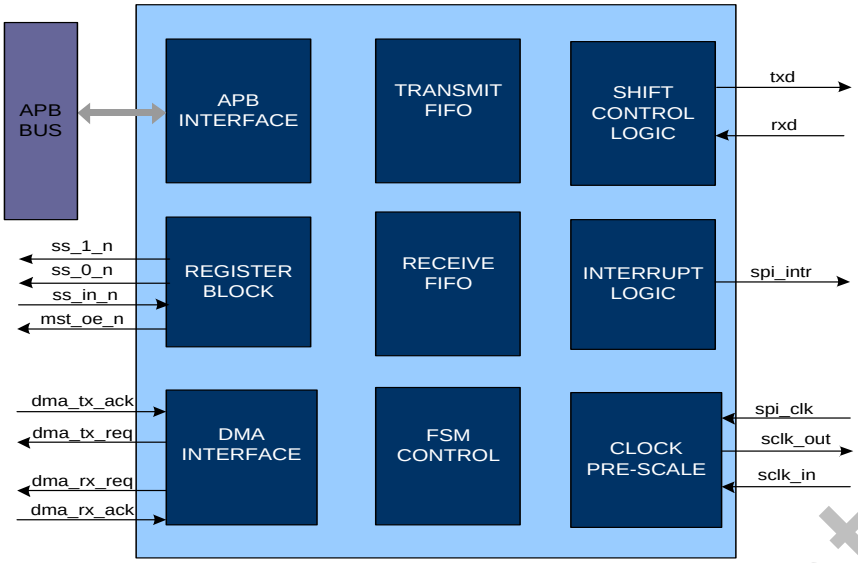


Fig. 23-1 SPI Controller Block Diagram

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 64x16bits.

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 OR relationship between all other SPI interrupts after masking.

23.3 Function Description

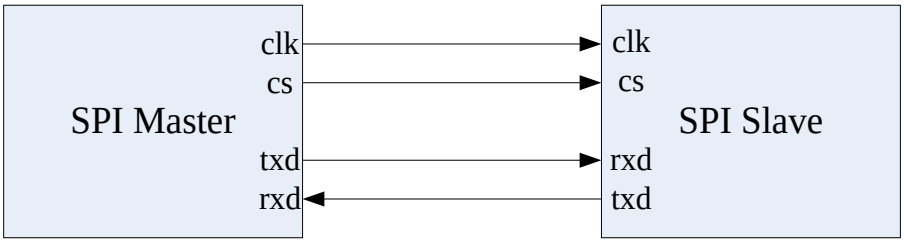


Fig. 23-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram shows 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.

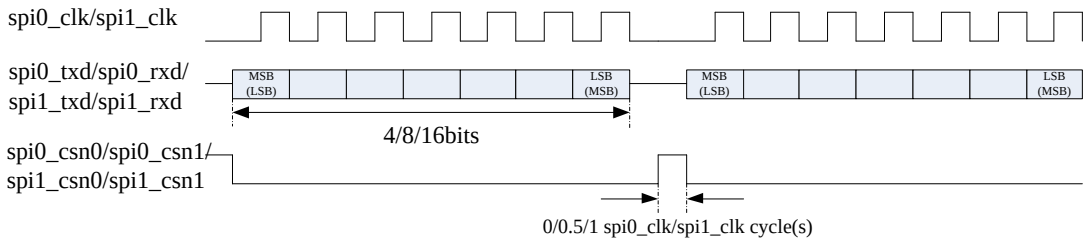


Fig. 23-3 SPI Format (SCPH=0 SCPOL=0)

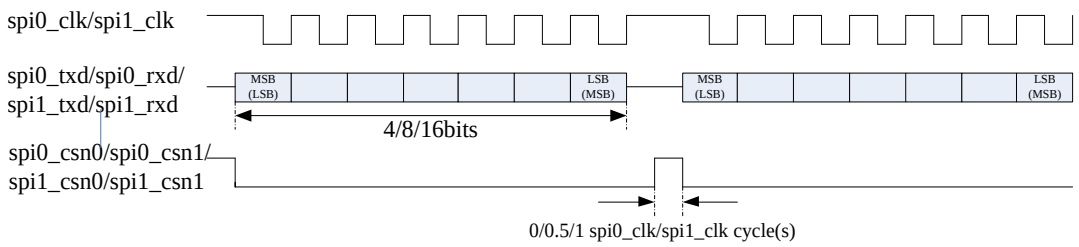


Fig. 23-4 SPI Format (SCPH=0 SCPOL=1)

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.

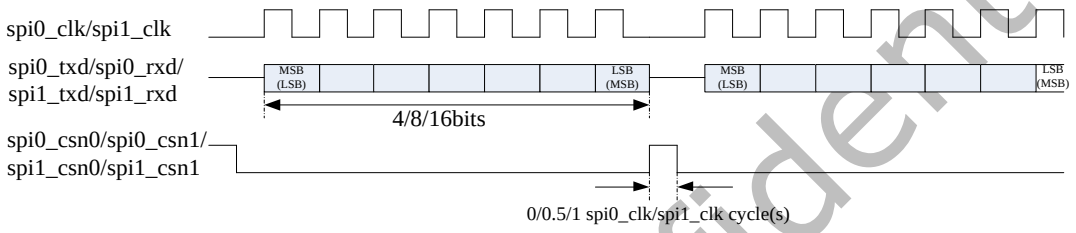


Fig. 23-5 SPI Format (SCPH=1 SCPOL=0)

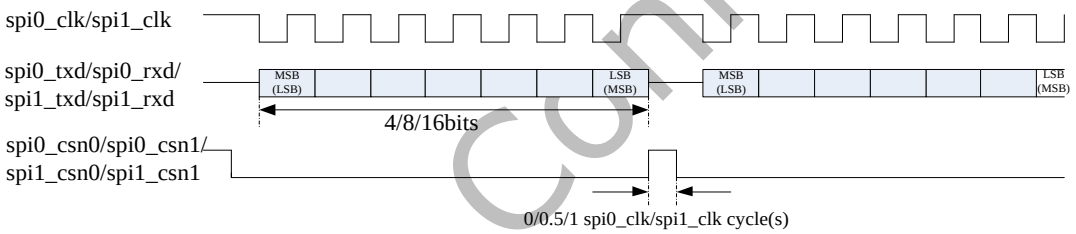


Fig. 23-6 SPI Format (SCPH=1 SCPOL=1)

23.4 Register Description

23.4.1 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	0x0000004C	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

Name	Offset	Size	Reset Value	Description
SPI_TIMEOUT	0x004C	W	0x00000000	Timeout control register
SPI_BYPASS	0x0050	W	0x00000000	BYPASS control register
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, **DW**-Double WORD (64 bits) access*

23.4.2 Detail Registers Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:26	RO	0x00	reserved
25	RW	0x0	lbk Loop back mode select. 1'b0: Normal mode. 1'b1: Loop back mode,rxd is connected to txd.
24:23	RW	0x0	soi SS_N output inverted. 1'b0: Corresponding bit of ss_in is not inverted. 1'b1: Corresponding bit of ss_in is inverted.
22	RO	0x0	reserved
21	RW	0x0	mtm Valid when frame format is set to National Semiconductors Microwire. 1'b0: Non sequential transfer 1'b1: Sequential transfer
20	RW	0x0	opm Master and slave mode select. 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	xfm Transmit and receive mode select. 2'b00: Transmit & Receive 2'b01: Transmit Only 2'b10: Receive Only 2'b11: Reserved
17:16	RW	0x0	frf 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11: Reserved
15:14	RW	0x0	rsd 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 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.

Bit	Attr	Reset Value	Description
12	RW	0x0	fbm 1'b0: First bit is MSB. 1'b1: First bit is LSB.
11	RW	0x0	em 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 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 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 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 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.
5:2	RW	0x0	cfs 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 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
31:0	RW	0x00000000	ndm 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 4GB of data in a continuous transfer.

SPI_ENR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	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	0x00000000	reserved
1:0	RW	0x0	ser Slave enable register.The register enable the individual slave select output lines, 2 slave-select output pins are available.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	0x0000	reserved
15:0	RW	0x0000	baudr SPI Clock Divider. 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
31:6	RO	0x0000000	reserved
5:0	RW	0x00	xftlr 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
31:6	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	rxftlr 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
31:7	RO	0x0000000	reserved
6:0	RO	0x00	txflr Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:0	RO	0x00	rxflr Contains the number of valid data entries in the receive FIFO.

SPI_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RO	0x1	ssi 1'b0: ss_in_n is low. 1'b1: ss_in_n is high.
5	RO	0x0	stb 1'b0: Slave tx not busy. 1'b1: Slave tx busy.
4	RO	0x0	rff 1'b0: Receive FIFO is not full. 1'b1: Receive FIFO is full.
3	RO	0x1	rfe 1'b0: Receive FIFO is not empty. 1'b1: Receive FIFO is empty.
2	RO	0x1	tfe 1'b0: Transmit FIFO is not empty. 1'b1: Transmit FIFO is empty.
1	RO	0x0	tff 1'b0: Transmit FIFO is not full. 1'b1: Transmit FIFO is full.
0	RO	0x0	bsf 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	0x00000000	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
31:8	RO	0x000000	reserved
7	RW	0x0	txfim 1'b0: TX finish interrupt is masked. 1'b1: TX finish interrupt is not masked.
6	RW	0x0	sspim 1'b0: ss_in_n posedege interrupt is masked. 1'b1: ss_in_n posedege interrupt is not masked.
5	RW	0x0	toim 1'b0: spi timeout interrupt is masked. 1'b1: spi timeout interrupt is not masked.
4	RW	0x0	rffim 1'b0: spi_rxf_intr interrupt is masked. 1'b1: spi_rxf_intr interrupt is not masked.
3	RW	0x0	rfoim 1'b0: spi_rxo_intr interrupt is masked. 1'b1: spi_rxo_intr interrupt is not masked.
2	RW	0x0	rfuim 1'b0: spi_rxu_intr interrupt is masked. 1'b1: spi_rxu_intr interrupt is not masked.
1	RW	0x0	tfoim 1'b0: spi_txo_intr interrupt is masked. 1'b1: spi_txo_intr interrupt is not masked.
0	RW	0x0	tfeim 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
31:8	RO	0x000000	reserved
7	RW	0x0	txfis 1'b0: TX finish interrupt is not active after masking. 1'b1: TX finish interrupt is active after masking.
6	RW	0x0	sspis 1'b0: ss_in_n posedege interrupt is not active after masking. 1'b1: ss_in_n posedege interrupt is active after masking.
5	RW	0x0	tois 1'b0: spi timeout interrupt is not active after masking. 1'b1: spi timeout interrupt is active after masking.
4	RO	0x0	rffis 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 1'b0: spi_rxo_intr interrupt is not active after masking. 1'b1: spi_rxo_intr interrupt is active after masking.
2	RO	0x0	rfuis 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 1'b0: spi_txo_intr interrupt is not active after masking. 1'b1: spi_txo_intr interrupt is active after masking.

Bit	Attr	Reset Value	Description
0	RO	0x0	tfeis 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
31:8	RO	0x000000	reserved
7	RW	0x0	txfris 1'b0: TX finish interrupt is not active prior to masking. 1'b1: TX finish interrupt is active prior to masking.
6	RW	0x0	sspris 1'b0: ss_in_n posedege interrupt is not active prior to masking. 1'b1: ss_in_n posedege interrupt is active prior to masking.
5	RW	0x0	toris 1'b0: spi_timeout interrupt is not active prior to masking. 1'b1: spi_timeout interrupt is active prior to masking.
4	RO	0x0	rffris 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 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 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 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 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
31:7	RO	0x0000000	reserved
6	WO	0x0	ctxfi Write 1 to Clear tx finish Interrupt.
5	WO	0x0	csspi Write 1 to Clear ss_in_n posdege Interrupt.
4	WO	0x0	ctoi Write 1 to Clear Timeout Interrupt.
3	WO	0x0	ctfoi Write 1 to Clear Transmit FIFO Overflow Interrupt.
2	WO	0x0	crfoi Write 1 to Clear Receive FIFO Overflow Interrupt.
1	WO	0x0	crfui Write 1 to Clear Receive FIFO Underflow Interrupt.
0	WO	0x0	cci Write 1 to Clear Combined Interrupt.

SPI_DMACR

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RW	0x0	tde 1'b0: Transmit DMA disabled. 1'b1: Transmit DMA enabled.
0	RW	0x0	rde 1'b0: Receive DMA disabled. 1'b1: Receive DMA enabled.

SPI_DMATDLR

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	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 transmit FIFO is equal to or below this field value, and transmit DMA is enabled (DMACR[1] =1).

SPI_DMARDLR

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:0	RW	0x00	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, and receive DMA is enabled (DMACR[0]=1).

SPI_TIMEOUT

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	toe Timeout enable. 1'b0: Timeout counter is inactive. 1'b1: Timeout counter will be active after the first rising edge of sclk_in.
15:0	RW	0x0000	tov Timeout threshold value. If sclk_in keep inactive for a threshold time, timeout interrupt will be triggered. The timeout threshold time is TOV*pclk_perid*16.

SPI_BYPASS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	txcp TX clock polarity.This bit is only valid in bypass mode. 1'b0: TX logic use raw SCLK. 1'b1: TX logic use inverted SCLK.

Bit	Attr	Reset Value	Description
3	RW	0x0	rxcp RX clock polarity.This bit is only valid in bypass mode. 1'b0: RX logic use raw SCLK. 1'b1: RX logic use inverted SCLK.
2	RW	0x0	end Endian mode.This bit is only valid in bypass mode. 1'b0: Work in little endian mode. 1'b1: Work in big endian mode.
1	RW	0x0	fbm First bit mode.This bit is only valid in bypass mode. 1'b0: First bit is LSB. 1'b1: First bit is MSB.
0	RW	0x0	byen Bypass enable. 1'b0: Normal mode. 1'b1: Bypass mode, SPI serial/parallel convert logic is drive by SCLK.

SPI_TXDR

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15:0	WO	0x0000	txdr 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	0x0000	reserved
15:0	RO	0x0000	rxdr When the register is read, data in the receive FIFO is accessed.

23.5 Interface Description

Table 23-1 SPI0 Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
spi_sclk	I/O	SDMMC1_D0_M1/SPI0_CLK_M0/PWM4_M2/VICAP_D3_M1/LCD_D6/GPIO1_C1_d	VENC_IOC_GPIO1C_IOMUX_SEL_L[6:4]=3'h4
spi_mosi	I/O	SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VENC_IOC_GPIO1C_IOMUX_SEL_L[10:8]=3'h6
spi_miso	I/O	SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VENC_IOC_GPIO1C_IOMUX_SEL_L[14:12]=3'h6
spi_csn0	I/O	SDMMC1_D1_M1/SPI0_CS0n_M0/PWM2_M2/VICAP_D2_M1/LCD_D7/GPIO1_C0_d	VENC_IOC_GPIO1C_IOMUX_SEL_L[2:0]=3'h4
spi_csn1	O	DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VENC_IOC_GPIO1D_IOMUX_SEL_L[10:8]=3'h5

Table 23-2 SPI1 Tnterface Ddescription

Module Pin	Dir	Pad Name	IOMUX Setting
spi_sclk	I/O	I2C2_SCL_M1/UART1_RX_M2/SPI1_CLK_M0/EMMC_D5/GPIO4_A7_u	PERI_IOC_GPIO4A_IOMUX_SEL_H[14:12]=3'h2

spi_mosi	I/O	TEST_CLK3_OUT/I2C0_SCL_M1/UART0_TX_M2/SPI1_MOSI_M0/EMMC_D6/GPIO4_A1_u	PERI_IOC_GPIO4A_IOMUX_SEL_L[6:4]=3'h2
spi_miso	I/O	TEST_CLK2_OUT/I2C0_SDA_M1/UART0_RX_M2/SPI1_MISO_M0/EMMC_D7/GPIO4_A0_u	PERI_IOC_GPIO4A_IOMUX_SEL_L[2:0]=3'h2
spi_csn0	I/O	I2C2_SDA_M1/UART1_TX_M2/SPI1_CS0n_M0/EMMC_D4/GPIO4_A5_u	PERI_IOC_GPIO4A_IOMUX_SEL_H[6:4]=3'h2
spi_csn1	O	VICAP_D1_M1/SPI1_CS1n_M0/PWM7_IR_M1/UART4_TX_M0/GPIO1_B1_d	VENC_IOC_GPIO1B_IOMUX_SEL_L[6:4]=3'h3

23.6 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>= 4 × (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.

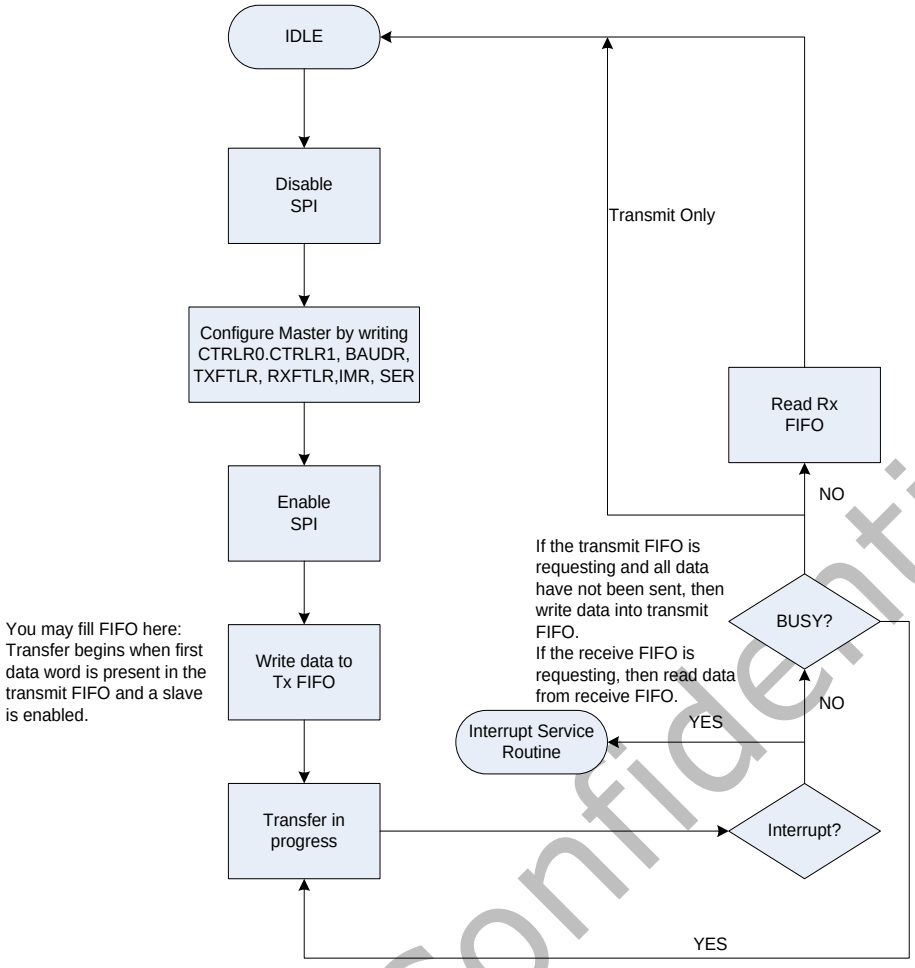


Fig. 23-7 SPI Master transfer flow diagram

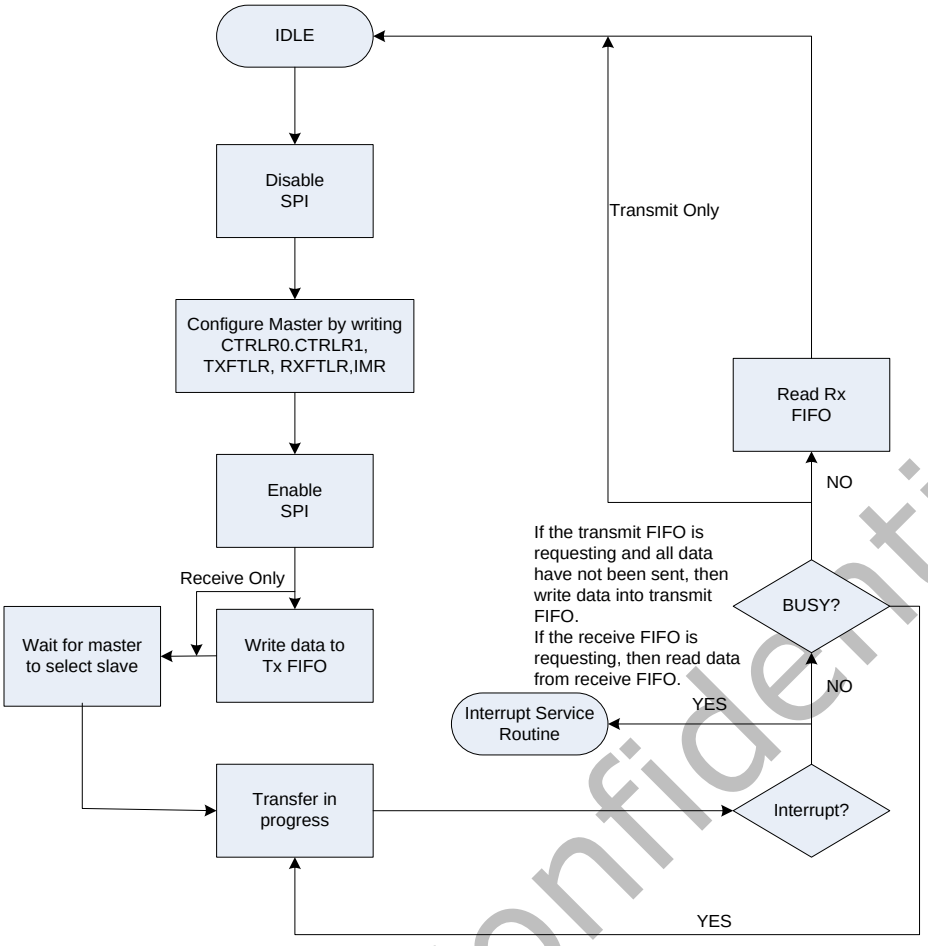


Fig. 23-8 SPI Slave transfe

Chapter 24 Real Time Clock (RTC)

24.1 Overview

The main function of a real-time clock (RTC) is to keep track of the current time of computer. The main characteristics of RTC are listed below:

- Supply voltage: 1.6v~3.3v.
- Support battery voltage detect.
- Support RTC external power off indication.
- Analog working current typical 2.5uA.
- Support 32.768 kHz clock self-detect.
- Provides year, month, day, weekday, hours, minutes, and seconds based on a 32.768 kHz crystal oscillator.
- Support compensation for the second and hour count.
- BCD representation of time, calendar and alarm.
- 12- or 24-hour clock with AM and PM in 12-hour mode.
- APB interface with 8bit data bus width.
- Bus-compatible interrupt signals (IRQ)
- Interrupts are separately software maskable
  - Alarm interrupt
  - Periodic interrupt
  - Chip power off interrupt
  - Battery power atypical interrupt

24.2 Block Diagram

Fig. 24-1 RTC Block Diagram

System Interface
The system interface implements the APB slave operation. Rtc_test and RTC use the common APBus address decoder, the common control bus and data bus.
rtc_test
rtc 32k clock test module, test the 32k clock accuracy.
rtc_analog
Battery power domain, support power and generate 32k clock for RTC digital, and some detect function, such as external power off and battery voltage atypical.
rtc_timer
Include npor, clock_gen, irq_gen, timer and reg_file, the main function is to keep track of the current time, alarm, and some control registers for RTC Analog.

24.3 Function Description

24.3.1 APB Interface

There is an APB slave interface in RTC. It is responsible for accessing registers.

24.3.2 Alarm interrupt

Alarm interrupt generate from the real-time, the alarm time, and the alarm_irq_mask bit.
alarm_irq_mask[0]: 1'b0: fix alarm second, 1'b1: every second.
alarm_irq_mask[1]: 1'b0: fix alarm minute, 1'b1: every minute.
alarm_irq_mask[2]: 1'b0: fix alarm hour, 1'b1: every hour.
alarm_irq_mask[3]: 1'b0: fix alarm day, 1'b1: every day.
alarm_irq_mask[4]: 1'b0: fix alarm month, 1'b1: every month.
alarm_irq_mask[5]: 1'b0: fix alarm year
when alarm_irq_mask = 6'h3, means every hour fix min and fix second alarm.
When alarm_irq_mask = 6'b101011, fix year, every month, fix day every hour fix min and fix second alarm.

24.3.3 RTC 32k clock test

Due to the analog generator 32k clock affected by temperature, voltage, clock precision need test with the environment change. In rtc test, use 24M clock as reference clock to measure the 32k clock. Before start test 32k clock, we should enable clk32k test(0x80), and configure test length, when rtc test done(0x84[2]), latch the 24M clock domain counter, and read out the counter from rtc_test registers(0x8c~0x98) via apb bus.

24.3.4 Compensation

In RTC digital design, we set three level compensation, the compensation value due to the RTC 32k clock test result, and if we need compensation, we need configure the compensation enable bit.

Compensation every hour, compensation at last minute every hour, and support add time and sub time by the MSB bit.

Compensation every day, compensation at last minute in last hour every day, and support add time and sub time by the MSB bit.

Compensation every month, compensation at last minute in last hour in last day every month, and support add time and sub time by the MSB bit.

24.4 Register Description

24.4.1 Internal Address Mapping

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

RTC digital register address space: 0x00~0x80

RTC test register address space: 0x84~0x98

No reset registers: 0x00~0x38, 0x50[2:1], 0x54, 0x58~0x60.

24.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
RTC_SET_SECONDS	0x0000	W	0x00000000	Seconds
RTC_SET_MINUTES	0x0004	W	0x00000000	Minutes
RTC_SET_HOURS	0x0008	W	0x00000000	Hours
RTC_SET_DAYS	0x000C	W	0x00000000	Days
RTC_SET_MONTHS	0x0010	W	0x00000000	Months
RTC_SET_YEARSL	0x0014	W	0x00000000	Years(LSB)
RTC_SET_YEARSH	0x0018	W	0x00000000	Years(MSB)
RTC_SET_WEEKS	0x001C	W	0x00000000	Weeks
RTC_ALARM_SECONDS	0x0020	W	0x00000000	Alarm seconds
RTC_ALARM_MINUTES	0x0024	W	0x00000000	Alarm minutes
RTC_ALARM_HOURS	0x0028	W	0x00000000	Alarm hours
RTC_ALARM_DAYS	0x002C	W	0x00000000	Alarm days
RTC_ALARM_MONTHS	0x0030	W	0x00000000	Alarm months
RTC_ALARM_YEARSL	0x0034	W	0x00000000	Alarm years(LSB)
RTC_ALARM_YEARSH	0x0038	W	0x00000000	Alarm years(MSB)
RTC_CTRL	0x003C	W	0x00000000	RTC controll registers
RTC_STATUS0	0x0040	W	0x00000000	RTC Statues registers
RTC_STATUS1	0x0044	W	0x00000000	RTC Statues registers
RTC_INT0_EN	0x0048	W	0x00000000	RTC interrupt enable registers
RTC_INT1_EN	0x004C	W	0x00000000	RTC interrupt enable registers
RTC_MSEC_CTRL	0x0050	W	0x00000000	millisecond control
RTC_MSEC_CNT	0x0054	W	0x00000000	millisecond periodic count
RTC_COMP_H	0x0058	W	0x00000000	Hour compensation registers
RTC_COMP_D	0x005C	W	0x00000000	Day compensation registers
RTC_COMP_M	0x0060	W	0x00000000	Month compensation registers
RTC_VREF_DET	0x0064	W	0x00000000	Vref trim bits
RTC_VREF_TRIM	0x0068	W	0x00000000	Vref trim bits
RTC_XO_TRIM0	0x006C	W	0x00000000	XO trim registers

Name	Offset	Size	Reset Value	Description
RTC_XO_TRIM1	0x0070	W	0x00000000	XO trim registers
RTC_TEST_SEL	0x0074	W	0x00000000	Analog test
RTC_LDO_CTRL	0x0078	W	0x00000000	LDO control
RTC_ANALOG_EN	0x007C	W	0x00000000	Analog config.
RTC_CLK32K_TEST	0x0080	W	0x00000000	Test 32k Hz clock enable
RTC_TEST_ST	0x0084	W	0x00000000	Test 32k clock start and status
RTC_TEST_LEN	0x0088	W	0x00000000	Test 32k clock length, unit second.
RTC_CNT_0	0x008C	W	0x00000000	24M clock counter[7:0]
RTC_CNT_1	0x0090	W	0x00000000	24M clock counter[15:8]
RTC_CNT_2	0x0094	W	0x00000000	24M clock counter[23:16]
RTC_CNT_3	0x0098	W	0x00000000	24M clock counter[31:24]

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

24.4.3 Detail Registers Description

RTC_SET_SECONDS

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	SEC1 Set the second digit of the RTC seconds (0-5).
3:0	RW	0x0	SEC0 Set the first digit of the RTC seconds (0-9).

RTC_SET_MINUTES

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	MIN1 Set the second digit of the RTC minutes (0-5).
3:0	RW	0x0	MIN0 Set the first digit of the RTC minutes (0-9).

RTC_SET_HOURS

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	AMPM Only used in PM-AM mode 1: PM 0: AM.
6	RO	0x0	reserved
5:4	RW	0x0	HOUR1 Set the second digit of the RTC hours.
3:0	RW	0x0	HOUR0 Set the first digit of the RTC hours.

RTC_SET_DAYS

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	DAY1 Set the second digit of the RTC days.

Bit	Attr	Reset Value	Description
3:0	RW	0x0	DAY0 Set the first digit of the RTC days.

RTC_SET_MONTHS

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	MONTH1 Set the second digit of the RTC months.
3:0	RW	0x0	MONTH0 Set the first digit of the RTC months.

RTC_SET_YEARSL

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	YEAR1 Set the second digit of the RTC years.
3:0	RW	0x0	YEAR0 Set the first digit of the RTC years.

RTC_SET_YEARSH

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	YEAR3 Set the 3th digit of the RTC years.
3:0	RW	0x0	YEAR2 Set the second digit of the RTC years.

RTC_SET_WEEKS

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2:0	RW	0x0	WEEK Set the second digit of the RTC weeks. configure value: 1~7.

RTC_ALARM_SECONDS

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	ALARM_SEC1 Set the second digit of the RTC alarm seconds.
3:0	RW	0x0	ALARM_SEC0 Set the first digit of the RTC alarm seconds.

RTC_ALARM_MINUTES

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	ALARM_MIN1 Set the second digit of the RTC alarm minutes.
3:0	RW	0x0	ALARM_MIN0 Set the first digit of the RTC alarm minutes.

RTC_ALARM_HOURS

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	ALARM_PM_AM Set alarm PM or AM: only used in PM-AM mode. 1: PM 0: AM.
6	RO	0x0	reserved
5:4	RW	0x0	ALARM_HOUR1 Set the second digit of the RTC alarm hours.
3:0	RW	0x0	ALARM_HOUR0 Set the first digit of the RTC alarm hours.

RTC_ALARM_DAYS

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5:4	RW	0x0	ALARM_DAY1 Set the second digit of the RTC alarm days.
3:0	RW	0x0	ALARM_DAY0 Set the first digit of the RTC alarm days.

RTC_ALARM_MONTHS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RW	0x0	ALARM_MONTH1 Set the second digit of the RTC alarm months.
3:0	RW	0x0	ALARM_MONTH0 Set the first digit of the RTC alarm months.

RTC_ALARM_YEARSL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	ALARM_YEAR1 Set the second digit of the RTC alarm years.
3:0	RW	0x0	ALARM_YEAR0 Set the first digit of the RTC alarm years.

RTC_ALARM_YEARSH

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	ALARM_YEAR3 Set the 3th digit of the RTC alarm years.
3:0	RW	0x0	ALARM_YEAR2 Set the second digit of the RTC alarm years.

RTC_CTRL

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	RTC_READ_SEL Reserved
6	RW	0x0	GET_TIME Reserved
5:4	RO	0x0	reserved
3	RW	0x0	AMPM_MODE AMPM mode set. 0: 24 hours mode 1: 12 hours mode.
2	RW	0x0	COMP_EN Auto compensation enabile bit. 0: No auto compensation 1: Auto compensation enabled.
1	RW	0x0	PSEL_SEL Psel select bit. 1'b0: psel 1'b1: psel & sys_npor
0	RW	0x0	START_RTC RTC trigger bit. 0: RTC stop 1: RTC start. Note: RTC_time(SET_SECONDS/MINUTES/HOURS/DAYS/MONTHS/YEARS) can only be changed during RTC stop.

RTC_STATUS0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6	RW	0x0	MSEC_ST milisec periodic interrupt occurred.
5	RO	0x0	reserved
4	RW	0x0	ALARM Indicates that an alarm interrupt has been generated.
3	RW	0x0	EVENT_1D One day has occurred.
2	RW	0x0	EVENT_1H One hour has occurred.
1	RW	0x0	EVENT_1M One minutes has occurred.
0	RW	0x0	EVENT_1S One second has occurred.

RTC_STATUS1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RO	0x0	EXT_OFF RTC external power detect status. 0: external power on 1: external power off.
2	RO	0x0	BAT_DET Battery voltage detect status. 0: battery voltage in typical range 1: battery voltage lower.

Bit	Attr	Reset Value	Description
1	RO	0x0	SYS_INT Reserved
0	RO	0x0	RUN_ST 0: RTC is frozen. 1: RTC is running. Note: This bit shows the real state of the RTC.

RTC_INT0_EN

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	ALARM_IRQ_EN Alarm interrupt output enbale bit. 1: enable 0: disable. include external power on or off.
6	RW	0x0	ALARM_IRQ_EN_PD Alarm interrupt output enbale bit(only asserted when external power down). 1: enable 0: disable. not need config ALARM_IRQ_EN.
5:0	RW	0x00	ALARM_IRQ_MASK Alarm irq mask: alarm irq mask[0]: second alarm alarm irq mask[1]: minute alarm alarm irq mask[2]: hour alarm alarm irq mask[3]: day alarm alarm irq mask[4]: month alarm alarm irq mask[5]: year alarm if mask bit equal 1'b1, means that fix point, if mask bit equal 1'b0, means every. example: if configure ALARM_IRQ_MASK = 6'h01, configure RTC_ALARM_SECONDS = 8'h03, every minute, if second == 8'h03, alarm active; if configure ALARM_IRQ_MASK = 6'h17, configure RTC_ALARM_SECONDS = 8'h30: configure RTC_ALARM_MINUTES = 8'h30: configure RTC_ALARM_HOURS = 8'h8(24hour mode): configure RTC_ALARM_MONTH=8'h2. every day in February, when Am 8:30:30(H:M:S), alarm active.

RTC_INT1_EN

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	MSEC_IRQ_EN Milisecond interrupt output enable bit. 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
6	RW	0x0	EXT_IRQ_EN External power off interrupt output enable bit. 0: disable 1: Enable.
5	RW	0x0	BAT_IRQ_EN Detect battery voltage lower interrupt output enable bit. 0: Disable 1: Enable.
4	RW	0x0	SYS_IRQ_MASK System interrupt mask bit. 0: Normal mode, no system interrupt masked 1: system interrupt masked, no system interrupt output.
3	RW	0x0	PER_PD_MASK_EN Periodic interrupt output enable bit, when RTC external power down. 1: Mask periodic interrupt while the external power down 0: Normal mode, no periodic interrupt masked when external power down Need config PER_TIMER_EN enable.
2	RW	0x0	PER_TIMER_EN Periodic interrupt output enable bit. 0: Disable 1: Enable.
1:0	RW	0x0	EVERY_TYPE_SEL Periodic interrupt type select. 2'b00: Every second 2'b01: Every minute 2'b10: Every hour 2'b11: Every day.

RTC_MSEC_CTRL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2:1	RW	0x0	MSEC_SEL Milisec periodic clock select. 00b: 8hz 01b: 128hz 10b: 1khz 11b: 32khz.
0	RW	0x0	MSEC_EN Milisec periodic function enable.

RTC_MSEC_CNT

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	MSEC_CNT Number of milisec periodic interrupt interval clock.

RTC_COMP_H

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	COMP_H_DIR Compensation direction every hour. 1: Add second 0: Sub second.
6:0	RW	0x00	COMP_HOUR This register contains the number of second periods to be added/sub every hour. the 7bit value need 8421BCD code.

RTC_COMP_D

Address: Operational Base + offset (0x005C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	COMP_D_DIR Compensation direction every day. 1: Add second 0: Sub second.
6:0	RW	0x00	COMP_DAY This register contains the number of second periods to be added/sub every day. the 7bit value need 8421BCD code.

RTC_COMP_M

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7	RW	0x0	COMP_M_DIR Compensation direction every month. 1: Add second 0: Sub second.
6:0	RW	0x00	COMP_MONTH This register contains the number of second periods to be added/sub every month. the 7bit value need 8421BCD code.

RTC_VREF_DET

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2:0	RW	0x0	D2A_TRIM_VREF_DET Vref set, for battery voltage detect.

RTC_VREF_TRIM

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	D2A_TRIM_VREF_1P0 Vref set, for xo ldo output voltage.
3	RO	0x0	reserved
2:0	RW	0x0	D2A_TRIM_VREF_DIG Vref set, for rtc digital voltage.

RTC_XO_TRIM0

Address: Operational Base + offset (0x006C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	D2A_XO_OTA Reserved
3:1	RW	0x0	D2A_XO_CAP Capacitance compensation for xo load. 3'h0: 0pf 3'h1: 1pf 3'h2: 2pf 3'h3: 3pf 3'h4: 4pf 3'h5: 5pf 3'h6: 6pf 3'h7: 7pf
0	RW	0x0	D2A_XO_EN XO enable bit. 0: Disable 1: Enable configure 1'b1 for rtc analog 32k generate.

RTC_XO_TRIM1

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:6	RW	0x0	D2A_XO_GM Reserved
5:3	RW	0x0	D2A_XO_START_UP Reserved
2:0	RW	0x0	D2A_XO_START Reserved

RTC_TEST_SEL

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:4	RW	0x0	D2A_LDO_XO_SEL Calibrate LDO output voltage for xo.
3:0	RW	0x0	D2A_TEST_SEL Reserved

RTC_LDO_CTRL

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:7	RO	0x0000000	reserved
6:4	RW	0x0	D2A_RESERVE D2A_RESERVE[0]: d2a_por_ref_sel[1] D2A_RESERVE[1]: clk_32k_h gating enable bit D2A_RESERVE[2]: reserved
3	RO	0x0	reserved
2:0	RW	0x0	D2A_DIG_LDO_VREF Calibrate LDO output voltage for rtc digital.

RTC_ANALOG_EN

Address: Operational Base + offset (0x007C)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	D2A_CLK_OUT_EN XO clock output enable bit. 0: Disable 1: Enable. configure d2a_clk_out_en 1'b1 after d2a_xo_en, about 500ms.
4	RW	0x0	D2A_BAT_DET_EN battery voltage detect enable bit. 0: Disable 1: Enable.
3:2	RW	0x0	D2A_VDET_CUR_SEL Reserved
1	RW	0x0	D2A_POR_REF_SEL Reserved
0	RW	0x0	D2A_CLK_BYPS_EN 32K clock input from XIN pin. 0: Disable 1: Enable.

RTC_CLK32K_TEST

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RW	0x0	CLK_TEST_EN 32k clock output enable bit. 0: Disable 1: Enable.

RTC_TEST_ST

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:3	RO	0x00000000	reserved
2	RO	0x0	TEST_DONE 32K clock test status. 0: Test idle or in test. 1: Test done.
1	RO	0x0	TEST_START_ST Test clk32k status. 0: Idle 1: In test.
0	RW	0x0	TEST_START Test clk32k start. 0: Stop 1: Start.

RTC_TEST_LEN

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	TEST_LENGTH Test length, unit second.

RTC_CNT_0

Address: Operational Base + offset (0x008C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	CLK24M_CNT_0 Count[7:0]

RTC_CNT_1

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	CLK24M_CNT_1 Count[15:8]

RTC_CNT_2

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	CLK24M_CNT_2 Count[23:16]

RTC_CNT_3

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	RW	0x00	CLK24M_CNT_3 Count[31:24]

24.5 Interface Description

Table 24-1 RTC Interface Description

Pin Name	Direction	Description
pclk	input	APB2.0, use the common bus signal with rtc_test.
psel	input	APB2.0, use the common bus signal with rtc_test.
penable	input	APB2.0, use the common bus signal with rtc_test.
pwrite	input	APB2.0, use the common bus signal with rtc_test.
paddr[7:0]	input	APB2.0, use the common bus signal with rtc_test.
pwdata[7:0]	input	APB2.0, use the common bus signal with rtc_test.
prdata[7:0]	output	APB2.0, To rtc_test block.
clk24M	input	24M clock.
clk_32k_o	output	32k clock output.
clk_32k_H	output	32k clock output.
rtc_intr	output	High active, interrupt from RTC.
dig_ldo_rdy_o	output	High active.
external_off_o	output	Low active, used to iso cell enable.

24.6 Application Notes

1) APB configuration operation:
   pwdata[31:8] == 24'hc45229, pwdata[7:0] store the configure value.
2) If enable AMPM mode, AMPM mode need set before real time set.
3) If update real time, need stop RTC first, and then set real time registers and start RTC.
4) Rtc work sequence:
   Wait ldo rdy;
   APB cfg D2A_XO_EN: 0x6c[0] = 1
   Wait 200ms for xo 32k clock stable;
   APB cfg D2A_CLK_OUT_EN: 0x7c[5] = 1

APB cfg real time: 0x00~0x1c;
APB cfg alarm time: 0x20~0x38;
APB cfg interrupt enable: 0x48, 0x4c;
APB cfg millisecond register: 0x50, 0x54;
APB cfg compensation register: 0x58, 0x5c, 0x60;
Set START_RTC, 0x3c[0] = 1
Read RUN_ST, 0x44[0], check rtc status.

5) Read time: Read time from: 0x00~0x1c
   Due to shadow registers delete, read time maybe not all right, need read at least three times: time0, time1, time2(need in 1 second).
   If time0, time1, time2 all different, the right time is the time2;
   If time0, time1, time2 have two or three the same, the right time is the same time.
6) Stop RTC: APB cfg 0x3c[0]=0; read RUN_ST check RTC stop.

Chapter 25 DECOM

25.1 Overview

DECOM can decompress compressed files in GZIP, DEFALTE and ZLIB formats. DECOM controller supports the following features:

- Support for decompressing GZIP files
- Support for decompressing data in DEFLATE format
- Support for decompressing data in ZLIB format
- Not support for decompressing LZ4 files
- There is a 32bit APB slave interface for configuring decompression parameters and querying register status
- There is a 128-bit AXI master interface for reading compressed data and outputting decompressed data. The AXI master interface supports burst 4/8/16 and single transmissions
- Support one internal 128-bit wide and 64-location deep FIFO(RX_FIFO) for caching source compressed data
- Support one internal 24-bit wide and 64-location deep FIFO(HF_FIFO) for caching intermediate data during decompression
- Built-in a 8KB two-port RAM for storing decompressed data
- Support complete interrupt and error interrupt output
- Support the limit_size function of the decompressed data to prevent the memory from being maliciously destroyed during the decompression process
- Support software to stop the decompression process

25.2 Block Diagram

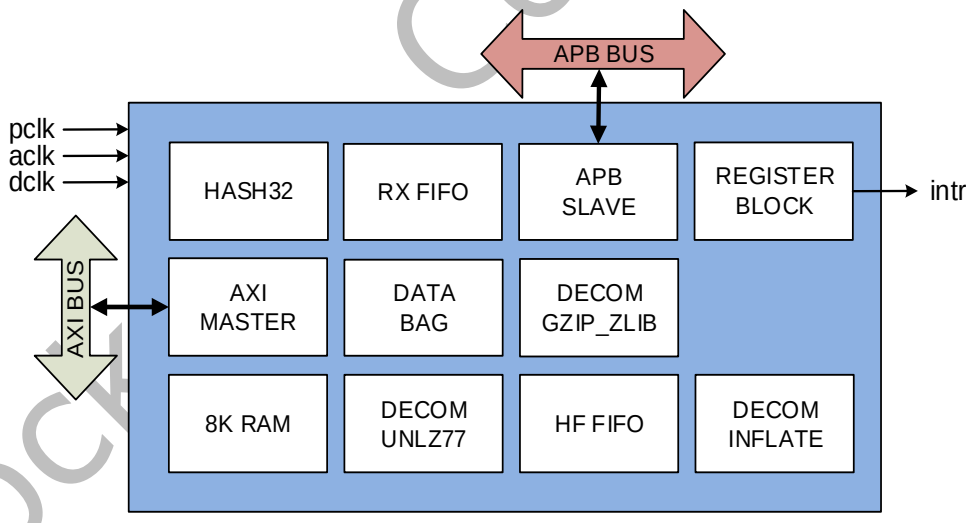


Fig. 25-1 DECOM Block Diagram

25.3 Function Description

25.3.1 Module Introduction

25.3.1.1 APB INTERFACE

The APB slave interface is used for registers configuration, decompression file parameter reading and interrupt status query, etc.

25.3.1.2 AXI MASTER INTERFACE

The AXI master interface is used to read compressed file data from external storage into the RX_FIFO and output the decompressed data to external storage.

25.3.1.3 RX_FIFO

The RX_FIFO is used to buffer the source compressed file data read back through the AXI master interface. The RX_FIFO is an asynchronous FIFO, the write interface uses aclk, and the read interface uses dclk.

25.3.1.4 Data Bag

The Data Bag reads and bites the data in the RX_FIFO, and extracts the corresponding number of bits according to the needs of the decompression module.

25.3.1.5 GZIP_ZLIB

The GZIP_ZLIB module is used to perform file header and file end parsing for compressed files in GZIP format and compressed data in ZLIB format, and the compressed data portion is decompressed by the INFLATE module. The GZIP or ZLIB decompression mode can be configured by configuring the CTRL register.

25.3.1.6 INFLATE Module

INFLATE Module is used to decompress compressed files in DEFLATE format, implement static/dynamic huffman decoding, decompress compressed data in DEFLATE format into LZ77 format (match length-match distance) data pairs or character data, and store them in HF_FIFO.

25.3.1.7 HF_FIFO

The HF_FIFO is used to buffer the LZ77 format (match length-match distance) data pair or character data decompressed by the INFLATE module, and wait for the UNLZ77 module to further decompress the intermediate data. The HF_FIFO is an asynchronous FIFO, the write interface uses dclk, and the read interface uses aclk.

25.3.1.8 UNLZ77 Module

UNLZ77 is used to implement LZ77 decoding and store the decompressed data in 8K RAM. When the matching distance is less than 8k byte, matching replication is performed in 8K RAM. When the matching distance is greater than 8k byte, the AXI master interface is controlled to perform matching replication in the external storage.

25.3.1.9 8K-RAM

The 8K RAM is used to buffer the decompressed data. During the decompression process, the UNLZ77 module also performs read access to the RAM to complete the data matching.

25.3.1.10 Interface and Clock

The 32-bit APB Slave interface in DECOM is used for register configuration and interrupt status query interface, using pclk as the interface clock.

DECOM uses 128bit AXI master interface to realize the functions of reading the data to be compressed, decompressing the data output and reading external matching data; it should be noted that the AXI master interface supports Single and Burst transmission, which only supports incr4, incr8 and Incr16 three transmission types, does not support Wrap transmission. DECOM automatically configures the burst type according to the length of the transmitted data and the starting address. The software configuration is not supported to select the burst type.

DECOM has three sets of clock and reset inputs, pclk and presetn, dclk and drstn, aclk and aresetn. Among them pclk and aclk are APB and AXI bus clocks respectively, and dclk is decompression clock.

25.3.2 Data Flow

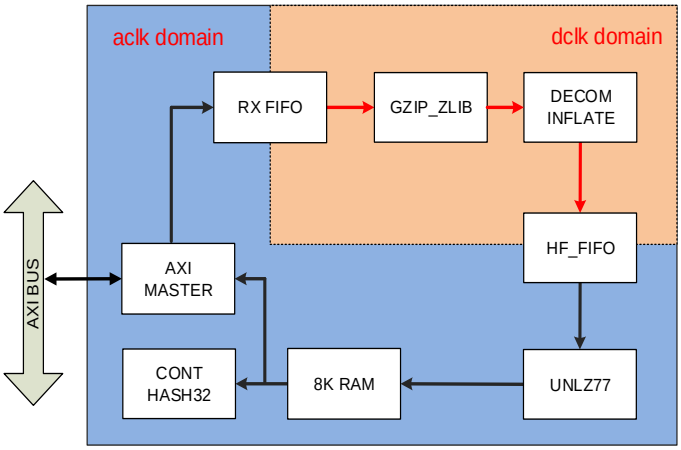


Fig. 25-2 DECOM Data Flow

25.4 Register Description

25.4.1 Internal Address Mapping

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

25.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
DECOM_CTRL	0x0000	W	0x00000000	Control Register
DECOM_ENR	0x0004	W	0x00000000	Enable Register
DECOM_RADDR	0x0008	W	0x00000000	Read Address
DECOM_WADDR	0x000C	W	0x00000000	Write Address
DECOM_UDDSL	0x0010	W	0xFFFFFFFF	Undecompressed Data Size
DECOM_UDDSH	0x0014	W	0xFFFFFFFF	Undecompressed Data Size
DECOM_TXTHR	0x0018	W	0x00000100	Transmit Threshold Level
DECOM_SLEN	0x0020	W	0x00000000	INFLATE Store Length
DECOM_STAT	0x0024	W	0x00000000	DECOM Status Register
DECOM_ISR	0x0028	W	0x00000000	Interrupt Status Register
DECOM_IEN	0x002C	W	0x00000000	Interrupt Enable Register
DECOM_AXI_STAT	0x0030	W	0x00000010	AXI Master Interface State
DECOM_TSIZEL	0x0034	W	0x00000000	Decompressed Data Total Size Lower 32bit
DECOM_TSIZEH	0x0038	W	0x00000000	Decompressed Data Total Size Upper 32bit
DECOM_MGNUM	0x003C	W	0x00000000	LZ4 File Magic Number
DECOM_FRAME	0x0040	W	0x00000000	LZ4 File Frame Descriptor
DECOM_DICTID	0x0044	W	0x00000000	Dictionary ID
DECOM_CSL	0x0048	W	0x00000000	LZ4 Content Size(CS) Lower 32-bits
DECOM_CSH	0x004C	W	0x00000000	LZ4 Content Size(CS) Upper 32-bits
DECOM_LMTSL	0x0050	W	0xFFFFFFFF	Limit Size of Decompressed Data
DECOM_LMTSH	0x0054	W	0xFFFFFFFF	Limit Size of Decompressed Data
DECOM_GZFHD	0x0058	W	0x00000000	GZIP/ZLIB File Header Information
DECOM_VERSION	0x00F0	W	0x20211216	DECOM Version Number

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

25.4.3 Detail Registers Description

DECOM_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:6	RO	0x0000000	reserved
5	RW	0x0	ZLIBM ZLIB mode. 1'b0: Disable 1'b1: Enable When decompressing ZLIB compressed data, ZLIBM must be enabled and DEM should be set to 1.
4	RW	0x0	GZIPM GZIP mode. 1'b0: Disable 1'b1: Enable When decompressing GZIP files, GZIPM must be enabled and DEM should be set to 1.
3	RW	0x0	CCEN LZ4 content checksum check enable. 1'b0: Disable 1'b1: Enable In LZ4 files, a 32-bits content checksum will be appended at the end of the file. When CCEN is enabled, DECOM will verify the content checksum to check if the decompressed data is correct. Only valid when decompressing LZ4 files.
2	RW	0x0	BCEN LZ4 block checksum check enable. Not used. 1'b0: Disable 1'b1: Enable In LZ4 files, each data block will be followed by a 4-bytes checksum, calculated by using the xxHash-32 algorithm on the compressed data block. The intention is to detect data corruption(storage or transmission errors) immediately, before decoding. When BCEN is enabled, DECOM will check the block checksum to determine if the block to be decompressed is correct. Only valid when decompressing LZ4 files.
1	RW	0x0	HCEN LZ4 header checksum check enable. Not used. 1'b0: Disable 1'b1: Enable The LZ4 header contains a 1-byte xxh32 checksum value(HC). When HCEN is enabled, DECOM will check the HC to check if the header is correct. Only valid when decompressing LZ4 files.
0	RW	0x0	DEM DECOM mode select. 1'b0: UNLZ4 mode. Decompress the file in LZ4 format. 1'b1: INFLATE mode. Decompress the file in INFLATE format. If GZIPM=1'b1 or GZLIBM=1'b1, DEM must be set 1.

DECOM_ENR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved

Bit	Attr	Reset Value	Description
0	R/WSC	0x0	ENR Enables and disables all decompress operations. 1'b0: Disable decompressor 1'b1: Enable decompressor to work All FIFO buffers and aclk/dclk domain work registers are cleared when the device is disabled. Self clear when decompression is complete. The ENR should be enabled after the other registers are configured.

DECOM_RADDR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	RADDR Read address. This register is used to configure the starting address of the file to be decompressed.

DECOM_WADDR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	WADDR Write address. This register is used to configure the starting address of the location where the decompressed data is written. Note: WADDR[31:0] must be configured with a 128bit aligned address (WADDR[3:0]==4'b0).

DECOM_UDDSL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	UDDSL Undecompressed data size lower 32bits. (Unit:byte) UDDS[63:0] is the total size of undecompressed file. DECOM will read the compressed file data according to the value of UDDS[63:0], but the configuration of this register is optional. If this register is not configured, DECOM will use the default value of 64'hffff_ffff_ffff_ffff, and DECOM will keep reading data until the current decompression process ends.

DECOM_UDDSH

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	UDDSH Undecompressed data size higher 32bits.

DECOM_TXTHR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000100	TXTHR When the number of 8K_MEM entries is greater than or equal to this value, the DECOM will automatically transfer the data in the 8K_MEM to the external via the AXI Master interface. The value ranges is 0~7680.(Unit:byte)

DECOM_SLEN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	SLEN INFLATE store block size. (Unit:byte) Indicate the block size when decompressing the store block of INFALTE file. For debugging SLC_ERR.

DECOM_STAT

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:2	RO	0x00000000	reserved
1	RO	0x0	LAST Used to indicate whether the currently processed compressed data block is the last one. 1'b0: Not last block 1'b1: Last block
0	RO	0x0	COMPLETE DECOM complete flag. 1'b0: Not complete 1'b1: Decompress is complete with no error Only decompress total complete with no error, this flag is set to 1.

DECOM_ISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19	W1C	0x0	DSOLI Decompressed data size over limit_size interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
18	W1C	0x0	ZDICTEI ZLIB dictionary error interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
17	W1C	0x0	GCMEI GZIP/ZLIB compression method check error interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
16	W1C	0x0	GIDEI GZIP ID error interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
15	W1C	0x0	CCCEI UNLZ4 content checksum check error interrupt. 1'b0: ccc_err interrupt is not active 1'b1: ccc_err interrupt is active
14	W1C	0x0	BCCEI UNLZ4 block checksum check error interrupt. 1'b0: bcc_err interrupt is not active 1'b1: bcc_err interrupt is active
13	W1C	0x0	HCCEI UNLZ4 header checksum check error interrupt. 1'b0: hcc_err interrupt is not active 1'b1: hcc_err interrupt is active

Bit	Attr	Reset Value	Description
12	W1C	0x0	CSEI UNLZ4 content size (or GZIP isize) error interrupt. 1'b0: cs_err interrupt is not active 1'b1: cs_err interrupt is active
11	W1C	0x0	DICTEI UNLZ4 dictionary error interrupt. 1'b0: Interrupt is not active 1'b1: Interrupt is active
10	W1C	0x0	VNEI UNLZ4 version number error interrupt. 1'b0: vn_err interrupt is not active 1'b1: vn_err interrupt is active
9	W1C	0x0	MNEI UNLZ4 magic number error interrupt. 1'b0: mn_err interrupt is not active 1'b1: mn_err interrupt is active
8	W1C	0x0	RDCEI AXI read channel error interrupt. 1'b0: INFLATE AXI RDC_ERR interrupt is not active 1'b1: INFLATE AXI RDC_ERR interrupt is active
7	W1C	0x0	WRCEI AXI write channel error interrupt. 1'b0: INFLATE AXI WRC_ERR interrupt is not active 1'b1: INFLATE AXI WRC_ERR interrupt is active
6	W1C	0x0	DISEI INFLATE huffman distance error interrupt. 1'b0: INFLATE HFDIS_ERR interrupt is not active 1'b1: INFLATE HFDIS_ERR interrupt is active
5	W1C	0x0	LENEI INFLATE huffman length error interrupt. 1'b0: INFLATE HFLEN_ERR interrupt is not active 1'b1: INFLATE HFLEN_ERR interrupt is active
4	W1C	0x0	LITEI INFLATE huffman literal error interrupt. 1'b0: INFLATE HFLIT_ERR interrupt is not active 1'b1: INFLATE HFLIT_ERR interrupt is active
3	W1C	0x0	SQMEI INFLATE SQ match error interrupt. 1'b0: INFLATE SQM_ERR interrupt is not active 1'b1: INFLATE SQM_ERR interrupt is active
2	W1C	0x0	SLCEI INFLATE store block length check error interrupt. 1'b0: INFLATE SLC_ERR interrupt is not active 1'b1: INFLATE SLC_ERR interrupt is active
1	W1C	0x0	HDEI INFLATE file header error interrupt. 1'b0: INFLATE header error interrupt is not active 1'b1: INFLATE header error interrupt is active
0	W1C	0x0	DSI DECOM stop interrupt. 1'b0: Decompression stop interrupt is not active 1'b1: Decompression stop interrupt is active This interrupt indicates that DECOM has stopped working. Including decompression completion or decompression encountered an error.

DECOM_IEN

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19	RW	0x0	DSOLIEN Decompressed data size over limit_size interrupt enable. 1'b0: Disable 1'b1: Enable
18	RW	0x0	ZDICTEIEN ZLIB dictionary error interrupt enable. 1'b0: Disable 1'b1: Enable
17	RW	0x0	GCMEIEN GZIP/ZLIB compression method error interrupt enable. 1'b0: Disable 1'b1: Enable
16	RW	0x0	GIDEIEN GZIP ID error interrupt enable. 1'b0: Disable 1'b1: Enable
15	RW	0x0	CCCEIEN UNLZ4 content checksum check error interrupt enable. 1'b0: Disable 1'b1: Enable
14	RW	0x0	BCCEIEN UNLZ4 block checksum check error interrupt enable. 1'b0: Disable 1'b1: Enable
13	RW	0x0	HCCEIEN UNLZ4 header checksum check error interrupt enable. 1'b0: Disable 1'b1: Enable
12	RW	0x0	CSEIEN UNLZ4 content size (or GZIP isize) error interrupt enable. 1'b0: Disable 1'b1: Enable
11	RW	0x0	DICTEIEN UNLZ4 dictionary error interrupt enable. 1'b0: Disable 1'b1: Enable
10	RW	0x0	VNEIEN UNLZ4 version number error interrupt enable. 1'b0: Disable 1'b1: Enable
9	RW	0x0	WNEIEN UNLZ4 magic number error interrupt enable. 1'b0: Disable 1'b1: Enable
8	RW	0x0	RDCEIEN AXI read channel error interrupt enable. 1'b0: Disable 1'b1: Enable
7	RW	0x0	WRCEIEN AXI write channel error interrupt enable. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
6	RW	0x0	DISEIEN INFLATE huffman distance error interrupt enable. 1'b0: Disable 1'b1: Enable
5	RW	0x0	LENEIEN INFLATE huffman length error interrupt enable. 1'b0: Disable 1'b1: Enable
4	RW	0x0	LITEIEN INFLATE huffman literal error interrupt enable. 1'b0: Disable 1'b1: Enable
3	RW	0x0	SQMEIEN INFLATE SQ match error interrupt enable. 1'b0: Disable 1'b1: Enable
2	RW	0x0	SLCIEN INFLATE store block length check error interrupt enable. 1'b0: Disable 1'b1: Enable
1	RW	0x0	HDEIEN INFLATE file header error interrupt enable. 1'b0: Disable 1'b1: Enable
0	RW	0x0	DSIEN DECOM stop interrupt enable. 1'b0: Disable 1'b1: Enable

DECOM_AXI_STAT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4	RO	0x1	AXI_IDLE AXI master idle state register. 1'b0: AXI master is busy 1'b1: AXI master is idle When the decompression is aborted, DECOM will wait for AXI master's current read/write transfer to complete, that is, wait for AXI_IDLE=1, then reset DECOM, otherwise the uncompleted AXI transmission will cause AXI bus exception. It should be noted that after the decompression is abnormal, the AXI master's unfinished write operation will continue, but the output value of wstrb[15:0] will become 16'b0.
3:2	RO	0x0	RRESP AXI read channel response state.
1:0	RO	0x0	BRESP AXI write channel response state.

DECOM_TSIZEL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TSIZEL The total size of the data after decompression. TSIZE[63:0]. (Unit:byte)

DECOM_TSIZEH

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TSIZEH The total size of the data after decompression.

DECOM_MGNUM

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	MGNUM Magic number of LZ4 is 0x184D2204. If the MGNUM is not 0x184D2204, a magic number error(mn_err) will be generated. And DECOM will stop decompressing.

DECOM_FRAME

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23:16	RO	0x00	HCC Header checksum byte. One-byte checksum of combined descriptor fields, including optional ones. The value is the second byte of xxh32() using zero as a see, and the full frame descriptor as an input (including optional fields when they are present). A wrong checksum indicates an error in the descriptor. Header checksum is informational and can be skipped. The header checksum check function can be enabled by controlling the HCEN enable signal in the CTRL register. After being turned on, DECOM calculates the hash32 check value based on the data of the frame descriptor part of the LZ4 compressed file, and performs the header checksum byte in the file. In contrast, if they are not equal, an HCC error will be generated and an interrupt will be generated.
15:8	RO	0x00	BD BD byte. Including Block Maximum Size information. This information is useful to help the decoder allocate memory. Size here refers to the original(uncompressed) data size.
7:6	RO	0x0	VN_NUM Version number. 2-bits field, must be set to 2'b01. Any other value cannot be decoded by this version of the specification. If the version number is error, Version number error(VNE) will be generated.
5	RO	0x0	BCC_FLG Block checksum flag. If this flag is set, each data block will be followed by a 4-bytes checksum, calculated by using the xxHash-32 algorithm on the raw (compressed) data block. The intention is to detect data corruption (storage or transmission errors) immediately, before decoding. Block checksum usage is optional.

Bit	Attr	Reset Value	Description
4	RO	0x0	BIND_FLG Block independence flag. If this flag is set to "1", blocks are independent. If this flag is set to "0", each block depends on previous ones (up to LZ4 window size, which is 64 KB). In such case, it's necessary to decode all blocks in sequence. Block dependency improves compression ratio, especially for small blocks. On the other hand, it makes random access or multi-threaded decoding impossible. For debugging.
3	RO	0x0	CS_FLG Content size flag. If this flag is set, the uncompressed size of data included within the frame will be present as an 8 bytes unsigned little endian value, after the flags. Content size usage is optional.
2	RO	0x0	CCC_FLG Content checksum flag. If this flag is set, a 32-bits content checksum will be appended after the EndMark.
1	RO	0x0	reserved
0	RO	0x0	DICT_FLG Dictionary ID flag. If this flag is set, a 4-bytes Dict-ID field will be present, after the descriptor flags and the Content Size. If this flag is set, Dictionary ID error will be generated. DECOM will stop decompressing.

DECOM_DICTID

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DICTID Dictionary ID in the compressed file header. Dictionary ID is only present if the DID_FLG is set. It works as a kind of "known prefix" which is used by both the compressor and the decompressor to "warm-up" reference tables.

DECOM_CSL

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CSL Content size(CS) lower 32-bits. (Unit:byte) CS[63:0] is the original(uncompressed) size. This information is optional in LZ4 file header, and only present if CS_FLG is set. CS[63:0]={CSH, CSL};

DECOM_CSH

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	CSH Content size(CS) upper 32-bits.

DECOM_LMTSL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	LMTSL Limit size of decompressed data lower 32bits. (Unit:byte) When the amount of decompressed data is greater than the LMTS, DSOLI interrupt is generated and the decompression process is stopped.

DECOM_LMTSH

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	LMTSH Limit size of decompressed data higher 32bits.

DECOM_GZFHD

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	GZFHD GZIP/ZLIB file header information. When GZIPM = 1, GZFHD is GZIP file header information. When ZLIBM = 1, GZFHD is ZLIB file header information.

DECOM_VERSION

Address: Operational Base + offset (0x00F0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000926	VERSION Version number = 32'h2021_1216.

25.5 Application Notes

- After configuring the decompression mode and other decompression parameters, configure the decompression enable register ENR
- When the decompression is not completed, setting ENR from 1 to 0 will force the decompression process to stop and DECOM will return to the IDLE state. Need to reconfigure and enable DECOM to start the next decompression
- When an error is encountered in the decompression, DECOM will immediately stop the decompression process, set the relevant status register, and generate the corresponding interrupt
- After the decompression process is stopped, ENR will automatically change to 0
- After the decompression process is stopped, DSI will be set to 1. If DSIEN is 1, an interrupt will be generated; if DSIEN is 0, no interrupt will be generated
- After the decompression process is stopped, COMPLETE (STAT[0]) is not set to 1 if there is an error in the decompression process; COMPLETE is set to 1 only when the decompression is complete and there are no errors

Chapter 26 Video Capture (VICAP)

26.1 Overview

The Video Capture, receives the data from Camera via DVP/MIPI/LVDS, and transfers the data into system main memory by AXI bus.

- Support BT601 YCbCr 422 8bit input, RAW 8/10/12bit input
- Support BT656 YCbCr 422 8bit progressive/interlace input
- Support BT1120 YCbCr 422 16bit progressive/interlace input, single/dual-edge sampling
- Support YUYV sequence configurable
- Support the polarity of hsync and vsync configurable
- Support receiving two interfaces of MIPI CSI/MIPI DSI/LVDS, up to four IDs for each interface
- Support five CSI data formats: RAW8/10/12/14, YUV422
- Support three modes of MIPI CSI HDR: virtual channel mode, identification code mode, line counter mode
- Support four LVDS data formats: RAW8/10/12, YUV422
- Support reducing frame rate
- Support window cropping
- Support RAW data through to ISP
- Support 8/16/32 times down-sampling for RAW data
- Support virtual stride when write to DDR
- Support NV16/NV12/YUV400/YUYV output format for YUV data
- Support compact/non-compact output format for RAW data
- Support DMA burst gather
- Support soft reset, auto-reset when DMA error.

26.2 Block Diagram

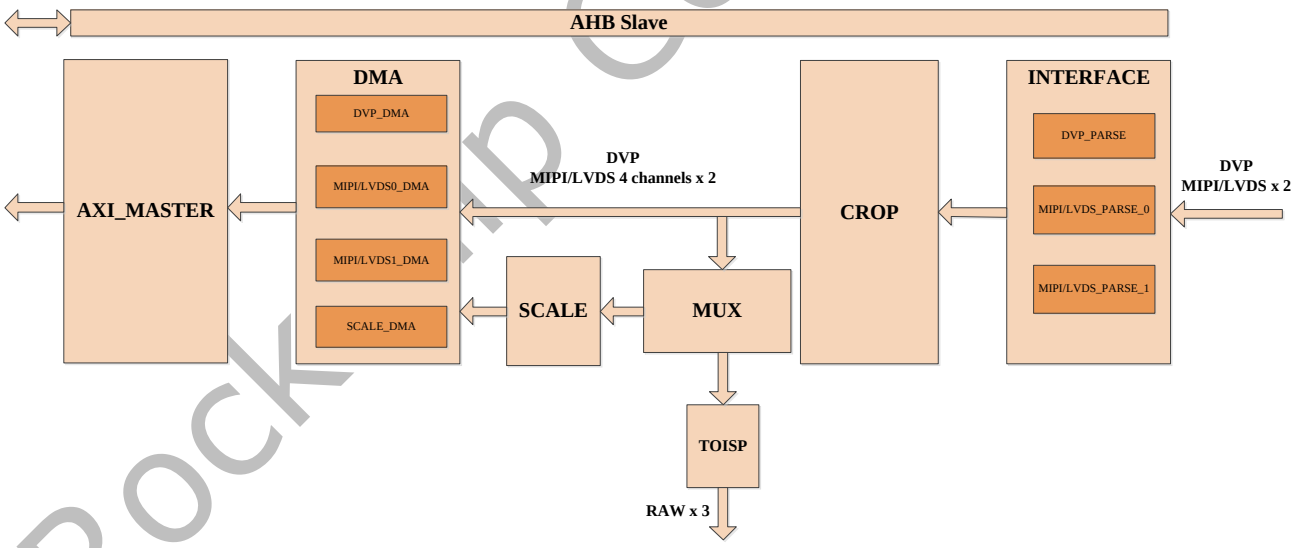


Fig. 26-1 VICAP Block Diagram

VICAP comprises with:

- AHB Slave
- AXI Master
- INTERFACE
- CROP
- SCALE
- TOISP
- DMA

26.3 Function Description

26.3.1 Interface

Translate the input video data(DVP/MIPI CSI/LVDS) into the requisite data format

- DVP BT656/BT1120 format

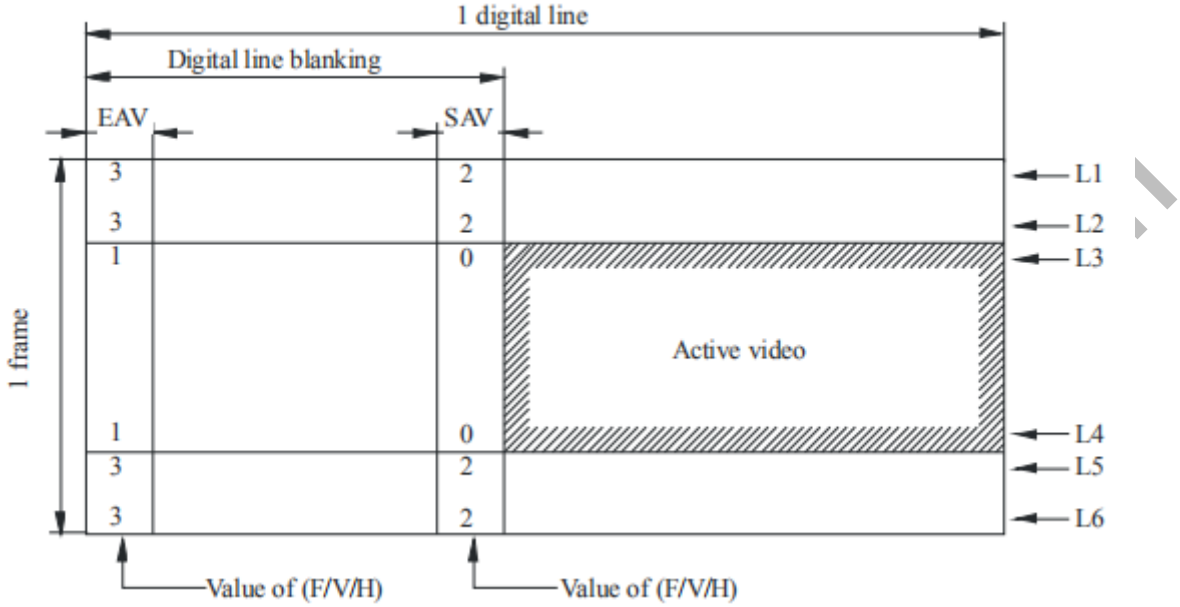


Fig. 26-2 BT656/BT1120 interlace timing relationship

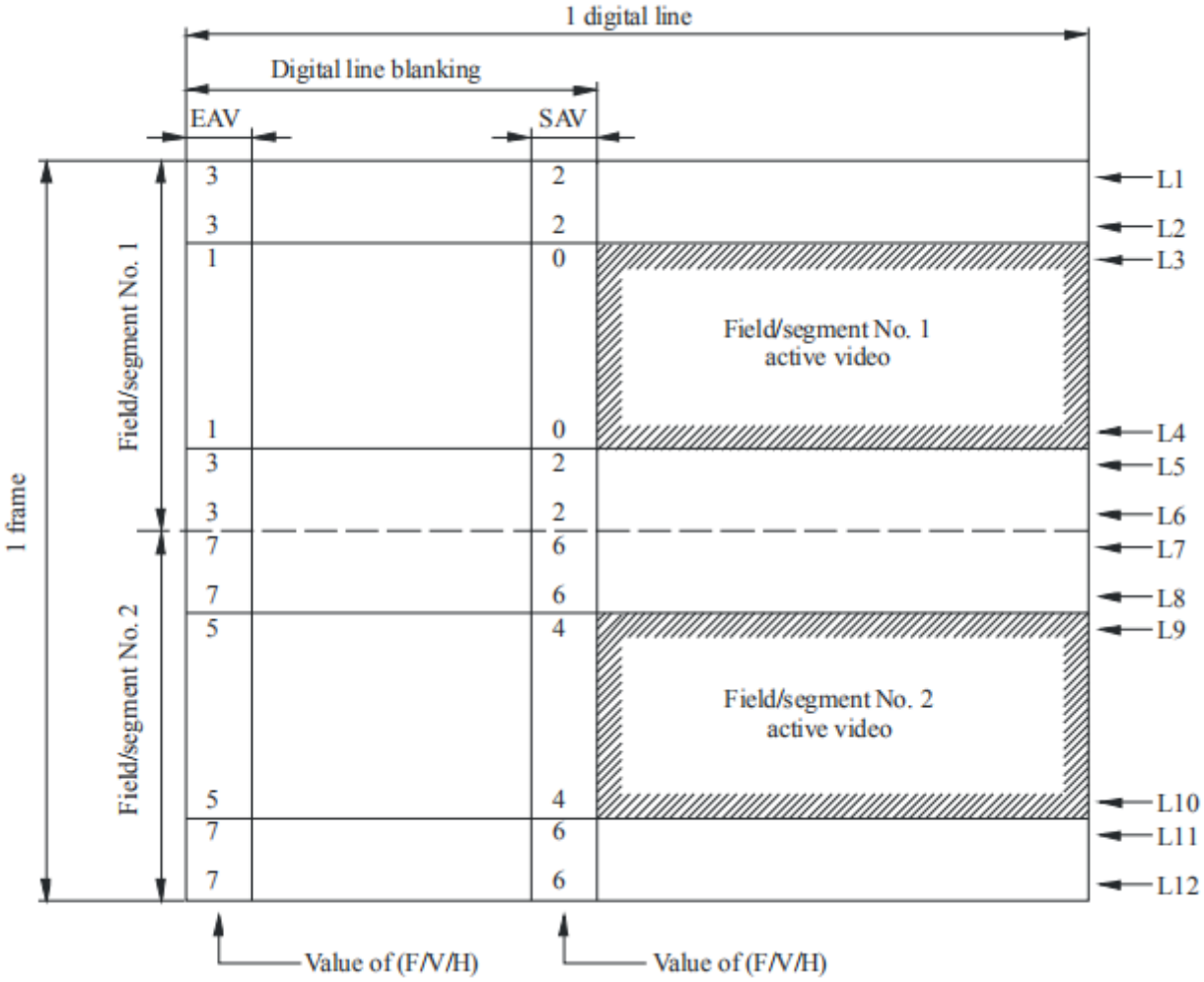


Fig. 26-3 BT656/BT1120 progressive timing relationship

- DVP BT601 format

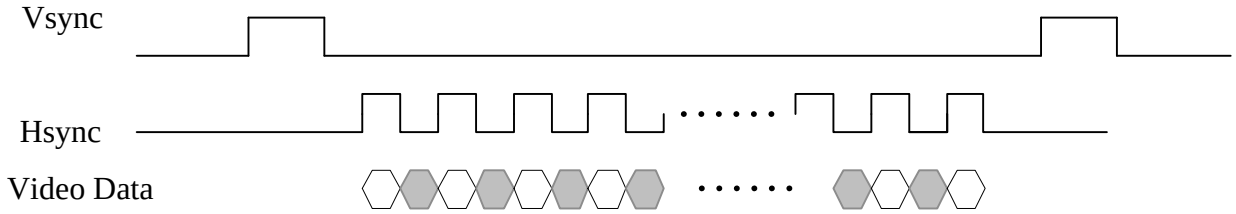


Fig. 26-4 BT656/BT1120 timing relationship

- MIPI CSI RAW8 format

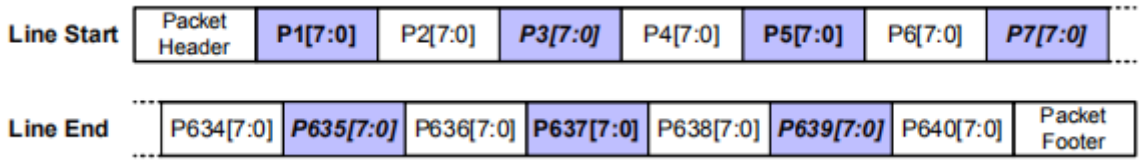


Fig. 26-5 MIPI CSI RAW8 format

- MIPI CSI RAW10 format

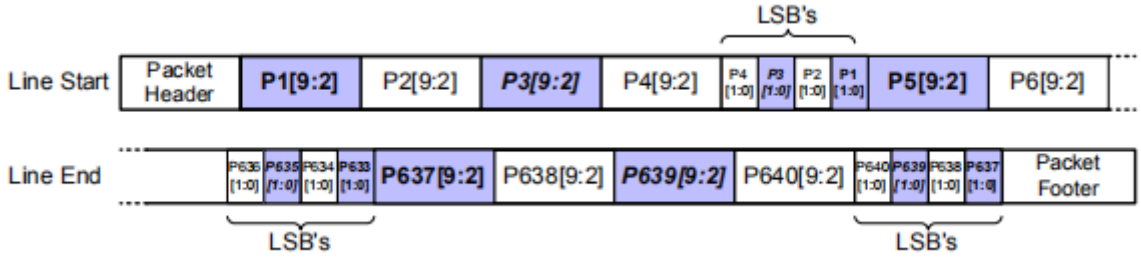


Fig. 26-6 MIPI CSI RAW10 format

- MIPI CSI RAW12 format

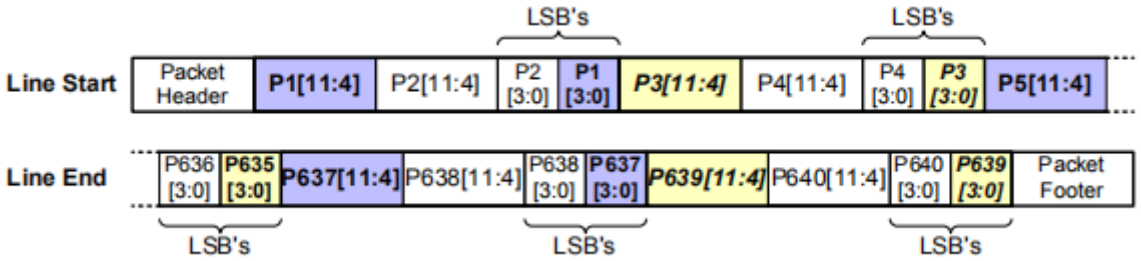


Fig. 26-7 MIPI CSI RAW12 format

- MIPI CSI RAW14 format

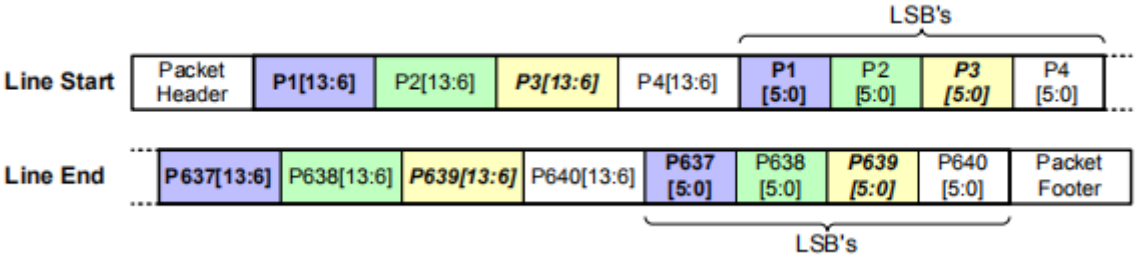


Fig. 26-8 MIPI CSI RAW14 format

- MIPI CSI YUV422 format

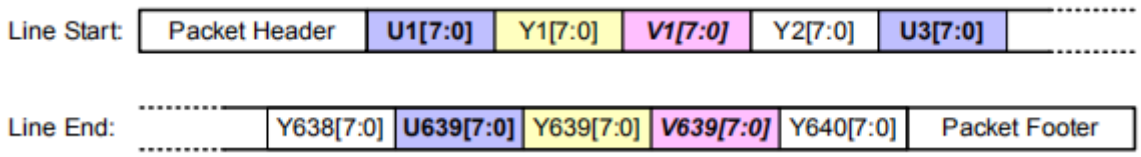


Fig. 26-9 MIPI CSI YUV422 format

- LVDS RAW8 format

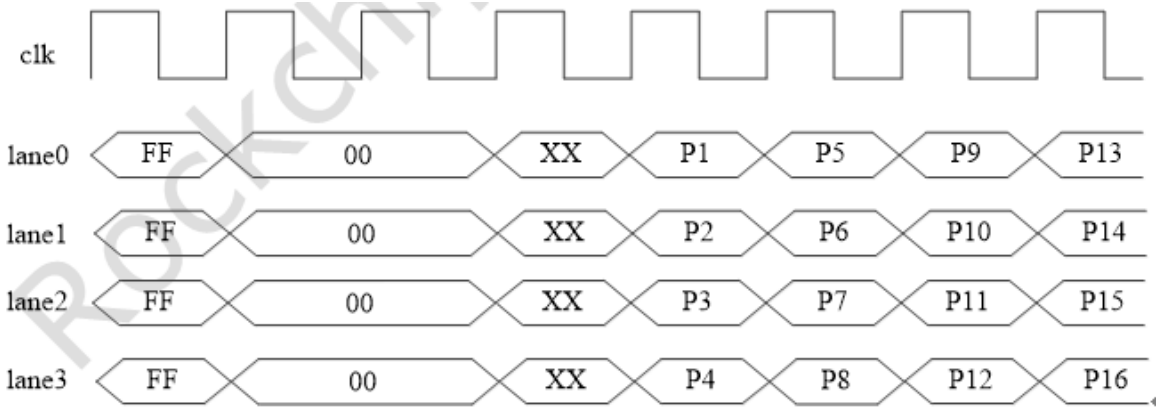


Fig. 26-10 LVDS RAW8 format

- LVDS RAW10 format

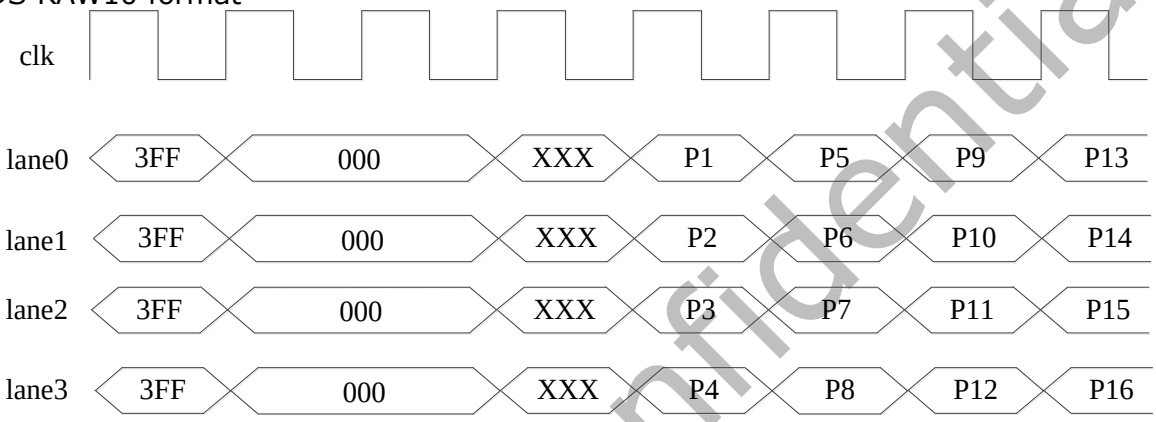


Fig. 26-11 LVDS RAW10 format

- LVDS RAW12 format

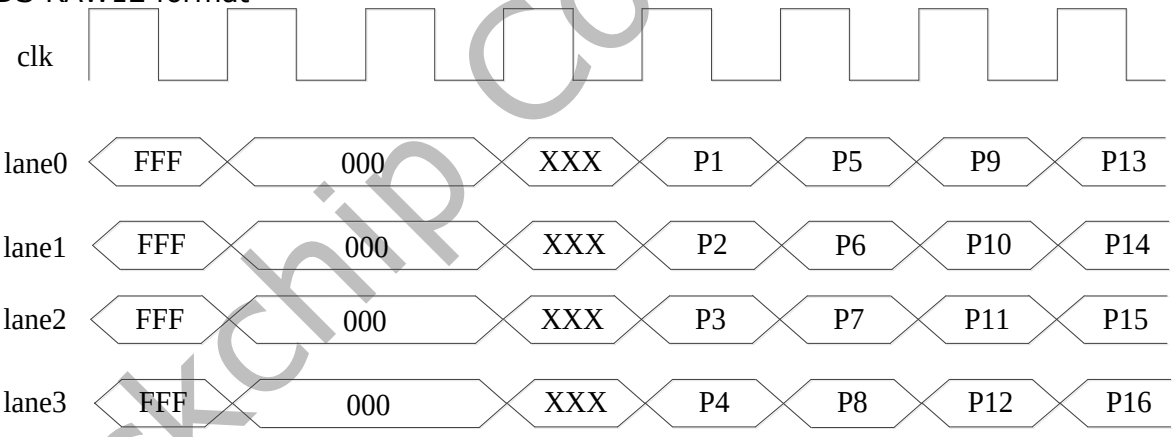


Fig. 26-12 LVDS RAW12 format

26.3.2 Crop

Bypass or crop the source video data to a smaller size destination.

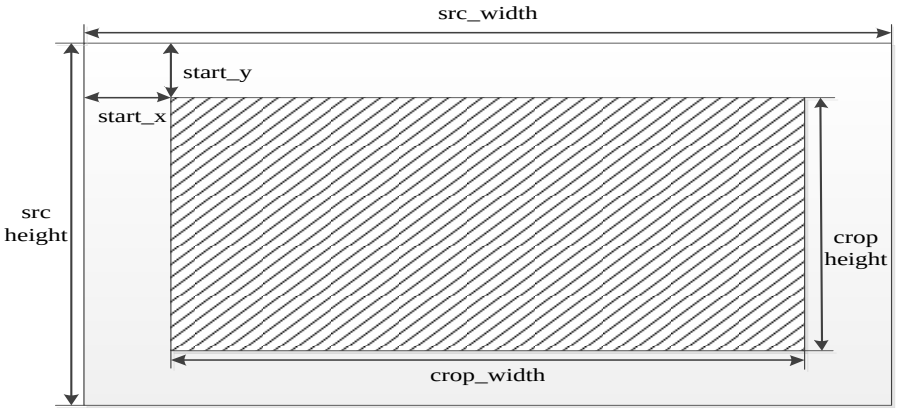


Fig. 26-13 Crop

26.3.3 Scale

Scale down for RAW data .

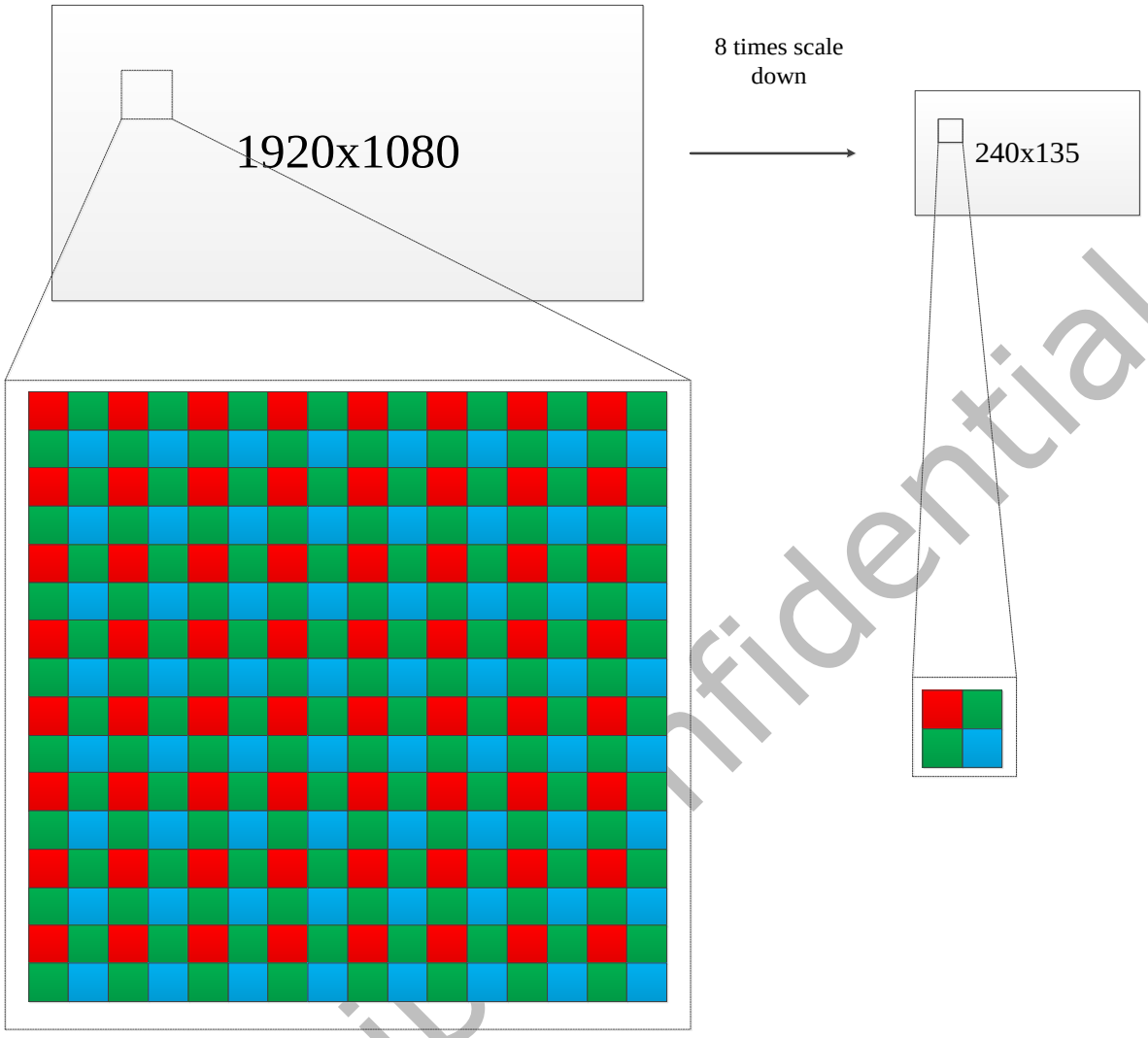


Fig. 26-14 8 times Scale down

26.3.4 Toisp

The parsed video data can go straight to ISP for real-time processing.

26.3.5 DMA

The DMA is used to transfer the data from crop module to the AXI master block which will send the data to the AXI bus.

- NV16

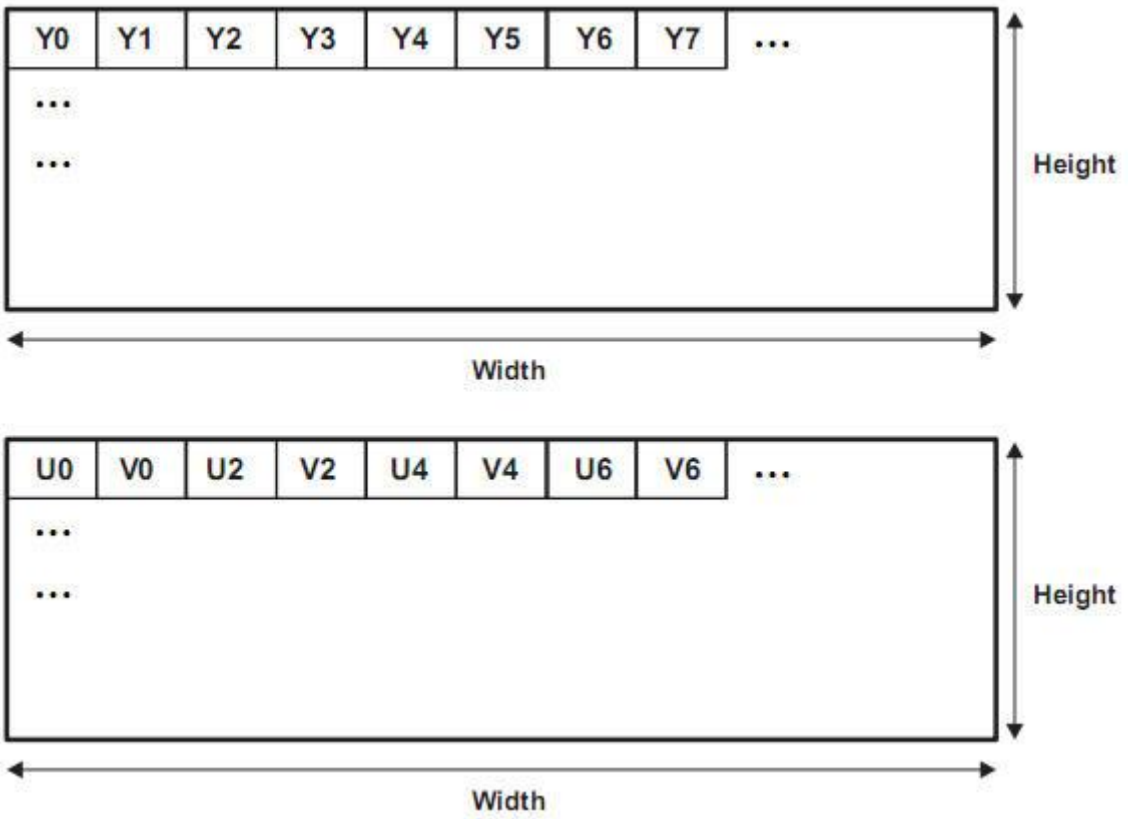


Fig. 26-15 NV16

- NV12

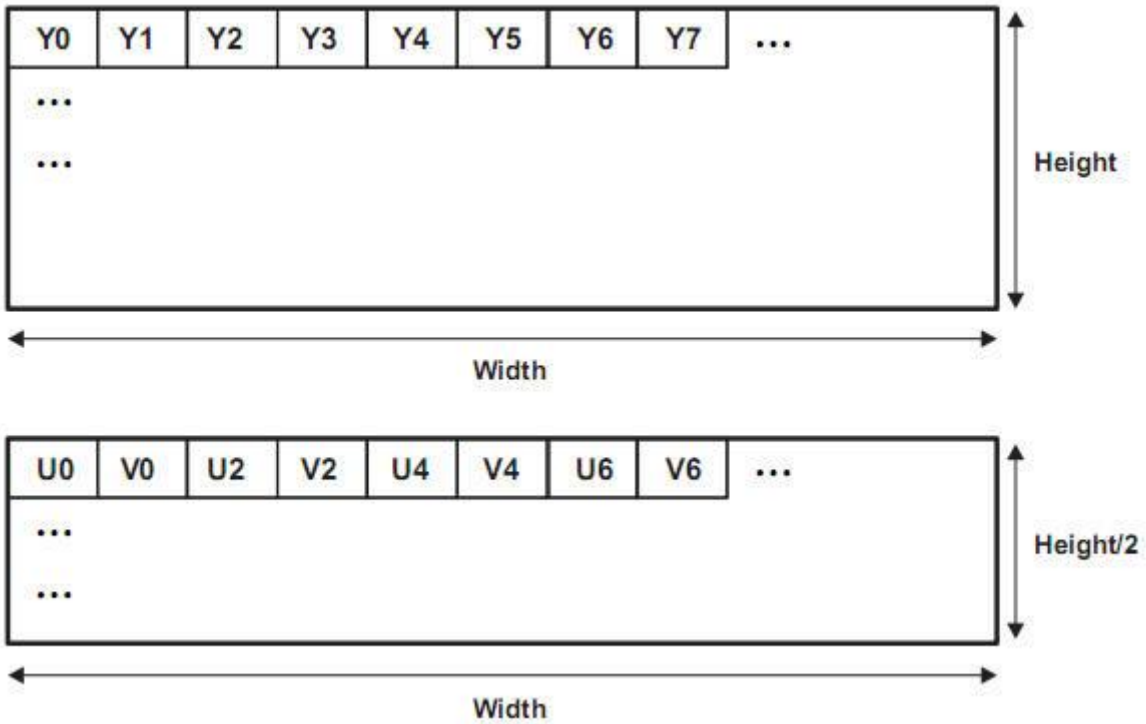


Fig. 26-16 NV12

- YUV400

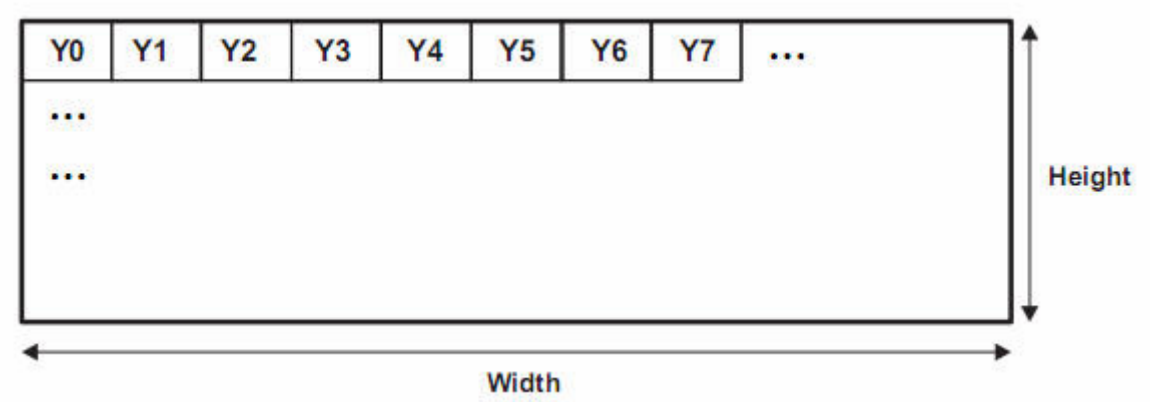


Fig. 26-17 YUV400

- YUYV

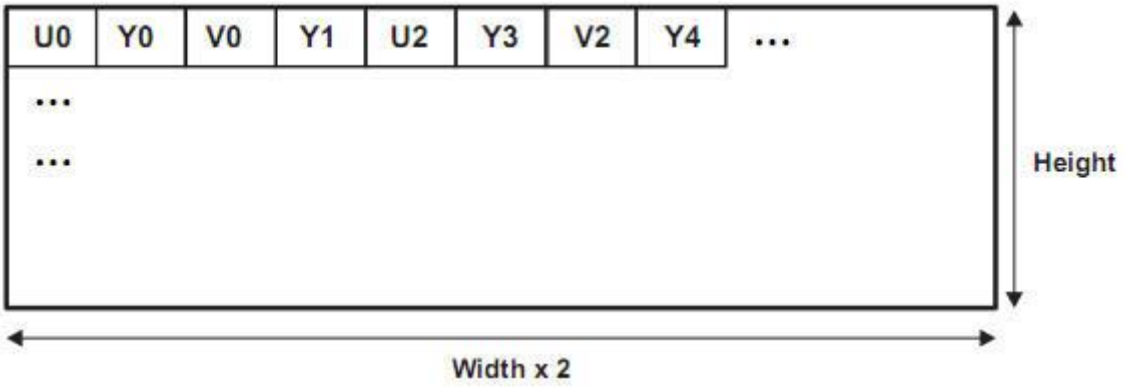


Fig. 26-18 YUYV

- Compact RAW

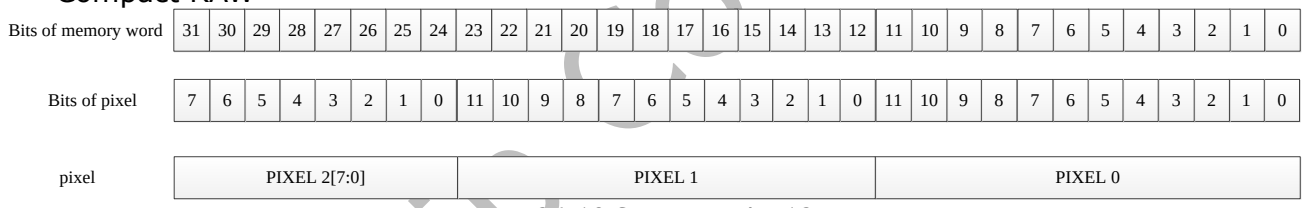


Fig. 26-19 Compact RAW12

- Noncompact RAW

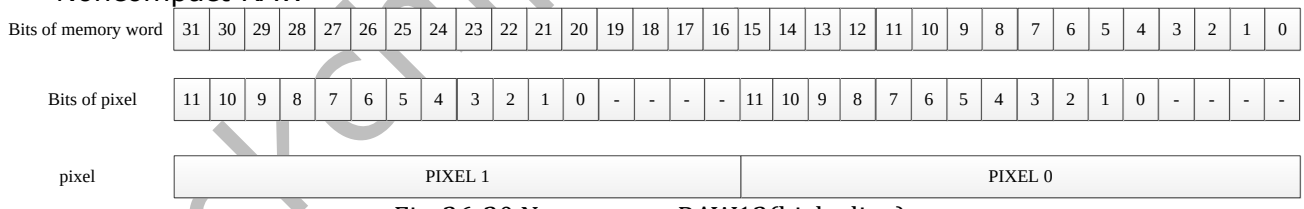


Fig. 26-20 Noncompact RAW12(high align)

26.3.6 AXI Master

Transmit the data to chip memory via the AXI Master.

26.3.7 AHB Slave

Host configure the registers via the AHB Slave.

26.4 Register Description

26.4.1 Internal Address Mapping

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

Table 26-1 VICAP Address Mapping

Base Address[11:8]	Device	Address Length	Offset Address Range
4'b0000	Global/DVP	256 BYTE	0x0000 ~ 0x00ff
4'b0001	MIPI/LVDS0	512 BYTE	0x0100 ~ 0x02ff

Base Address[11:8]	Device	Address Length	Offset Address Range
4'b0011	MIPI/LVDS1	512 BYTE	0x0300 ~ 0x04ff
4'b0111	Scale/TOISP	256 BYTE	0x0700 ~ 0x07ff

26.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
VICAP_GLB_CTRL	0x0000	W	0x00011C01	VICAP global control
VICAP_GLB_INTEN	0x0004	W	0x00000000	VICAP global interrupt enable.
VICAP_GLB_INTST	0x0008	W	0x00000000	VICAP global interrupt status.
VICAP_DVP_CTRL	0x0010	W	0x00000000	DVP path control
VICAP_DVP_INTEN	0x0014	W	0x00000000	DVP path interrupt status
VICAP_DVP_INTSTAT	0x0018	W	0x00000000	DVP path interrupt status
VICAP_DVP_FORMAT	0x001C	W	0x00000000	DVP path format
VICAP_DVP_SAV_EAV	0x0024	W	0xFEDCBA98	SAV/EAV for ACT/BLK
VICAP_DVP_CROP_SIZE	0x0028	W	0x01E002D0	The expected width and height of received image
VICAP_DVP_CROP_START	0x002C	W	0x00000000	The start point of DVP path cropping
VICAP_DVP_FRM0_ADDR_Y_ID0	0x0030	W	0x00000000	DVP path frame0 y address
VICAP_DVP_FRM0_ADDR_UV_ID0	0x0034	W	0x00000000	DVP path frame0 uv address
VICAP_DVP_FRM1_ADDR_Y_ID0	0x0038	W	0x00000000	DVP path frame1 y address
VICAP_DVP_FRM1_ADDR_UV_ID0	0x003C	W	0x00000000	DVP path frame1 uv address
VICAP_DVP_VIR_LINE_WIDTH	0x0070	W	0x00000000	DVP path virtual line width
VICAP_DVP_LINE_INT_NUM_ID0_1	0x0074	W	0x00000040	DVP path line interrupt number
VICAP_DVP_LINE_CNT_ID0_1	0x007C	W	0x00000000	DVP path line count
VICAP_DVP_PIX_NUM_ID0	0x0084	W	0x00000000	DVP path ID0 PIXEL NUMBER
VICAP_DVP_LINE_NUM_ID0	0x0088	W	0x00000000	DVP path ID0 LINE NUMBER
VICAP_DVP_SYNC_HEADER	0x00A4	W	0x00000000	DVP SYNC HEADER FOR BT656/BT1120
VICAP_MIPI0_ID0_CTRL0	0x0100	W	0x00000000	MIPI0 path id0 control0
VICAP_MIPI0_LVDS0_ID0_CTRL1	0x0104	W	0x00000000	MIPI0/LVDS0 path id0 control1
VICAP_MIPI0_ID1_CTRL0	0x0108	W	0x00000000	MIPI0 path id1 control0
VICAP_MIPI0_LVDS0_ID1_CTRL1	0x010C	W	0x00000000	MIPI0/LVDS0 path id1 control1
VICAP_MIPI0_ID2_CTRL0	0x0110	W	0x00000000	MIPI0 path id2 control0
VICAP_MIPI0_LVDS0_ID2_CTRL1	0x0114	W	0x00000000	MIPI0/LVDS0 path id2 control1
VICAP_MIPI0_ID3_CTRL0	0x0118	W	0x00000000	MIPI0 path id3 control0
VICAP_MIPI0_LVDS0_ID3_CTRL1	0x011C	W	0x00000000	MIPI0/LVDS0 path id3 control1
VICAP_MIPI0_LVDS0_CTRL	0x0120	W	0x00000000	MIPI0/LVDS0 path control

Name	Offset	Size	Reset Value	Description
VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID0	0x0124	W	0x00000000	First address of even frame for ID0 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID0	0x0128	W	0x00000000	First address of odd frame for ID0 Y path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID0	0x012C	W	0x00000000	First address of even frame for ID0 UV path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID0	0x0130	W	0x00000000	First address of odd frame for ID0 UV path
VICAP_MIPI0_LVDS0_VLW_ID0	0x0134	W	0x00000000	Virtual line width for ID0
VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID1	0x0138	W	0x00000000	First address of even frame for ID1 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID1	0x013C	W	0x00000000	First address of odd frame for ID1 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID1	0x0140	W	0x00000000	First address of even frame for ID1 UV path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID1	0x0144	W	0x00000000	First address of odd frame for ID1 UV path
VICAP_MIPI0_LVDS0_VLW_ID1	0x0148	W	0x00000000	Virtual line width of even frame for ID1 path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID2	0x014C	W	0x00000000	First address of even frame for ID2 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID2	0x0150	W	0x00000000	First address of odd frame for ID2 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID2	0x0154	W	0x00000000	First address of even frame for ID2 UV path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID2	0x0158	W	0x00000000	First address of odd frame for ID2 UV path
VICAP_MIPI0_LVDS0_VLW_ID2	0x015C	W	0x00000000	Virtual line width of even frame for ID2 path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID3	0x0160	W	0x00000000	First address of even frame for ID3 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID3	0x0164	W	0x00000000	First address of odd frame for ID3 Y/RAW/RGB path
VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID3	0x0168	W	0x00000000	First address of even frame for ID3 UV path
VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID3	0x016C	W	0x00000000	First address of odd frame for ID3 UV path
VICAP_MIPI0_LVDS0_VLW_ID3	0x0170	W	0x00000000	Virtual line width of even frame for ID3 path
VICAP_MIPI0_LVDS0_INTEN	0x0174	W	0x00000000	MIPI0/LVDS0 path interrupt enable
VICAP_MIPI0_LVDS0_INTSTAT	0x0178	W	0x00000000	MIPI0/LVDS0 path interrupt status
VICAP_MIPI0_LVDS0_LINE_INT_NUM_ID0_1	0x017C	W	0x00400040	Line number of the MIPI0/LVDS0 path ID0/1 line interrupt
VICAP_MIPI0_LVDS0_LINE_INT_NUM_ID2_3	0x0180	W	0x00400040	Line number of the MIPI0/LVDS0 path ID2/3 line interrupt
VICAP_MIPI0_LVDS0_LINE_CNT_ID0_1	0x0184	W	0x00000000	Line count of the MIPI0/LVDS0 path ID0/1
VICAP_MIPI0_LVDS0_LINE_CNT_ID2_3	0x0188	W	0x00000000	Line count of the MIPI0/LVDS0 path ID2/3
VICAP_MIPI0_LVDS0_ID0_CROP_START	0x018C	W	0x00000000	The start point of MIPI0/LVDS0 ID0 cropping

Name	Offset	Size	Reset Value	Description
VICAP_MIPI0_LVDS0_ID1_CROP_START	0x0190	W	0x00000000	The start point of MIPI0/LVDS0 ID1 cropping
VICAP_MIPI0_LVDS0_ID2_CROP_START	0x0194	W	0x00000000	The start point of MIPI0/LVDS0 ID2 cropping
VICAP_MIPI0_LVDS0_ID3_CROP_START	0x0198	W	0x00000000	The start point of MIPI0/LVDS0 ID3 cropping
VICAP_MIPI0_FRAME_NUM_VC0	0x019C	W	0x00000000	The frame number of virtual channel 0
VICAP_MIPI0_FRAME_NUM_VC1	0x01A0	W	0x00000000	The frame number of virtual channel 1
VICAP_MIPI0_FRAME_NUM_VC2	0x01A4	W	0x00000000	The frame number of virtual channel 2
VICAP_MIPI0_FRAME_NUM_VC3	0x01A8	W	0x00000000	The frame number of virtual channel 3
VICAP_MIPI0_ID0_EFFECT_CODE	0x01AC	W	0x00000000	The effect code of MIPI0 ID0
VICAP_MIPI0_ID1_EFFECT_CODE	0x01B0	W	0x00000000	The effect code of MIPI0 ID1
VICAP_MIPI0_ID2_EFFECT_CODE	0x01B4	W	0x00000000	The effect code of MIPI0 ID2
VICAP_MIPI0_ID3_EFFECT_CODE	0x01B8	W	0x00000000	The effect code of MIPI0 ID3
VICAP_MIPI0_ON_PAD_VALUE	0x01BC	W	0x00000000	The ON padding value of MIPI0
VICAP_MIPI0_LVDS0_SIZE_NUM_ID0	0x01C0	W	0x00000000	MIPI0/LVDS0 path ID0 SIZE NUMBER
VICAP_MIPI0_LVDS0_SIZE_NUM_ID1	0x01C4	W	0x00000000	MIPI0/LVDS0 path ID1 SIZE NUMBER
VICAP_MIPI0_LVDS0_SIZE_NUM_ID2	0x01C8	W	0x00000000	MIPI0/LVDS0 path ID2 SIZE NUMBER
VICAP_MIPI0_LVDS0_SIZE_NUM_ID3	0x01CC	W	0x00000000	MIPI0/LVDS0 path ID3 SIZE NUMBER
VICAP_LVDS0_ID0_CTRL0	0x01D0	W	0x00000000	LVDS0 path id0 control0
VICAP_LVDS0_ID1_CTRL0	0x01D4	W	0x00000000	LVDS0 path id1 control0
VICAP_LVDS0_ID2_CTRL0	0x01D8	W	0x00000000	LVDS0 path id2 control0
VICAP_LVDS0_ID3_CTRL0	0x01DC	W	0x00000000	LVDS0 path id3 control0
VICAP_LVDS0_SAV_EAV_ACT0_ID0	0x01E0	W	0x00000000	LVDS0 sync code of SAV_ACT0/EAV_ACT0 for id0
VICAP_LVDS0_SAV_EAV_BLK0_ID0	0x01E4	W	0x00000000	LVDS0 sync code of SAV_BLK0/EAV_BLK0 for id0
VICAP_LVDS0_SAV_EAV_ACT1_ID0	0x01E8	W	0x00000000	LVDS0 sync code of SAV_ACT1/EAV_ACT1 for id0
VICAP_LVDS0_SAV_EAV_BLK1_ID0	0x01EC	W	0x00000000	LVDS0 sync code of SAV_BLK1/EAV_BLK1 for id0
VICAP_LVDS0_SAV_EAV_ACT0_ID1	0x01F0	W	0x00000000	LVDS0 sync code of SAV_ACT0/EAV_ACT0 for id1
VICAP_LVDS0_SAV_EAV_BLK0_ID1	0x01F4	W	0x00000000	LVDS0 sync code of SAV_BLK0/EAV_BLK0 for id1
VICAP_LVDS0_SAV_EAV_ACT1_ID1	0x01F8	W	0x00000000	LVDS0 sync code of SAV_ACT1/EAV_ACT1 for id1
VICAP_LVDS0_SAV_EAV_BLK1_ID1	0x01FC	W	0x00000000	LVDS0 sync code of SAV_BLK1/EAV_BLK1 for id1
VICAP_LVDS0_SAV_EAV_ACT0_ID2	0x0200	W	0x00000000	LVDS0 sync code of SAV_ACT0/EAV_ACT0 for id2

Name	Offset	Size	Reset Value	Description
VICAP_LVDS0_SAV_EAV_BLK0_ID2	0x0204	W	0x00000000	LVDS0 sync code of SAV_BLK0/EAV_BLK0 for id2
VICAP_LVDS0_SAV_EAV_ACT1_ID2	0x0208	W	0x00000000	LVDS0 sync code of SAV_ACT1/EAV_ACT1 for id2
VICAP_LVDS0_SAV_EAV_BLK1_ID2	0x020C	W	0x00000000	LVDS0 sync code of SAV_BLK1/EAV_BLK1 for id2
VICAP_LVDS0_SAV_EAV_ACT0_ID3	0x0210	W	0x00000000	LVDS0 sync code of SAV_ACT0/EAV_ACT0 for id3
VICAP_LVDS0_SAV_EAV_BLK0_ID3	0x0214	W	0x00000000	LVDS0 sync code of SAV_BLK0/EAV_BLK0 for id3
VICAP_LVDS0_SAV_EAV_ACT1_ID3	0x0218	W	0x00000000	LVDS0 sync code of SAV_ACT1/EAV_ACT1 for id3
VICAP_LVDS0_SAV_EAV_BLK1_ID3	0x021C	W	0x00000000	LVDS0 sync code of SAV_BLK1/EAV_BLK1 for id3
VICAP_MIPI1_ID0_CTRL0	0x0300	W	0x00000000	MIPI1 path id0 control0
VICAP_MIPI1_LVDS1_ID0_CTRL1	0x0304	W	0x00000000	MIPI1/LVDS1 path id0 control1
VICAP_MIPI1_ID1_CTRL0	0x0308	W	0x00000000	MIPI1 path id1 control0
VICAP_MIPI1_LVDS1_ID1_CTRL1	0x030C	W	0x00000000	MIPI1/LVDS1 path id1 control1
VICAP_MIPI1_ID2_CTRL0	0x0310	W	0x00000000	MIPI1 path id2 control0
VICAP_MIPI1_LVDS1_ID2_CTRL1	0x0314	W	0x00000000	MIPI1/LVDS1 path id2 control1
VICAP_MIPI1_ID3_CTRL0	0x0318	W	0x00000000	MIPI1 path id3 control0
VICAP_MIPI1_LVDS1_ID3_CTRL1	0x031C	W	0x00000000	MIPI1/LVDS1 path id3 control1
VICAP_MIPI1_LVDS1_CTRL	0x0320	W	0x00000000	MIPI1/LVDS1 path control
VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID0	0x0324	W	0x00000000	First address of even frame for ID0 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID0	0x0328	W	0x00000000	First address of odd frame for ID0 Y path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID0	0x032C	W	0x00000000	First address of even frame for ID0 UV path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID0	0x0330	W	0x00000000	First address of odd frame for ID0 UV path
VICAP_MIPI1_LVDS1_VLW_ID0	0x0334	W	0x00000000	Virtual line width for ID0
VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID1	0x0338	W	0x00000000	First address of even frame for ID1 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID1	0x033C	W	0x00000000	First address of odd frame for ID1 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID1	0x0340	W	0x00000000	First address of even frame for ID1 UV path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID1	0x0344	W	0x00000000	First address of odd frame for ID1 UV path
VICAP_MIPI1_LVDS1_VLW_ID1	0x0348	W	0x00000000	Virtual line width of even frame for ID1 path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID2	0x034C	W	0x00000000	First address of even frame for ID2 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID2	0x0350	W	0x00000000	First address of odd frame for ID2 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID2	0x0354	W	0x00000000	First address of even frame for ID2 UV path

Name	Offset	Size	Reset Value	Description
VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID2	0x0358	W	0x00000000	First address of odd frame for ID2 UV path
VICAP_MIPI1_LVDS1_VLW_ID2	0x035C	W	0x00000000	Virtual line width of even frame for ID2 path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID3	0x0360	W	0x00000000	First address of even frame for ID3 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID3	0x0364	W	0x00000000	First address of odd frame for ID3 Y/RAW/RGB path
VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID3	0x0368	W	0x00000000	First address of even frame for ID3 UV path
VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID3	0x036C	W	0x00000000	First address of odd frame for ID3 UV path
VICAP_MIPI1_LVDS1_VLW_ID3	0x0370	W	0x00000000	Virtual line width of even frame for ID3 path
VICAP_MIPI1_LVDS1_INTEN	0x0374	W	0x00000000	MIPI1/LVDS1 path interrupt enable
VICAP_MIPI1_LVDS1_INTSTAT	0x0378	W	0x00000000	MIPI1/LVDS1 path interrupt status
VICAP_MIPI1_LVDS1_LINE_INT_NUM_ID0_1	0x037C	W	0x00400040	Line number of the MIPI1/LVDS1 path ID0/1 line interrupt
VICAP_MIPI1_LVDS1_LINE_INT_NUM_ID2_3	0x0380	W	0x00400040	Line number of the MIPI1/LVDS1 path ID2/3 line interrupt
VICAP_MIPI1_LVDS1_LINE_CNT_ID0_1	0x0384	W	0x00000000	Line count of the MIPI1/LVDS1 path ID0/1
VICAP_MIPI1_LVDS1_LINE_CNT_ID2_3	0x0388	W	0x00000000	Line count of the MIPI1/LVDS1 path ID2/3
VICAP_MIPI1_LVDS1_ID0_CROP_START	0x038C	W	0x00000000	The start point of MIPI1/LVDS1 ID0 cropping
VICAP_MIPI1_LVDS1_ID1_CROP_START	0x0390	W	0x00000000	The start point of MIPI1/LVDS1 ID1 cropping
VICAP_MIPI1_LVDS1_ID2_CROP_START	0x0394	W	0x00000000	The start point of MIPI1/LVDS1 ID2 cropping
VICAP_MIPI1_LVDS1_ID3_CROP_START	0x0398	W	0x00000000	The start point of MIPI1/LVDS1 ID3 cropping
VICAP_MIPI1_FRAME_NUM_VC0	0x039C	W	0x00000000	The frame number of virtual channel 0
VICAP_MIPI1_FRAME_NUM_VC1	0x03A0	W	0x00000000	The frame number of virtual channel 1
VICAP_MIPI1_FRAME_NUM_VC2	0x03A4	W	0x00000000	The frame number of virtual channel 2
VICAP_MIPI1_FRAME_NUM_VC3	0x03A8	W	0x00000000	The frame number of virtual channel 3
VICAP_MIPI1_ID0_EFFECT_CODE	0x03AC	W	0x00000000	The effect code of MIPI1 ID0
VICAP_MIPI1_ID1_EFFECT_CODE	0x03B0	W	0x00000000	The effect code of MIPI1 ID1
VICAP_MIPI1_ID2_EFFECT_CODE	0x03B4	W	0x00000000	The effect code of MIPI2 ID2
VICAP_MIPI1_ID3_EFFECT_CODE	0x03B8	W	0x00000000	The effect code of MIPI1 ID3
VICAP_MIPI1_ON_PAD_VALUE	0x03BC	W	0x00000000	The ON padding value of MIPI1
VICAP_MIPI1_LVDS1_SIZE_NUM_ID0	0x03C0	W	0x00000000	MIPI1/LVDS1 path ID0 SIZE NUMBER

Name	Offset	Size	Reset Value	Description
VICAP_MIPI1_LVDS1_SIZE_NUM_ID1	0x03C4	W	0x00000000	MIPI1/LVDS1 path ID1 SIZE NUMBER
VICAP_MIPI1_LVDS1_SIZE_NUM_ID2	0x03C8	W	0x00000000	MIPI1/LVDS1 path ID2 SIZE NUMBER
VICAP_MIPI1_LVDS1_SIZE_NUM_ID3	0x03CC	W	0x00000000	MIPI1/LVDS1 path ID3 SIZE NUMBER
VICAP_LVDS1_ID0_CTRL0	0x03D0	W	0x00000000	LVDS1 path id0 control0
VICAP_LVDS1_ID1_CTRL0	0x03D4	W	0x00000000	LVDS1 path id1 control0
VICAP_LVDS1_ID2_CTRL0	0x03D8	W	0x00000000	LVDS1 path id2 control0
VICAP_LVDS1_ID3_CTRL0	0x03DC	W	0x00000000	LVDS1 path id3 control0
VICAP_LVDS1_SAV_EAV_ACT0_ID0	0x03E0	W	0x00000000	LVDS1 sync code of SAV_ACT0/EAV_ACT0 for id0
VICAP_LVDS1_SAV_EAV_BLK0_ID0	0x03E4	W	0x00000000	LVDS1 sync code of SAV_BLK0/EAV_BLK0 for id0
VICAP_LVDS1_SAV_EAV_ACT1_ID0	0x03E8	W	0x00000000	LVDS1 sync code of SAV_ACT1/EAV_ACT1 for id0
VICAP_LVDS1_SAV_EAV_BLK1_ID0	0x03EC	W	0x00000000	LVDS1 sync code of SAV_BLK1/EAV_BLK1 for id0
VICAP_LVDS1_SAV_EAV_ACT0_ID1	0x03F0	W	0x00000000	LVDS1 sync code of SAV_ACT0/EAV_ACT0 for id1
VICAP_LVDS1_SAV_EAV_BLK0_ID1	0x03F4	W	0x00000000	LVDS1 sync code of SAV_BLK0/EAV_BLK0 for id1
VICAP_LVDS1_SAV_EAV_ACT1_ID1	0x03F8	W	0x00000000	LVDS1 sync code of SAV_ACT1/EAV_ACT1 for id1
VICAP_LVDS1_SAV_EAV_BLK1_ID1	0x03FC	W	0x00000000	LVDS1 sync code of SAV_BLK1/EAV_BLK1 for id1
VICAP_LVDS1_SAV_EAV_ACT0_ID2	0x0400	W	0x00000000	LVDS1 sync code of SAV_ACT0/EAV_ACT0 for id2
VICAP_LVDS1_SAV_EAV_BLK0_ID2	0x0404	W	0x00000000	LVDS1 sync code of SAV_BLK0/EAV_BLK0 for id2
VICAP_LVDS1_SAV_EAV_ACT1_ID2	0x0408	W	0x00000000	LVDS1 sync code of SAV_ACT1/EAV_ACT1 for id2
VICAP_LVDS1_SAV_EAV_BLK1_ID2	0x040C	W	0x00000000	LVDS1 sync code of SAV_BLK1/EAV_BLK1 for id2
VICAP_LVDS1_SAV_EAV_ACT0_ID3	0x0410	W	0x00000000	LVDS1 sync code of SAV_ACT0/EAV_ACT0 for id3
VICAP_LVDS1_SAV_EAV_BLK0_ID3	0x0414	W	0x00000000	LVDS1 sync code of SAV_BLK0/EAV_BLK0 for id3
VICAP_LVDS1_SAV_EAV_ACT1_ID3	0x0418	W	0x00000000	LVDS1 sync code of SAV_ACT1/EAV_ACT1 for id3
VICAP_LVDS1_SAV_EAV_BLK1_ID3	0x041C	W	0x00000000	LVDS1 sync code of SAV_BLK1/EAV_BLK1 for id3
VICAP_SCL_CH_CTRL	0x0700	W	0x00000000	Scale channel0 path control
VICAP_SCL_CTRL	0x0704	W	0x00000000	MIPI3 path control
VICAP_SCL_FRAME0_ADDR_CH0	0x0708	W	0x00000000	First address of even frame for CH0 path
VICAP_SCL_FRAME1_ADDR_CH0	0x070C	W	0x00000000	First address of odd frame for CH0 path
VICAP_SCL_VLW_CH0	0x0710	W	0x00000000	Virtual line width for CH0
VICAP_SCL_CH0_BLACK_LEVEL	0x0738	W	0x00000000	Scale channel0 black level
VICAP_TOISP0_CH_CTRL	0x0780	W	0x00000000	TOISP0 path control
VICAP_TOISP0_CROP_SIZE	0x0784	W	0x01E002D0	The expected width and height of received image

Name	Offset	Size	Reset Value	Description
VICAP_TOISP0_CROP_START	0x0788	W	0x00000000	The start point of toisp0 path cropping

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

26.4.3 Detail Registers Description

VICAP_GLB_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x1	ro_axi_idle 1'b0: AXI is working 1'b1: AXI is idle.
15:13	RO	0x0	reserved
12	RW	0x1	sw_dma_group_mode 1'b0: Group 2(2 bursts gather) 1'b1: Group 4(4 bursts gather). Thi bit is valid when sw_dma_group_en=1.
11	RW	0x1	sw_dma_adapt_en 1'b0: Disable dma self-adapt(dma will transport data according the configured height) 1'b1: Enable dma self-adapt(dma will transport data according the actual height).
10	RW	0x1	sw_dma_group_en 1'b0: Disable dma burst group 1'b1: Enable dma burst group.
9	RW	0x0	sw_infrst_en 1'b0: Disable pre-interface reset 1'b1: Enable pre-interface reset.
8	RW	0x0	sw_clk_gating_dis 1'b0: Enable the auto clk gating 1'b1: Disable the auto clk gating
7	RO	0x0	reserved
6:5	RW	0x0	sw_soft_rst_mode 2'b00: Soft reset and protect toisp path frame integrity; 2'b01: Soft reset and protect toisp path frame integrity and pull down sw_cap_en automatically; 2'b10: Soft reset directly; 2'b11: Reserved.
4	RW	0x0	sw_soft_rst 1'b0: Not reset 1'b1: Reset the VICAP(except MMU/AXI MASTER/REG FILE)
3:1	RO	0x0	reserved
0	RW	0x1	sw_cap_en 1'b0: Disable all interface capture 1'b1: Enable capture(still depends on sw_xxx_cap_en).

VICAP_GLB_INTEN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	RW	0x0	toisp0_fifo_overflow_inten Enable the interrupt of fifo overflow of TOISP0 path. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
25:23	RO	0x0	reserved
22	RW	0x0	frame_end_toisp0_ch2_inten Enable the interrupt of frame end for TOISP0 channel2. 1'b0: Disable 1'b1: Enable
21	RW	0x0	frame_end_toisp0_ch1_inten Enable the interrupt of frame end for TOISP0 channel1. 1'b0: Disable 1'b1: Enable
20	RW	0x0	frame_end_toisp0_ch0_inten Enable the interrupt of frame end for TOISP0 channel0. 1'b0: Disable 1'b1: Enable
19:17	RO	0x0	reserved
16	RW	0x0	frame_start_toisp0_ch2_inten Enable the interrupt of frame start for TOISP0 channel2. 1'b0: Disable 1'b1: Enable
15	RW	0x0	frame_start_toisp0_ch1_inten Enable the interrupt of frame start for TOISP0 channel1. 1'b0: Disable 1'b1: Enable
14	RW	0x0	frame_start_toisp0_ch0_inten Enable the interrupt of frame start for TOISP0 channel0. 1'b0: Disable 1'b1: Enable
13:11	RO	0x0	reserved
10	RW	0x0	dma_fifo_overflow_scl_ch0_inten Enable the interrupt of dma fifo overflow of scale path CH0. 1'b0: Disable 1'b1: Enable
9:4	RO	0x00	reserved
3	RW	0x0	frame1_dma_end_scl_ch0_inten Enable the interrupt of end of odd frame for scale CH0. 1'b0: Disable 1'b1: Enable
2	RW	0x0	frame0_dma_end_scl_ch0_inten Enable the interrupt of end of even frame for scale CH0. 1'b0: Disable 1'b1: Enable
1	RO	0x0	reserved
0	RW	0x0	axi0_bus_err_inten Enable the interrupt of axi0 bus error. 1'b0: Disable 1'b1: Enable

VICAP_GLB_INTST

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26	W1C	0x0	toisp0_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
25:23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22	W1C	0x0	frame_end_toisp0_ch2_intst 1'b0: No interrupt 1'b1: Interrupt
21	W1C	0x0	frame_end_toisp0_ch1_intst 1'b0: No interrupt 1'b1: Interrupt
20	W1C	0x0	frame_end_toisp0_ch0_intst 1'b0: No interrupt 1'b1: Interrupt
19:17	RO	0x0	reserved
16	W1C	0x0	frame_start_toisp0_ch2_intst 1'b0: No interrupt 1'b1: Interrupt
15	W1C	0x0	frame_start_toisp0_ch1_intst 1'b0: No interrupt 1'b1: Interrupt
14	W1C	0x0	frame_start_toisp0_ch0_intst 1'b0: No interrupt 1'b1: Interrupt
13:11	RO	0x0	reserved
10	W1C	0x0	dma_fifo_overflow_scl_ch0_intst 1'b0: No interrupt 1'b1: Interrupt
9:4	RO	0x00	reserved
3	W1C	0x0	frame1_dma_end_scl_ch0_intst 1'b0: No interrupt 1'b1: Interrupt
2	W1C	0x0	frame0_dma_end_scl_ch0_intst 1'b0: No interrupt 1'b1: Interrupt
1	RO	0x0	reserved
0	W1C	0x0	axi0_bus_err_intst 1'b0: No interrupt 1'b1: Interrupt

VICAP_DVP_CTRL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:24	RO	0x00	reserved
23	RW	0x0	sw_dma_en_id3 1'b0: Disable 1'b1: Enable
22	RW	0x0	sw_dma_en_id2 1'b0: Disable 1'b1: Enable
21	RW	0x0	sw_dma_en_id1 1'b0: Disable 1'b1: Enable
20	RW	0x0	sw_dma_en_id0 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
19:18	RW	0x0	sw_soft_rst_mode 2'b00: Soft reset and protect toisp path frame integrity; 2'b01: Soft reset and protect toisp path frame integrity and pull down sw_cap_en automatically; 2'b10: Soft reset directly; 2'b11: Reserved.
17	RW	0x0	sw_soft_rst 1'b0: Not reset 1'b1: Reset the VICAP DVP path(except MMU/AXI MASTER/REG FILE)
16	RW	0x0	sw_dma_rst Write 1 will reset VICAP DVP path dma. When this bit change to 0,dma is reseted completely.
15:13	RW	0x0	sw_press_value Press value.
12	RW	0x0	sw_press_en 1'b0: Disable 1'b1: Enable
11:9	RW	0x0	sw_hurry_value Hurry value.
8	RW	0x0	sw_hurry_en 1'b0: Disable 1'b1: Enable
7	RO	0x0	reserved
6:5	RW	0x0	sw_water_line 2'b00: 75% 2'b01: 50% 2'b10: 25% 2'b11: 0%
4	RW	0x0	sw_cap_en_id3 1'b0: Disable 1'b1: Enable
3	RW	0x0	sw_cap_en_id2 1'b0: Disable 1'b1: Enable
2	RW	0x0	sw_cap_en_id1 1'b0: Disable 1'b1: Enable
1	RW	0x0	sw_cap_en_id0 1'b0: Disable 1'b1: Enable
0	RW	0x0	sw_cap_en 1'b0: Disable 1'b1: Enable

VICAP_DVP_INTEN

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22	RW	0x0	size_err_id0_inten 1'b0: Disable 1'b1: Enable
21:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18	RW	0x0	line_id0_inten 1'b0: Disable 1'b1: Enable
17	RW	0x0	bandwidth_lack_inten Enable the interrupt of bandwidth lack. 1'b0: Disable 1'b1: Enable
16	RW	0x0	dma_fifo_overflow_inten Enable the interrupt of dma fifo overflow . 1'b0: Disable 1'b1: Enable
15:10	RO	0x00	reserved
9	RW	0x0	frame1_dma_end_id0_inten Enable the interrupt of end of odd frame for ID0. 1'b0: Disable 1'b1: Enable
8	RW	0x0	frame0_dma_end_id0_inten Enable the interrupt of end of even frame for ID0. 1'b0: Disable 1'b1: Enable
7:2	RO	0x00	reserved
1	RW	0x0	frame1_start_id0_inten Enable the interrupt of start of odd frame for ID0. 1'b0: Disable 1'b1: Enable
0	RW	0x0	frame0_start_id0_inten Enable the interrupt of start of even frame for ID0. 1'b0: Disable 1'b1: Enable

VICAP_DVP_INTSTAT

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:23	RO	0x000	reserved
22	W1C	0x0	size_err_id0_intst 1'b0: No interrupt 1'b1: Interrupt
21:19	RO	0x0	reserved
18	W1C	0x0	line_id0_intst 1'b0: No interrupt 1'b1: Interrupt
17	W1C	0x0	bandwidth_lack_intst 1'b0: No interrupt 1'b1: Interrupt
16	W1C	0x0	dma_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
15:10	RO	0x00	reserved
9	W1C	0x0	frame1_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
8	W1C	0x0	frame0_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
7:2	RO	0x00	reserved

Bit	Attr	Reset Value	Description
1	W1C	0x0	frame1_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt
0	W1C	0x0	frame0_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt

VICAP_DVP_FORMAT

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:22	RO	0x000	reserved
21	RW	0x0	sw_align 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
20:19	RW	0x0	sw_yuv_out_order 2'b00: UYVY 2'b01: VYUY 2'b10: YUYV 2'b11: YVYU
18:16	RW	0x0	sw_wrddr_type 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
15:14	RO	0x0	reserved
13	RW	0x0	sw_only_sav_mode 1'b0: Detect SAV and EAV 1'b1: Only detect SAV
12	RW	0x0	sw_yc_swap 1'b0: No swap for y and c 1'b1: Swap for y and c Only for BT1120 mode.
11	RW	0x0	sw_dualedge_en 1'b0: Only use single edge of clock 1'b1: Use double edges of clock Only for BT1120 mode.
10	RW	0x0	sw_field_order 1'b0: Odd field first 1'b1: Even field first.
9	RW	0x0	sw_interlace_en 1'b0: Progress 1'b1: Interlace
8:7	RW	0x0	sw_raw_width 2'b00: 8bit raw data 2'b01: 10bit raw data 2'b10: 12bit raw data 2'b11: Reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	sw_yuv_in_order 2'b00: UYVY 2'b01: VYUY 2'b10: YUYV 2'b11: YVYU
4:2	RW	0x0	sw_input_mode 3'b000: BT601 YUV422 3'b001: BT601 RAW 3'b010: BT656 YUV422 3'b011: BT1120 YUV422 3'b100: SONY RAW Others: Reserved
1	RW	0x0	sw_href_pol 1'b0: High active 1'b1: Low active
0	RW	0x0	sw_vsync_pol 1'b0: Low active 1'b1: High active

VICAP_DVP_SAV_EAV

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:28	RW	0xf	sw_eav_blk1 EAV for even field blank.
27:24	RW	0xe	sw_sav_blk1 SAV for even field blank.
23:20	RW	0xd	sw_eav_act1 EAV for even field active.
19:16	RW	0xc	sw_sav_act1 SAV for even field active.
15:12	RW	0xb	sw_eav_blk0 EAV for odd field blank.
11:8	RW	0xa	sw_sav_blk0 SAV for odd field blank.
7:4	RW	0x9	sw_eav_act0 EAV for odd field active.
3:0	RW	0x8	sw_sav_act0 SAV for odd field active.

VICAP_DVP_CROP_SIZE

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x01e0	sw_height The expected height of received image.
15:13	RO	0x0	reserved
12:0	RW	0x02d0	sw_width The expected width of received image.

VICAP_DVP_CROP_START

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0000	sw_start_y The vertical ordinate of the start point.

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12:0	RW	0x0000	sw_start_x The horizontal ordinate of the start point.

VICAP_DVP_FRM0_ADDR_Y_ID0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id0 DVP path frame0 y address.

VICAP_DVP_FRM0_ADDR_UV_ID0

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id0 DVP path frame0 uv address.

VICAP_DVP_FRM1_ADDR_Y_ID0

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id1 DVP path frame1 y address.

VICAP_DVP_FRM1_ADDR_UV_ID0

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id0 DVP path frame1 uv address.

VICAP_DVP_VIR_LINE_WIDTH

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	vir_line_width DVP path virtual line width.

VICAP_DVP_LINE_INT_NUM_ID0_1

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	reserved
15	RW	0x0	line_int_mode_id0 1'b0: One-time mode 1'b1: Circular mode
14:13	RO	0x0	reserved
12:0	RW	0x0040	sw_line_int_num_id0 For one-time mode,if line_int_num_id0=100,then channel 0 receive 100th line,the line_id0_intst will be 1. For circular mode ,if line_int_num_id0=100,then channel 0 receive 100th line200th line300th line.....the line_id0_intst will be 1.

VICAP_DVP_LINE_CNT_ID0_1

Address: Operational Base + offset (0x007C)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved

Bit	Attr	Reset Value	Description
13:0	RO	0x0000	ro_line_cnt_id0 Current line count.

VICAP_DVP_PIX_NUM_ID0

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_uv_pix_num_id0 The id0 UV pixel number in one line.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_y_pix_num_id0 The id0 Y pixel number in one line.

VICAP_DVP_LINE_NUM_ID0

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_uv_line_num_id0 The id0 UV line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_y_line_num_id0 The id0 Y line number in one frame.

VICAP_DVP_SYNC_HEADER

Address: Operational Base + offset (0x00A4)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	ro_sync_header_7 Sync header 7.
27:24	RO	0x0	ro_sync_header_6 Sync header 6.
23:20	RO	0x0	ro_sync_header_5 Sync header 5.
19:16	RO	0x0	ro_sync_header_4 Sync header 4.
15:12	RO	0x0	ro_sync_header_3 Sync header 3.
11:8	RO	0x0	ro_sync_header_2 Sync header 2.
7:4	RO	0x0	ro_sync_header_1 Sync header 1.
3:0	RO	0x0	ro_sync_header_0 Sync header 0. Header 0 is the latest,and header 7 is the earliest.

VICAP_MIPI0_ID0_CTRL0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id0 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.

Bit	Attr	Reset Value	Description
29	RW	0x0	sw_vc_hdr_protect_id0 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id0 Enable dma transport id0. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id0 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id0 Select command mode for id0. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id0 If sw_mipi0_on_hdr_mode_id0=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id0=1, id0 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id0 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id0 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id0 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id0 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id0 Data type for id0.
9:8	RW	0x0	sw_vc_id0 Virtual channel for id0.
7:5	RW	0x0	sw_wrddr_type_id0 The write ddr type of id0. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	sw_crop_en_id0 Enable to crop for id0. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id0 The parse type of id0. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id0 Enable to capture id0. 1'b0: Disable 1'b1: Enable

VICAP_MIPI0_LVDS0_ID0_CTRL1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id0 Height for id0.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id0 Width for id0.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id0 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_ID1_CTRL0

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id1 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id1 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id1 Enable dma transport id1. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id1 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.

Bit	Attr	Reset Value	Description
26	RW	0x0	sw_command_mode_en_id1 Select command mode for id1. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id1 If sw_mipi0_on_hdr_mode_id1=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id1=1, id1 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id1 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id1 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id1 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id1 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id1 Data type for id1.
9:8	RW	0x0	sw_vc_id1 Virtual channel for id1.
7:5	RW	0x0	sw_wrddr_type_id1 The write ddr type of id1. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id1 Enable to crop for id1. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id1 The parse type of id1. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_cap_en_id1 Enable to capture id1. 1'b0: Disable 1'b1: Enable

VICAP_MIPI0_LVDS0_ID1_CTRL1

Address: Operational Base + offset (0x010C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id1 Height for id1.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id1 Width for id1.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id1 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_ID2_CTRL0

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id2 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id2 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id2 Enable dma transport id2. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id2 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id2 Select command mode for id2. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id2 If sw_mipi0_on_hdr_mode_id2=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id2=1, id2 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id2 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved

Bit	Attr	Reset Value	Description
21:20	RW	0x0	sw_hdr_mode_id2 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id2 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id2 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id2 Data type for id2.
9:8	RW	0x0	sw_vc_id2 Virtual channel for id2.
7:5	RW	0x0	sw_wrddr_type_id2 The write ddr type of id2. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id2 Enable to crop for id2. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id2 The parse type of id2. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id2 Enable to capture id2. 1'b0: Disable 1'b1: Enable

VICAP_MIPI0_LVDS0_ID2_CTRL1

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id2 Height for id2.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	sw_width_id2 Width for id2.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id2 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_ID3_CTRL0

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id3 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id3 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id3 Enable dma transport id3. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id3 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id3 Select command mode for id3. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id3 If sw_mipi0_on_hdr_mode_id3=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id3=1, id3 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id3 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id3 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id3 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU

Bit	Attr	Reset Value	Description
17:16	RW	0x0	sw_yuyv_in_order_id3 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id3 Data type for id3.
9:8	RW	0x0	sw_vc_id3 Virtual channel for id3.
7:5	RW	0x0	sw_wrddr_type_id3 The write ddr type of id3. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id3 Enable to crop for id3. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id3 The parse type of id3. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id3 Enable to capture id3. 1'b0: Disable 1'b1: Enable

VICAP_MIPI0_LVDS0_ID3_CTRL1

Address: Operational Base + offset (0x011C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id3 Height for id3.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id3 Width for id3.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id3 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_LVDS0_CTRL

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_drop_frame_main_id The id that is the last coming in HDR mode.

Bit	Attr	Reset Value	Description
29:24	RW	0x00	sw_drop_frame_m Receive m frames,and drop n frames.(m>0,n>0)
23:21	RW	0x0	sw_drop_frame_n Receive m frames,and drop n frames.(m>0,n>0)
20	RW	0x0	sw_drop_frame_en_all 1'b0: Disable drop all id frame 1'b1: Enable drop all id frame.
19:18	RW	0x0	sw_soft_rst_mode 2'b00: Soft reset and protect toisp path frame integrity; 2'b01: Soft reset and protect toisp path frame integrity and pull down sw_cap_en automatically; 2'b10: Soft reset directly; 2'b11: Reserved.
17	RW	0x0	sw_soft_rst 1'b0: Not reset 1'b1: Reset the VICAP MIPI0 path(except MMU/AXI MASTER/REG FILE)
16	RW	0x0	sw_dma_rst Write 1 will reset VICAP MIPI0 path dma. When this bit change to 0,dma is reseted completely.
15:13	RW	0x0	sw_press_value Press value.
12	RW	0x0	sw_press_en 1'b0: Disable press 1'b1: Enable press
11	RW	0x0	sw_drop_frame_en_id3 1'b0: Disable drop id3 frame 1'b1: Enable drop id3 frame.
10	RW	0x0	sw_drop_frame_en_id2 1'b0: Disable drop id2 frame 1'b1: Enable drop id2 frame.
9	RW	0x0	sw_drop_frame_en_id1 1'b0: Disable drop id1 frame 1'b1: Enable drop id1 frame.
8	RW	0x0	sw_drop_frame_en_id0 1'b0: Disable drop id0 frame 1'b1: Enable drop id0 frame.
7:5	RW	0x0	sw_hurry_value Hurry value.
4	RW	0x0	sw_hurry_en 1'b0: Disable hurry 1'b1: Enable hurry
3	RW	0x0	sw_mipi_lvds_sel 1'b0: MIPI path 1'b1: LVDS path
2:1	RW	0x0	sw_water_line 2'b00: 75% 2'b01: 50% 2'b10: 25% 2'b11: 0%
0	RW	0x0	sw_cap_en 1'b0: Disable capture all id 1'b1: Enable capture all id.

VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID0

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id0 First address of even frame for ID0 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID0

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id0 First address of odd frame for ID0 Y path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID0

Address: Operational Base + offset (0x012C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id0 First address of even frame for ID0 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID0

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id0 First address of odd frame for ID0 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_VLW_ID0

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id0 Virtual line width of even frame for ID0 path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID1

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id1 First address of even frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID1

Address: Operational Base + offset (0x013C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id1 First address of odd frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID1

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id1 First address of even frame for ID1 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID1

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id1 First address of odd frame for ID1 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_VLW_ID1

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id1 Virtual line width of even frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID2

Address: Operational Base + offset (0x014C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id2 First address of even frame for ID2 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID2

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id2 First address of odd frame for ID2 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID2

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id2 First address of even frame for ID2 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID2

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id0 First address of odd frame for ID2 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_VLW_ID2

Address: Operational Base + offset (0x015C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id2 Virtual line width of even frame for ID2 path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_Y_ID3

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id3 First address of even frame for ID3 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_Y_ID3

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id3 First address of odd frame for ID3 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME0_ADDR_UV_ID3

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id3 First address of even frame for ID3 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_FRAME1_ADDR_UV_ID3

Address: Operational Base + offset (0x016C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id3 First address of odd frame for ID3 UV path(must be aligned to double word).

VICAP_MIPI0_LVDS0_VLW_ID3

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id3 Virtual line width of even frame for ID3 path(must be aligned to double word).

VICAP_MIPI0_LVDS0_INTEN

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	size_err_id3_inten 1'b0: Disable 1'b1: Enable
26	RW	0x0	size_err_id2_inten 1'b0: Disable 1'b1: Enable
25	RW	0x0	size_err_id1_inten 1'b0: Disable 1'b1: Enable
24	RW	0x0	size_err_id0_inten 1'b0: Disable 1'b1: Enable
23	RW	0x0	line_id3_inten 1'b0: Disable 1'b1: Enable
22	RW	0x0	line_id2_inten 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
21	RW	0x0	line_id1_inten 1'b0: Disable 1'b1: Enable
20	RW	0x0	line_id0_inten 1'b0: Disable 1'b1: Enable
19	RW	0x0	csi2rx_fifo_overflow_inten 1'b0: Disable 1'b1: Enable
18	RW	0x0	bandwidth_lack_inten 1'b0: Disable 1'b1: Enable
17	RW	0x0	dma_uv_fifo_overflow_inten Enable the interrupt of dma fifo overflow of MIPI uv path or LVDS id1 path. 1'b0: Disable 1'b1: Enable
16	RW	0x0	dma_y_fifo_overflow_inten Enable the interrupt of dma fifo overflow of MIPI y path or LVDS id0 path. 1'b0: Disable 1'b1: Enable
15	RW	0x0	frame1_dma_end_id3_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable
14	RW	0x0	frame0_dma_end_id3_inten Enable the interrupt of end of even frame for ID3. 1'b0: Disable 1'b1: Enable
13	RW	0x0	frame1_dma_end_id2_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable
12	RW	0x0	frame0_dma_end_id2_inten Enable the interrupt of end of even frame for ID2. 1'b0: Disable 1'b1: Enable
11	RW	0x0	frame1_dma_end_id1_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable
10	RW	0x0	frame0_dma_end_id1_inten Enable the interrupt of end of even frame for ID1. 1'b0: Disable 1'b1: Enable
9	RW	0x0	frame1_dma_end_id0_inten Enable the interrupt of end of odd frame for ID0. 1'b0: Disable 1'b1: Enable
8	RW	0x0	frame0_dma_end_id0_inten Enable the interrupt of end of even frame for ID0. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
7	RW	0x0	frame1_start_id3_inten Enable the interrupt of start of odd frame for ID3. 1'b0: Disable 1'b1: Enable
6	RW	0x0	frame0_start_id3_inten Enable the interrupt of start of even frame for ID3. 1'b0: Disable 1'b1: Enable
5	RW	0x0	frame1_start_id2_inten Enable the interrupt of start of odd frame for ID2. 1'b0: Disable 1'b1: Enable
4	RW	0x0	frame0_start_id2_inten Enable the interrupt of start of even frame for ID2. 1'b0: Disable 1'b1: Enable
3	RW	0x0	frame1_start_id1_inten Enable the interrupt of start of odd frame for ID1. 1'b0: Disable 1'b1: Enable
2	RW	0x0	frame0_start_id1_inten Enable the interrupt of start of even frame for ID1. 1'b0: Disable 1'b1: Enable
1	RW	0x0	frame1_start_id0_inten Enable the interrupt of start of odd frame for ID0. 1'b0: Disable 1'b1: Enable
0	RW	0x0	frame0_start_id0_inten Enable the interrupt of start of even frame for ID0. 1'b0: Disable 1'b1: Enable

VICAP_MIPI0_LVDS0_INTSTAT

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	W1C	0x0	size_err_id3_intst 1'b0: No interrupt 1'b1: Interrupt
26	W1C	0x0	size_err_id2_intst 1'b0: No interrupt 1'b1: Interrupt
25	W1C	0x0	size_err_id1_intst 1'b0: No interrupt 1'b1: Interrupt
24	W1C	0x0	size_err_id0_intst 1'b0: No interrupt 1'b1: Interrupt
23	W1C	0x0	line_id3_intst 1'b0: No interrupt 1'b1: Interrupt
22	W1C	0x0	line_id2_intst 1'b0: No interrupt 1'b1: Interrupt

Bit	Attr	Reset Value	Description
21	W1C	0x0	line_id1_intst 1'b0: No interrupt 1'b1: Interrupt
20	W1C	0x0	line_id0_intst 1'b0: No interrupt 1'b1: Interrupt
19	W1C	0x0	csi2rx_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
18	W1C	0x0	bandwidth_lack_intst 1'b0: No interrupt 1'b1: Interrupt
17	W1C	0x0	dma_uv_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
16	W1C	0x0	dma_y_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
15	W1C	0x0	frame1_dma_end_id3_intst 1'b0: No interrupt 1'b1: Interrupt
14	W1C	0x0	frame0_dma_end_id3_intst 1'b0: No interrupt 1'b1: Interrupt
13	W1C	0x0	frame1_dma_end_id2_intst 1'b0: No interrupt 1'b1: Interrupt
12	W1C	0x0	frame0_dma_end_id2_intst 1'b0: No interrupt 1'b1: Interrupt
11	W1C	0x0	frame1_dma_end_id1_intst 1'b0: No interrupt 1'b1: Interrupt
10	W1C	0x0	frame0_dma_end_id1_intst 1'b0: No interrupt 1'b1: Interrupt
9	W1C	0x0	frame1_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
8	W1C	0x0	frame0_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
7	W1C	0x0	frame1_start_id3_intst 1'b0: No interrupt 1'b1: Interrupt
6	W1C	0x0	frame0_start_id3_intst 1'b0: No interrupt 1'b1: Interrupt
5	W1C	0x0	frame1_start_id2_intst 1'b0: No interrupt 1'b1: Interrupt
4	W1C	0x0	frame0_start_id2_intst 1'b0: No interrupt 1'b1: Interrupt

Bit	Attr	Reset Value	Description
3	W1C	0x0	frame1_start_id1_intst 1'b0: No interrupt 1'b1: Interrupt
2	W1C	0x0	frame0_start_id1_intst 1'b0: No interrupt 1'b1: Interrupt
1	W1C	0x0	frame1_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt
0	W1C	0x0	frame0_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt

VICAP_MIPI0_LVDS0_LINE_INT_NUM_ID0_1

Address: Operational Base + offset (0x017C)

Bit	Attr	Reset Value	Description
31	RW	0x0	line_int_mode_id1 1'b0: One-time mode 1'b1: Circular mode
30	RO	0x0	reserved
29:16	RW	0x0040	line_int_num_id1 For one-time mode,if line_int_num_id1=100,then channel 1 receive 100th line,the line_id1_intst will be 1. For circular mode ,if line_int_num_id1=100,then channel 1 receive 100th line200th line300th line.....the line_id1_intst will be 1.
15	RW	0x0	line_int_mode_id0 1'b0: One-time mode 1'b1: Circular mode
14	RO	0x0	reserved
13:0	RW	0x0040	line_int_num_id0 For one-time mode,if line_int_num_id0=100,then channel 0 receive 100th line,the line_id0_intst will be 1. For circular mode ,if line_int_num_id0=100,then channel 0 receive 100th line200th line300th line.....the line_id0_intst will be 1.

VICAP_MIPI0_LVDS0_LINE_INT_NUM_ID2_3

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31	RW	0x0	line_int_mode_id3 1'b0: One-time mode 1'b1: Circular mode
30	RO	0x0	reserved
29:16	RW	0x0040	line_int_num_id3 For one-time mode,if line_int_num_id3=100,then channel 3 receive 100th line,the line_id3_intst will be 1. For circular mode ,if line_int_num_id3=100,then channel 3 receive 100th line200th line300th line.....the line_id3_intst will be 1.
15	RW	0x0	line_int_mode_id2 1'b0: One-time mode 1'b1: Circular mode
14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0040	line_int_num_id2 For one-time mode,if line_int_num_id2=100,then channel 2 receive 100th line,the line_id2_intst will be 1. For circular mode ,if line_int_num_id2=100,then channel 2 receive 100th line200th line300th line.....the line_id2_intst will be 1.

VICAP_MIPI0_LVDS0_LINE_CNT_ID0_1

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_cnt_id1 Current line count for id1.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_line_cnt_id0 Current line count for id0.

VICAP_MIPI0_LVDS0_LINE_CNT_ID2_3

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_cnt_id3 Current line count for id3.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_line_cnt_id2 Current line count for id2.

VICAP_MIPI0_LVDS0_ID0_CROP_START

Address: Operational Base + offset (0x018C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id0 The start y coordinate for id0.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id0 The start x coordinate for id0,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_LVDS0_ID1_CROP_START

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id1 The start y coordinate for id1.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	sw_start_x_id1 The start x coordinate for id1,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_LVDS0_ID2_CROP_START

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id2 The start y coordinate for id2.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id2 The start x coordinate for id2,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_LVDS0_ID3_CROP_START

Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id3 The start y coordinate for id3.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id3 The start x coordinate for id3,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI0_FRAME_NUM_VC0

Address: Operational Base + offset (0x019C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc0 The frame number of frame end of virtual channel 0.
15:0	RO	0x0000	ro_frame_num_start_vc0 The frame number of frame start of virtual channel 0.

VICAP_MIPI0_FRAME_NUM_VC1

Address: Operational Base + offset (0x01A0)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc1 The frame number of frame end of virtual channel 1.
15:0	RO	0x0000	ro_frame_num_start_vc1 The frame number of frame start of virtual channel 1.

VICAP_MIPI0_FRAME_NUM_VC2

Address: Operational Base + offset (0x01A4)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc2 The frame number of frame end of virtual channel 2.
15:0	RO	0x0000	ro_frame_num_start_vc2 The frame number of frame start of virtual channel 2.

VICAP_MIPI0_FRAME_NUM_VC3

Address: Operational Base + offset (0x01A8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc3 The frame number of frame end of virtual channel 3.
15:0	RO	0x0000	ro_frame_num_start_vc3 The frame number of frame start of virtual channel 3.

VICAP_MIPI0_ID0_EFFECT_CODE

Address: Operational Base + offset (0x01AC)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id0 Effect code of id0 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id0 Effect code of id0 for sony sensor.

VICAP_MIPI0_ID1_EFFECT_CODE

Address: Operational Base + offset (0x01B0)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id1 Effect code of id1 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id1 Effect code of id1 for sony sensor.

VICAP_MIPI0_ID2_EFFECT_CODE

Address: Operational Base + offset (0x01B4)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id2 Effect code of id2 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id2 Effect code of id2 for sony sensor.

VICAP_MIPI0_ID3_EFFECT_CODE

Address: Operational Base + offset (0x01B8)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id3 Effect code of id3 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id3 Effect code of id3 for sony sensor.

VICAP_MIPI0_ON_PAD_VALUE

Address: Operational Base + offset (0x01BC)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:0	RW	0x0000	sw_on_pad_value Padding value for ON sensor.

VICAP_MIPI0_LVDS0_SIZE_NUM_ID0

Address: Operational Base + offset (0x01C0)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id0 The id0 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id0 The id0 pixel number in one line.

VICAP_MIPI0_LVDS0_SIZE_NUM_ID1

Address: Operational Base + offset (0x01C4)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id1 The id1 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id1 The id1 pixel number in one line.

VICAP_MIPI0_LVDS0_SIZE_NUM_ID2

Address: Operational Base + offset (0x01C8)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id2 The id2 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id2 The id2 pixel number in one line.

VICAP_MIPI0_LVDS0_SIZE_NUM_ID3

Address: Operational Base + offset (0x01CC)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id3 The id3 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id3 The id3 pixel number in one line.

VICAP_LVDS0_ID0_CTRL0

Address: Operational Base + offset (0x01D0)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28:27	RW	0x0	sw_lvds_yuv_out_order_id0 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id0 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id0 The write ddr type of id0. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id0 The parse type of id0. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id0 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id0 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id0 Enable dma transport id0. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id0 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id0 Only be used when the sw_lvds_mode_id0=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id0 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id0.
9:8	RW	0x0	sw_lvds_main_lane_id0 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id0 Lane enable for id0. eg : 4'b0011 represent lane 0/1 is enable.

Bit	Attr	Reset Value	Description
3:1	RW	0x0	sw_lvds_mode_id0 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id0 Enable to capture lvds id0. 1'b0: Disable 1'b1: Enable

VICAP_LVDS0_ID1_CTRL0

Address: Operational Base + offset (0x01D4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id1 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id1 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id1 The write ddr type of id1. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id1 The parse type of id1. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id1 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id1 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id1 Enable dma transport id1. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
14	RW	0x0	sw_lvds_align_id1 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id1 Only be used when the sw_lvds_mode_id1=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id1 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id1.
9:8	RW	0x0	sw_lvds_main_lane_id1 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id1 Lane enable for id1. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id1 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id1 Enable to capture lvds id1. 1'b0: Disable 1'b1: Enable

VICAP_LVDS0_ID2_CTRL0

Address: Operational Base + offset (0x01D8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id2 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id2 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id2 The write ddr type of id2. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved

Bit	Attr	Reset Value	Description
21:19	RW	0x0	sw_lvds_parse_type_id2 The parse type of id2. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id2 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id2 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id2 Enable dma transport id2. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id2 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id2 Only be used when the sw_lvds_mode_id2=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id2 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id2.
9:8	RW	0x0	sw_lvds_main_lane_id2 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id2 Lane enable for id2. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id2 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id2 Enable to capture lvds id2. 1'b0: Disable 1'b1: Enable

VICAP_LVDS0_ID3_CTRL0

Address: Operational Base + offset (0x01DC)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28:27	RW	0x0	sw_lvds_yuv_out_order_id3 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id3 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id3 The write ddr type of id3. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id3 The parse type of id3. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id3 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id3 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id3 Enable dma transport id3. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id3 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id3 Only be used when the sw_lvds_mode_id3=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id3 Only be used when the sw_lvds_mode_id3=3'b011. Choose the fid for id3.
9:8	RW	0x0	sw_lvds_main_lane_id3 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id3 Lane enable for id3. eg : 4'b0011 represent lane 0/1 is enable.

Bit	Attr	Reset Value	Description
3:1	RW	0x0	sw_lvds_mode_id3 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id3 Enable to capture lvds id3. 1'b0: Disable 1'b1: Enable

VICAP_LVDS0_SAV_EAV_ACT0_ID0

Address: Operational Base + offset (0x01E0)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id0 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id0 LVDS path sync code of sav_act0.

VICAP_LVDS0_SAV_EAV_BLK0_ID0

Address: Operational Base + offset (0x01E4)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id0 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id0 LVDS path sync code of sav_blk0.

VICAP_LVDS0_SAV_EAV_ACT1_ID0

Address: Operational Base + offset (0x01E8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id0 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id0 LVDS path sync code of sav_act1.

VICAP_LVDS0_SAV_EAV_BLK1_ID0

Address: Operational Base + offset (0x01EC)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id0 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id0 LVDS path sync code of sav_blk1.

VICAP_LVDS0_SAV_EAV_ACT0_ID1

Address: Operational Base + offset (0x01F0)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id1 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id1 LVDS path sync code of sav_act0.

VICAP_LVDS0_SAV_EAV_BLK0_ID1

Address: Operational Base + offset (0x01F4)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id1 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id1 LVDS path sync code of sav_blk0.

VICAP_LVDS0_SAV_EAV_ACT1_ID1

Address: Operational Base + offset (0x01F8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id1 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id1 LVDS path sync code of sav_act1.

VICAP_LVDS0_SAV_EAV_BLK1_ID1

Address: Operational Base + offset (0x01FC)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id1 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id1 LVDS path sync code of sav_blk1.

VICAP_LVDS0_SAV_EAV_ACT0_ID2

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id2 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id2 LVDS path sync code of sav_act0.

VICAP_LVDS0_SAV_EAV_BLK0_ID2

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id2 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	sw_lvds_sav_blk0_id2 LVDS path sync code of sav_blk0.

VICAP_LVDS0_SAV_EAV_ACT1_ID2

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id2 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id2 LVDS path sync code of sav_act1.

VICAP_LVDS0_SAV_EAV_BLK1_ID2

Address: Operational Base + offset (0x020C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id2 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id2 LVDS path sync code of sav_blk1.

VICAP_LVDS0_SAV_EAV_ACT0_ID3

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id3 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id3 LVDS path sync code of sav_act0.

VICAP_LVDS0_SAV_EAV_BLK0_ID3

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id3 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id3 LVDS path sync code of sav_blk0.

VICAP_LVDS0_SAV_EAV_ACT1_ID3

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id3 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id3 LVDS path sync code of sav_act1.

VICAP_LVDS0_SAV_EAV_BLK1_ID3

Address: Operational Base + offset (0x021C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id3 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id3 LVDS path sync code of sav_blk1.

VICAP_MIPI1_ID0_CTRL0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id0 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id0 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id0 Enable dma transport id0. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id0 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id0 Select command mode for id0. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id0 If sw_mipi0_on_hdr_mode_id0=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id0=1, id0 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id0 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id0 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id0 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU

Bit	Attr	Reset Value	Description
17:16	RW	0x0	sw_yuyv_in_order_id0 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id0 Data type for id0.
9:8	RW	0x0	sw_vc_id0 Virtual channel for id0.
7:5	RW	0x0	sw_wrddr_type_id0 The write ddr type of id0. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id0 Enable to crop for id0. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id0 The parse type of id0. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id0 Enable to capture id0. 1'b0: Disable 1'b1: Enable

VICAP_MIPI1_LVDS1_ID0_CTRL1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id0 Height for id0.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id0 Width for id0.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id0 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_ID1_CTRL0

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id1 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id1 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id1 Enable dma transport id1. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id1 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id1 Select command mode for id1. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id1 If sw_mipi0_on_hdr_mode_id1=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id1=1, id1 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id1 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id1 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id1 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id1 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id1 Data type for id1.
9:8	RW	0x0	sw_vc_id1 Virtual channel for id1.

Bit	Attr	Reset Value	Description
7:5	RW	0x0	sw_wrddr_type_id1 The write ddr type of id1. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id1 Enable to crop for id1. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id1 The parse type of id1. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id1 Enable to capture id1. 1'b0: Disable 1'b1: Enable

VICAP_MIPI1_LVDS1_ID1_CTRL1

Address: Operational Base + offset (0x030C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id1 Height for id1.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id1 Width for id1.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id1 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_ID2_CTRL0

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id2 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id2 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).

Bit	Attr	Reset Value	Description
28	RW	0x0	sw_dma_en_id2 Enable dma transport id2. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id2 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id2 Select command mode for id2. 1'b0: Not command mode 1'b1: Command mode
25:24	RW	0x0	sw_on_hdr_line_cnt_id2 If sw_mipi0_on_hdr_mode_id2=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id2=1, id2 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id2 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id2 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id2 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id2 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id2 Data type for id2.
9:8	RW	0x0	sw_vc_id2 Virtual channel for id2.
7:5	RW	0x0	sw_wrddr_type_id2 The write ddr type of id2. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id2 Enable to crop for id2. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3:1	RW	0x0	sw_parse_type_id2 The parse type of id2. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id2 Enable to capture id2. 1'b0: Disable 1'b1: Enable

VICAP_MIPI1_LVDS1_ID2_CTRL1

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id2 Height for id2.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id2 Width for id2.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id2 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_ID3_CTRL0

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_vc_hdr_id_main_id3 The id that is the first coming. Note: This register is available when sw_hdr_mode_idx=2'b00 and sw_vc_hdr_protect=1'b1.
29	RW	0x0	sw_vc_hdr_protect_id3 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly. Note: This register is available when sw_hdr_mode_idx=2'b00. And sw_vc_hdr_id_main must be configured correctly(the first coming id).
28	RW	0x0	sw_dma_en_id3 Enable dma transport id3. 1'b0: Disable 1'b1: Enable
27	RW	0x0	sw_align_id3 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
26	RW	0x0	sw_command_mode_en_id3 Select command mode for id3. 1'b0: Not command mode 1'b1: Command mode

Bit	Attr	Reset Value	Description
25:24	RW	0x0	sw_on_hdr_line_cnt_id3 If sw_mipi0_on_hdr_mode_id3=1(2),the line count value will circulate between 0~1(2).And if sw_mipi0_on_hdr_line_cnt_id3=1, id3 channel will capture the line which line count value equal to 1.
23:22	RW	0x0	sw_on_hdr_mode_id3 2'b00: Normal mode(no HDR) 2'b01: 2 frames HDR 2'b10: 3 frames HDR 2'b11: Reserved
21:20	RW	0x0	sw_hdr_mode_id3 2'b00: HDR distinguished by virtual channel 2'b01: HDR distinguished by line number(ON HDR mode) 2'b10: HDR distinguished by first 4pixel(SONY HDR mode) 2'b11: Reserved
19:18	RW	0x0	sw_yuyv_out_order_id3 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
17:16	RW	0x0	sw_yuyv_in_order_id3 2'b00 :UYVY(CSI-2 standard) 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
15:10	RW	0x00	sw_dt_id3 Data type for id3.
9:8	RW	0x0	sw_vc_id3 Virtual channel for id3.
7:5	RW	0x0	sw_wrddr_type_id3 The write ddr type of id3. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
4	RW	0x0	sw_crop_en_id3 Enable to crop for id3. 1'b0: Disable 1'b1: Enable
3:1	RW	0x0	sw_parse_type_id3 The parse type of id3. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: For raw14(data will be clipped to 12bit) 3'b100: For yuv422 8bit Others: Reserved
0	RW	0x0	sw_cap_en_id3 Enable to capture id3. 1'b0: Disable 1'b1: Enable

VICAP_MIPI1_LVDS1_ID3_CTRL1

Address: Operational Base + offset (0x031C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_height_id3 Height for id3.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_width_id3 Width for id3.if sw_wrddr_type is rgb888,then the width is equal to the number of bytes(not pixel). If sw_crop_en_id3 is enable the width value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_LVDS1_CTRL

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_drop_frame_main_id The id that is the last coming in HDR mode.
29:24	RW	0x00	sw_drop_frame_m Receive m frames,and drop n frames.(m>0,n>0)
23:21	RW	0x0	sw_drop_frame_n Receive m frames,and drop n frames.(m>0,n>0)
20	RW	0x0	sw_drop_frame_en_all 1'b0: Disable drop all id frame 1'b1: Enable drop all id frame.
19:18	RW	0x0	sw_soft_rst_mode 2'b00: Soft reset and protect toisp path frame integrity; 2'b01: Soft reset and protect toisp path frame integrity and pull down sw_cap_en automatically; 2'b10: Soft reset directly; 2'b11: Reserved.
17	RW	0x0	sw_soft_rst 1'b0: Not reset 1'b1: Reset the VICAP MIPI1 path(except MMU/AXI MASTER/REG FILE)
16	RW	0x0	sw_dma_rst Write 1 will reset VICAP MIPI1 path dma. When this bit change to 0,dma is reseted completely.
15:13	RW	0x0	sw_press_value Press value.
12	RW	0x0	sw_press_en 1'b0: Disable press 1'b1: Enable press
11	RW	0x0	sw_drop_frame_en_id3 1'b0: Disable drop id3 frame 1'b1: Enable drop id3 frame.
10	RW	0x0	sw_drop_frame_en_id2 1'b0: Disable drop id2 frame 1'b1: Enable drop id2 frame.
9	RW	0x0	sw_drop_frame_en_id1 1'b0: Disable drop id1 frame 1'b1: Enable drop id1 frame.
8	RW	0x0	sw_drop_frame_en_id0 1'b0: Disable drop id0 frame 1'b1: Enable drop id0 frame.

Bit	Attr	Reset Value	Description
7:5	RW	0x0	sw_hurry_value Hurry value.
4	RW	0x0	sw_hurry_en 1'b0: Disable hurry 1'b1: Enable hurry
3	RW	0x0	sw_mipi_lvds_sel 1'b0: MIPI path 1'b1: LVDS path
2:1	RW	0x0	sw_water_line 2'b00: 75% 2'b01: 50% 2'b10: 25% 2'b11: 0%
0	RW	0x0	sw_cap_en 1'b0: Disable capture all id 1'b1: Enable capture all id.

VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID0

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id0 First address of even frame for ID0 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID0

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id0 First address of odd frame for ID0 Y path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID0

Address: Operational Base + offset (0x032C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id0 First address of even frame for ID0 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID0

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id0 First address of odd frame for ID0 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_VLW_ID0

Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id0 Virtual line width of even frame for ID0 path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID1

Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id1 First address of even frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID1

Address: Operational Base + offset (0x033C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id1 First address of odd frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID1

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id1 First address of even frame for ID1 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID1

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id1 First address of odd frame for ID1 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_VLW_ID1

Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id1 Virtual line width of even frame for ID1 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID2

Address: Operational Base + offset (0x034C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id2 First address of even frame for ID2 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID2

Address: Operational Base + offset (0x0350)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id2 First address of odd frame for ID2 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID2

Address: Operational Base + offset (0x0354)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id2 First address of even frame for ID2 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID2

Address: Operational Base + offset (0x0358)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id0 First address of odd frame for ID2 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_VLW_ID2

Address: Operational Base + offset (0x035C)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id2 Virtual line width of even frame for ID2 path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_Y_ID3

Address: Operational Base + offset (0x0360)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_y_id3 First address of even frame for ID3 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_Y_ID3

Address: Operational Base + offset (0x0364)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_y_id3 First address of odd frame for ID3 Y/RAW/RGB path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME0_ADDR_UV_ID3

Address: Operational Base + offset (0x0368)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_uv_id3 First address of even frame for ID3 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_FRAME1_ADDR_UV_ID3

Address: Operational Base + offset (0x036C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_uv_id3 First address of odd frame for ID3 UV path(must be aligned to double word).

VICAP_MIPI1_LVDS1_VLW_ID3

Address: Operational Base + offset (0x0370)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_id3 Virtual line width of even frame for ID3 path(must be aligned to double word).

VICAP_MIPI1_LVDS1_INTEN

Address: Operational Base + offset (0x0374)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RW	0x0	size_err_id3_inten 1'b0: Disable 1'b1: Enable
26	RW	0x0	size_err_id2_inten 1'b0: Disable 1'b1: Enable
25	RW	0x0	size_err_id1_inten 1'b0: Disable 1'b1: Enable
24	RW	0x0	size_err_id0_inten 1'b0: Disable 1'b1: Enable
23	RW	0x0	line_id3_inten 1'b0: Disable 1'b1: Enable
22	RW	0x0	line_id2_inten 1'b0: Disable 1'b1: Enable
21	RW	0x0	line_id1_inten 1'b0: Disable 1'b1: Enable
20	RW	0x0	line_id0_inten 1'b0: Disable 1'b1: Enable
19	RW	0x0	csi2rx_fifo_overflow_inten 1'b0: Disable 1'b1: Enable
18	RW	0x0	bandwidth_lack_inten 1'b0: Disable 1'b1: Enable
17	RW	0x0	dma_uv_fifo_overflow_inten Enable the interrupt of dma fifo overflow of MIPI uv path or LVDS id1 path. 1'b0: Disable 1'b1: Enable
16	RW	0x0	dma_y_fifo_overflow_inten Enable the interrupt of dma fifo overflow of MIPI y path or LVDS id0 path. 1'b0: Disable 1'b1: Enable
15	RW	0x0	frame1_dma_end_id3_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable
14	RW	0x0	frame0_dma_end_id3_inten Enable the interrupt of end of even frame for ID3. 1'b0: Disable 1'b1: Enable
13	RW	0x0	frame1_dma_end_id2_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
12	RW	0x0	frame0_dma_end_id2_inten Enable the interrupt of end of even frame for ID2. 1'b0: Disable 1'b1: Enable
11	RW	0x0	frame1_dma_end_id1_inten Enable the interrupt of end of odd frame for ID1. 1'b0: Disable 1'b1: Enable
10	RW	0x0	frame0_dma_end_id1_inten Enable the interrupt of end of even frame for ID1. 1'b0: Disable 1'b1: Enable
9	RW	0x0	frame1_dma_end_id0_inten Enable the interrupt of end of odd frame for ID0. 1'b0: Disable 1'b1: Enable
8	RW	0x0	frame0_dma_end_id0_inten Enable the interrupt of end of even frame for ID0. 1'b0: Disable 1'b1: Enable
7	RW	0x0	frame1_start_id3_inten Enable the interrupt of start of odd frame for ID3. 1'b0: Disable 1'b1: Enable
6	RW	0x0	frame0_start_id3_inten Enable the interrupt of start of even frame for ID3. 1'b0: Disable 1'b1: Enable
5	RW	0x0	frame1_start_id2_inten Enable the interrupt of start of odd frame for ID2. 1'b0: Disable 1'b1: Enable
4	RW	0x0	frame0_start_id2_inten Enable the interrupt of start of even frame for ID2. 1'b0: Disable 1'b1: Enable
3	RW	0x0	frame1_start_id1_inten Enable the interrupt of start of odd frame for ID1. 1'b0: Disable 1'b1: Enable
2	RW	0x0	frame0_start_id1_inten Enable the interrupt of start of even frame for ID1. 1'b0: Disable 1'b1: Enable
1	RW	0x0	frame1_start_id0_inten Enable the interrupt of start of odd frame for ID0. 1'b0: Disable 1'b1: Enable
0	RW	0x0	frame0_start_id0_inten Enable the interrupt of start of even frame for ID0. 1'b0: Disable 1'b1: Enable

VICAP_MIPI1_LVDS1_INTSTAT

Address: Operational Base + offset (0x0378)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	W1C	0x0	size_err_id3_intst 1'b0: No interrupt 1'b1: Interrupt
26	W1C	0x0	size_err_id2_intst 1'b0: No interrupt 1'b1: Interrupt
25	W1C	0x0	size_err_id1_intst 1'b0: No interrupt 1'b1: Interrupt
24	W1C	0x0	size_err_id0_intst 1'b0: No interrupt 1'b1: Interrupt
23	W1C	0x0	line_id3_intst 1'b0: No interrupt 1'b1: Interrupt
22	W1C	0x0	line_id2_intst 1'b0: No interrupt 1'b1: Interrupt
21	W1C	0x0	line_id1_intst 1'b0: No interrupt 1'b1: Interrupt
20	W1C	0x0	line_id0_intst 1'b0: No interrupt 1'b1: Interrupt
19	W1C	0x0	csi2rx_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
18	W1C	0x0	bandwidth_lack_intst 1'b0: No interrupt 1'b1: Interrupt
17	W1C	0x0	dma_uv_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
16	W1C	0x0	dma_y_fifo_overflow_intst 1'b0: No interrupt 1'b1: Interrupt
15	W1C	0x0	frame1_dma_end_id3_intst 1'b0: No interrupt 1'b1: Interrupt
14	W1C	0x0	frame0_dma_end_id3_intst 1'b0: No interrupt 1'b1: Interrupt
13	W1C	0x0	frame1_dma_end_id2_intst 1'b0: No interrupt 1'b1: Interrupt
12	W1C	0x0	frame0_dma_end_id2_intst 1'b0: No interrupt 1'b1: Interrupt
11	W1C	0x0	frame1_dma_end_id1_intst 1'b0: No interrupt 1'b1: Interrupt
10	W1C	0x0	frame0_dma_end_id1_intst 1'b0: No interrupt 1'b1: Interrupt

Bit	Attr	Reset Value	Description
9	W1C	0x0	frame1_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
8	W1C	0x0	frame0_dma_end_id0_intst 1'b0: No interrupt 1'b1: Interrupt
7	W1C	0x0	frame1_start_id3_intst 1'b0: No interrupt 1'b1: Interrupt
6	W1C	0x0	frame0_start_id3_intst 1'b0: No interrupt 1'b1: Interrupt
5	W1C	0x0	frame1_start_id2_intst 1'b0: No interrupt 1'b1: Interrupt
4	W1C	0x0	frame0_start_id2_intst 1'b0: No interrupt 1'b1: Interrupt
3	W1C	0x0	frame1_start_id1_intst 1'b0: No interrupt 1'b1: Interrupt
2	W1C	0x0	frame0_start_id1_intst 1'b0: No interrupt 1'b1: Interrupt
1	W1C	0x0	frame1_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt
0	W1C	0x0	frame0_start_id0_intst 1'b0: No interrupt 1'b1: Interrupt

VICAP_MIPI1_LVDS1_LINE_INT_NUM_ID0_1

Address: Operational Base + offset (0x037C)

Bit	Attr	Reset Value	Description
31	RW	0x0	line_int_mode_id1 1'b0: One-time mode 1'b1: Circular mode
30	RO	0x0	reserved
29:16	RW	0x0040	line_int_num_id1 For one-time mode,if line_int_num_id1=100,then channel 1 receive 100th line,the line_id1_intst will be 1. For circular mode ,if line_int_num_id1=100,then channel 1 receive 100th line200th line300th line.....the line_id1_intst will be 1.
15	RW	0x0	line_int_mode_id0 1'b0: One-time mode 1'b1: Circular mode
14	RO	0x0	reserved
13:0	RW	0x0040	line_int_num_id0 For one-time mode,if line_int_num_id0=100,then channel 0 receive 100th line,the line_id0_intst will be 1. For circular mode ,if line_int_num_id0=100,then channel 0 receive 100th line200th line300th line.....the line_id0_intst will be 1.

VICAP_MIPI1_LVDS1_LINE_INT_NUM_ID2_3

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31	RW	0x0	line_int_mode_id3 1'b0: One-time mode 1'b1: Circular mode
30	RO	0x0	reserved
29:16	RW	0x0040	line_int_num_id3 For one-time mode,if line_int_num_id3=100,then channel 3 receive 100th line,the line_id3_intst will be 1. For circular mode ,if line_int_num_id3=100,then channel 3 receive 100th line200th line300th line.....the line_id3_intst will be 1.
15	RW	0x0	line_int_mode_id2 1'b0: One-time mode 1'b1: Circular mode
14	RO	0x0	reserved
13:0	RW	0x0040	line_int_num_id2 For one-time mode,if line_int_num_id2=100,then channel 2 receive 100th line,the line_id2_intst will be 1. For circular mode ,if line_int_num_id2=100,then channel 2 receive 100th line200th line300th line.....the line_id2_intst will be 1.

VICAP_MIPI1_LVDS1_LINE_CNT_ID0_1

Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_cnt_id1 Current line count for id1.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_line_cnt_id0 Current line count for id0.

VICAP_MIPI1_LVDS1_LINE_CNT_ID2_3

Address: Operational Base + offset (0x0388)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	line_cnt_id3 Current line count for id3.
15:14	RO	0x0	reserved
13:0	RW	0x0000	line_cnt_id2 Current line count for id2.

VICAP_MIPI1_LVDS1_ID0_CROP_START

Address: Operational Base + offset (0x038C)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id0 The start y coordinate for id0.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	sw_start_x_id0 The start x coordinate for id0,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_LVDS1_ID1_CROP_START

Address: Operational Base + offset (0x0390)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id1 The start y coordinate for id1.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id1 The start x coordinate for id1,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_LVDS1_ID2_CROP_START

Address: Operational Base + offset (0x0394)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id2 The start y coordinate for id2.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id2 The start x coordinate for id2,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_LVDS1_ID3_CROP_START

Address: Operational Base + offset (0x0398)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y_id3 The start y coordinate for id3.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x_id3 The start x coordinate for id3,if sw_wrddr_type is rgb888,then the start x is measured in byte. The start x value must be 8 aligned when sw_wrddr_type is raw8/yuv422, must be 4 aligned when sw_wrddr_type is raw10/raw12,and must be 24 aligned when sw_wrddy_type is rgb888.

VICAP_MIPI1_FRAME_NUM_VC0

Address: Operational Base + offset (0x039C)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc0 The frame number of frame end of virtual channel 0.
15:0	RO	0x0000	ro_frame_num_start_vc0 The frame number of frame start of virtual channel 0.

VICAP_MIPI1_FRAME_NUM_VC1

Address: Operational Base + offset (0x03A0)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc1 The frame number of frame end of virtual channel 1.
15:0	RO	0x0000	ro_frame_num_start_vc1 The frame number of frame start of virtual channel 1.

VICAP_MIPI1_FRAME_NUM_VC2

Address: Operational Base + offset (0x03A4)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc2 The frame number of frame end of virtual channel 2.
15:0	RO	0x0000	ro_frame_num_start_vc2 The frame number of frame start of virtual channel 2.

VICAP_MIPI1_FRAME_NUM_VC3

Address: Operational Base + offset (0x03A8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	ro_frame_num_end_vc3 The frame number of frame end of virtual channel 3.
15:0	RO	0x0000	ro_frame_num_start_vc3 The frame number of frame start of virtual channel 3.

VICAP_MIPI1_ID0_EFFECT_CODE

Address: Operational Base + offset (0x03AC)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id0 Effect code of id0 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id0 Effect code of id0 for sony sensor.

VICAP_MIPI1_ID1_EFFECT_CODE

Address: Operational Base + offset (0x03B0)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id1 Effect code of id1 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id1 Effect code of id1 for sony sensor.

VICAP_MIPI1_ID2_EFFECT_CODE

Address: Operational Base + offset (0x03B4)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:16	RW	0x0000	sw_sony_effect_code1_id2 Effect code of id2 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id2 Effect code of id2 for sony sensor.

VICAP_MIPI1_ID3_EFFECT_CODE

Address: Operational Base + offset (0x03B8)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_sony_effect_code1_id3 Effect code of id3 for sony sensor.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_sony_effect_code0_id3 Effect code of id3 for sony sensor.

VICAP_MIPI1_ON_PAD_VALUE

Address: Operational Base + offset (0x03BC)

Bit	Attr	Reset Value	Description
31:14	RO	0x00000	reserved
13:0	RW	0x0000	sw_on_pad_value Padding value for ON sensor.

VICAP_MIPI1_LVDS1_SIZE_NUM_ID0

Address: Operational Base + offset (0x03C0)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id0 The id0 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id0 The id0 pixel number in one line.

VICAP_MIPI1_LVDS1_SIZE_NUM_ID1

Address: Operational Base + offset (0x03C4)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id1 The id1 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id1 The id1 pixel number in one line.

VICAP_MIPI1_LVDS1_SIZE_NUM_ID2

Address: Operational Base + offset (0x03C8)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id2 The id2 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id2 The id2 pixel number in one line.

VICAP_MIPI1_LVDS1_SIZE_NUM_ID3

Address: Operational Base + offset (0x03CC)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RO	0x0000	ro_line_num_id3 The id3 line number in one frame.
15:14	RO	0x0	reserved
13:0	RO	0x0000	ro_pix_num_id3 The id3 pixel number in one line.

VICAP_LVDS1_ID0_CTRL0

Address: Operational Base + offset (0x03D0)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id0 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id0 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id0 The write ddr type of id0. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id0 The parse type of id0. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id0 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id0 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id0 Enable dma transport id0. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id0 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	sw_lvds_hdr_frame_id0 Only be used when the sw_lvds_mode_id0=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id0 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id0.
9:8	RW	0x0	sw_lvds_main_lane_id0 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id0 Lane enable for id0. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id0 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id0 Enable to capture lvds id0. 1'b0: Disable 1'b1: Enable

VICAP_LVDS1_ID1_CTRL0

Address: Operational Base + offset (0x03D4)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id1 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id1 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id1 The write ddr type of id1. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id1 The parse type of id1. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved

Bit	Attr	Reset Value	Description
18:17	RW	0x0	sw_lvds_hdr_id_main_id1 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id1 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id1 Enable dma transport id1. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id1 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id1 Only be used when the sw_lvds_mode_id1=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id1 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id1.
9:8	RW	0x0	sw_lvds_main_lane_id1 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id1 Lane enable for id1. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id1 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id1 Enable to capture lvds id1. 1'b0: Disable 1'b1: Enable

VICAP_LVDS1_ID2_CTRL0

Address: Operational Base + offset (0x03D8)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id2 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id2 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU

Bit	Attr	Reset Value	Description
24:22	RW	0x0	sw_lvds_wrddr_type_id2 The write ddr type of id2. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id2 The parse type of id2. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id2 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id2 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id2 Enable dma transport id2. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id2 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id2 Only be used when the sw_lvds_mode_id2=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id2 Only be used when the sw_lvds_mode_id0=3'b011. Choose the fid for id2.
9:8	RW	0x0	sw_lvds_main_lane_id2 Choose the lane to parse the sync code
7:4	RW	0x0	sw_lvds_lane_en_id2 Lane enable for id2. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id2 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id2 Enable to capture lvds id2. 1'b0: Disable 1'b1: Enable

VICAP_LVDS1_ID3_CTRL0

Address: Operational Base + offset (0x03DC)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	RW	0x0	sw_lvds_yuv_out_order_id3 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
26:25	RW	0x0	sw_lvds_yuv_in_order_id3 2'b00 :UYVY 2'b01 :VYUY 2'b10 :YUYV 2'b11 :YVYU
24:22	RW	0x0	sw_lvds_wrddr_type_id3 The write ddr type of id3. 3'b000: Raw-compact 3'b001: Raw-uncompact 3'b010: YUV packet 3'b011: YUV 400 3'b100: YUV 422sp 3'b101: YUV 420sp Others: Reserved
21:19	RW	0x0	sw_lvds_parse_type_id3 The parse type of id3. 3'b000: For raw8/rgb888 3'b001: For raw10 3'b010: For raw12 3'b011: Reserved 3'b100: For yuv422 8bit Others: Reserved
18:17	RW	0x0	sw_lvds_hdr_id_main_id3 The id that is the first coming. Note: This register is available when sw_hdr_protect=1'b1.
16	RW	0x0	sw_lvds_hdr_protect_id3 1'b0: Not ensure hdr long exposure frame is captured firstly; 1'b1: Ensure hdr long exposure frame is captured firstly.
15	RW	0x0	sw_lvds_dma_en_id3 Enable dma transport id3. 1'b0: Disable 1'b1: Enable
14	RW	0x0	sw_lvds_align_id3 1'b0: Low alignment(raw10/12 will in [9:0]/[11:0]) 1'b1: Hign alignment(raw10/12 will in [15:6]/[15:4] Only be used when non-compacted mode.
13	RO	0x0	reserved
12	RW	0x0	sw_lvds_hdr_frame_id3 Only be used when the sw_lvds_mode_id3=3'b010. 1'b0: 2 frames hdr 1'b1: 3 frames hdr
11:10	RW	0x0	sw_lvds_fid_id3 Only be used when the sw_lvds_mode_id3=3'b011. Choose the fid for id3.
9:8	RW	0x0	sw_lvds_main_lane_id3 Choose the lane to parse the sync code

Bit	Attr	Reset Value	Description
7:4	RW	0x0	sw_lvds_lane_en_id3 Lane enable for id3. eg : 4'b0011 represent lane 0/1 is enable.
3:1	RW	0x0	sw_lvds_mode_id3 There are 4 modes. 3'b000: Ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk 3'b001: Fs-le...ls-fe 3'b010: Sony dol hdr pattern 1 3'b011: Sony dol hdr pattern 2 other: Reserverd
0	RW	0x0	sw_lvds_cap_en_id3 Enable to capture lvds id3. 1'b0: Disable 1'b1: Enable

VICAP_LVDS1_SAV_EAV_ACT0_ID0

Address: Operational Base + offset (0x03E0)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id0 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id0 LVDS path sync code of sav_act0.

VICAP_LVDS1_SAV_EAV_BLK0_ID0

Address: Operational Base + offset (0x03E4)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id0 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id0 LVDS path sync code of sav_blk0.

VICAP_LVDS1_SAV_EAV_ACT1_ID0

Address: Operational Base + offset (0x03E8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id0 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id0 LVDS path sync code of sav_act1.

VICAP_LVDS1_SAV_EAV_BLK1_ID0

Address: Operational Base + offset (0x03EC)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id0 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id0 LVDS path sync code of sav_blk1.

VICAP_LVDS1_SAV_EAV_ACT0_ID1

Address: Operational Base + offset (0x03F0)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id1 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id1 LVDS path sync code of sav_act0.

VICAP_LVDS1_SAV_EAV_BLK0_ID1

Address: Operational Base + offset (0x03F4)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id1 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id1 LVDS path sync code of sav_blk0.

VICAP_LVDS1_SAV_EAV_ACT1_ID1

Address: Operational Base + offset (0x03F8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id1 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id1 LVDS path sync code of sav_act1.

VICAP_LVDS1_SAV_EAV_BLK1_ID1

Address: Operational Base + offset (0x03FC)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id1 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id1 LVDS path sync code of sav_blk1.

VICAP_LVDS1_SAV_EAV_ACT0_ID2

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id2 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id2 LVDS path sync code of sav_act0.

VICAP_LVDS1_SAV_EAV_BLK0_ID2

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27:16	RW	0x000	sw_lvds_eav_blk0_id2 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id2 LVDS path sync code of sav_blk0.

VICAP_LVDS1_SAV_EAV_ACT1_ID2

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id2 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act1_id2 LVDS path sync code of sav_act1.

VICAP_LVDS1_SAV_EAV_BLK1_ID2

Address: Operational Base + offset (0x040C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id2 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id2 LVDS path sync code of sav_blk1.

VICAP_LVDS1_SAV_EAV_ACT0_ID3

Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act0_id3 LVDS path sync code of eav_act0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_act0_id3 LVDS path sync code of sav_act0.

VICAP_LVDS1_SAV_EAV_BLK0_ID3

Address: Operational Base + offset (0x0414)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk0_id3 LVDS path sync code of eav_blk0.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk0_id3 LVDS path sync code of sav_blk0.

VICAP_LVDS1_SAV_EAV_ACT1_ID3

Address: Operational Base + offset (0x0418)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_act1_id3 LVDS path sync code of eav_act1.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	sw_lvds_sav_act1_id3 LVDS path sync code of sav_act1.

VICAP_LVDS1_SAV_EAV_BLK1_ID3

Address: Operational Base + offset (0x041C)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	sw_lvds_eav_blk1_id3 LVDS path sync code of eav_blk1.
15:12	RO	0x0	reserved
11:0	RW	0x000	sw_lvds_sav_blk1_id3 LVDS path sync code of sav_blk1.

VICAP_SCL_CH_CTRL

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:3	RW	0x00	sw_scl_mux_ch0 5'd0: Select MIPI0 ID0 5'd1: Select MIPI0 ID1 5'd2: Select MIPI0 ID2 5'd3: Select MIPI0 ID3 5'd4: Select MIPI1 ID0 5'd5: Select MIPI1 ID1 5'd6: Select MIPI1 ID2 5'd7: Select MIPI1 ID3 5'd24: Select DVP Others: Reserved
2:1	RW	0x0	sw_scl_mode_ch0 Scale mode for channel 0. 2'b00: 8 times scale 2'b01: 16 times scale 2'b10: 32 times scale Others: Reserved
0	RW	0x0	sw_scl_en_ch0 Enable scale channel0. 1'b0: Disable 1'b1: Enable

VICAP_SCL_CTRL

Address: Operational Base + offset (0x0704)

Bit	Attr	Reset Value	Description
31:17	RO	0x0000	reserved
16	RW	0x0	sw_soft_rst_ch0 Write 1 will reset VICAP SCALE channel 0 path. When this bit change to 0,scale is reseted completely.
15:13	RW	0x0	sw_press_value Press value.
12	RW	0x0	sw_press_en 1'b0: Disable press 1'b1: Enable press
11:8	RO	0x0	reserved
7:5	RW	0x0	sw_hurry_value Hurry value.

Bit	Attr	Reset Value	Description
4	RW	0x0	sw_hurry_en 1'b0: Disable hurry 1'b1: Enable hurry
3	RO	0x0	reserved
2:1	RW	0x0	sw_water_line 2'b00: 75% 2'b01: 50% 2'b10: 25% 2'b11: 0%
0	RO	0x0	reserved

VICAP_SCL_FRAME0_ADDR_CH0

Address: Operational Base + offset (0x0708)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame0_addr_ch0 First address of even frame for CH0 path(must be aligned to double word).

VICAP_SCL_FRAME1_ADDR_CH0

Address: Operational Base + offset (0x070C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_frame1_addr_ch0 First address of odd frame for CH0 path(must be aligned to double word).

VICAP_SCL_VLW_CH0

Address: Operational Base + offset (0x0710)

Bit	Attr	Reset Value	Description
31:20	RO	0x000	reserved
19:0	RW	0x00000	sw_vlw_ch0 Virtual line width of even frame for CH0 path(must be aligned to double word).

VICAP_SCL_CH0_BLACK_LEVEL

Address: Operational Base + offset (0x0738)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	sw_scl_ch0_black_11 The black level for scale channel0 bayer pattern 11 position.
23:16	RW	0x00	sw_scl_ch0_black_10 The black level for scale channel0 bayer pattern 10 position.
15:8	RW	0x00	sw_scl_ch0_black_01 The black level for scale channel0 bayer pattern 01 position.
7:0	RW	0x00	sw_scl_ch0_black_00 The black level for scale channel0 bayer pattern 00 position.

VICAP_TOISP0_CH_CTRL

Address: Operational Base + offset (0x0780)

Bit	Attr	Reset Value	Description
31:27	RO	0x00	reserved
26:25	RW	0x0	sw_toisp0_vc_hdr_ch_main Toisp0 vc hdr main channel.
24	RW	0x0	sw_toisp0_vc_hdr_protect Enable toisp0 vc hdr protect. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
23:19	RW	0x00	sw_toisp0_mux_ch2 5'd0: Select MIPI0 ID0 5'd1: Select MIPI0 ID1 5'd2: Select MIPI0 ID2 5'd3: Select MIPI0 ID3 5'd4: Select MIPI1 ID0 5'd5: Select MIPI1 ID1 5'd6: Select MIPI1 ID2 5'd7: Select MIPI1 ID3 5'd24: Select DVP Others: Reserved
18:17	RO	0x0	reserved
16	RW	0x0	sw_toisp0_en_ch2 Enable toisp0 channel2. 1'b0: Disable 1'b1: Enable
15:11	RW	0x00	sw_toisp0_mux_ch1 5'd0: Select MIPI0 ID0 5'd1: Select MIPI0 ID1 5'd2: Select MIPI0 ID2 5'd3: Select MIPI0 ID3 5'd4: Select MIPI1 ID0 5'd5: Select MIPI1 ID1 5'd6: Select MIPI1 ID2 5'd7: Select MIPI1 ID3 5'd24: Select DVP Others: Reserved
10:9	RO	0x0	reserved
8	RW	0x0	sw_toisp0_en_ch1 Enable toisp0 channel1. 1'b0: Disable 1'b1: Enable
7:3	RW	0x00	sw_toisp0_mux_ch0 5'd0: Select MIPI0 ID0 5'd1: Select MIPI0 ID1 5'd2: Select MIPI0 ID2 5'd3: Select MIPI0 ID3 5'd4: Select MIPI1 ID0 5'd5: Select MIPI1 ID1 5'd6: Select MIPI1 ID2 5'd7: Select MIPI1 ID3 5'd24: Select DVP Others: Reserved
2:1	RO	0x0	reserved
0	RW	0x0	sw_toisp0_en_ch0 Enable toisp0 channel0. 1'b0: Disable 1'b1: Enable

VICAP_TOISP0_CROP_SIZE

Address: Operational Base + offset (0x0784)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x01e0	sw_height The expected height of received image.

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13:0	RW	0x02d0	sw_width The expected width of received image.

VICAP_TOISP0_CROP_START

Address: Operational Base + offset (0x0788)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_start_y The vertical ordinate of the start point.
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_start_x The horizontal ordinate of the start point.

26.5 Interface Description

Table 26-2 VICAP Interface Description

Module Pin	Dir	Pad Name	IOMUX Setting
vicap_clkout	O	MIPI_REFCLK_OUT0/VICAP_CLKOUT_M0/GPIO3_C4_d	VIIOC_GPIO3C_IOMUX_SEL_H[2:0]=3'h1
		DSMAUDIO_N//PWM11_IR_M2/UART5_TX_M1/I2C3_SCL_M1/VICAP_CLKOUT_M1/LCD_CLK/GPIO1_D3_d	VENCIOC_GPIO1D_IOMUX_SEL_L[14:12]=3'h2
vicap_clkin	I	MIPI/LVDS_D0N/VICAP_CLKIN_M0/GPIO3_C2_d	VIIOC_GPIO3C_IOMUX_SEL_L[10:8]=3'h1
		UART3_TX_M1/UART5_RTSN_M1/PWM3_IR_M2/VICAP_CLKIN_M1/LCD_DEN/GPIO1_D0_d	VENCIOC_GPIO1D_IOMUX_SEL_L[2:0]=3'h2
vicap_href	I	MIPI/LVDS_D0P/VICAP_HSYNC_M0/GPIO3_C3_d	VIIOC_GPIO3C_IOMUX_SEL_L[14:12]=3'h1
		UART3_RX_M1/UART5_CTSN_M1/PWM10_M2/VICAP_HSYNC_M1/LCD_HSYNC/GPIO1_D1_d	VENCIOC_GPIO1D_IOMUX_SEL_L[6:4]=3'h2
vicap_vsync	I	VICAP_VSYNC_M0/GPIO3_C5_d	VIIOC_GPIO3C_IOMUX_SEL_H[6:4]=3'h1
		DSMAUDIO_P/PWM0_M1/SPI0_CS1n_M0/UART5_RX_M1/I2C3_SDA_M1/VICAP_VSYNC_M1/LCD_VSYNC/GPIO1_D2_d	VENCIOC_GPIO1D_IOMUX_SEL_L[10:8]=3'h2
vicap_data0	I	MIPI/LVDS_D3N/VICAP_D0_M0/GPIO3_B0_d	VIIOC_GPIO3B_IOMUX_SEL_L[2:0]=3'h1
		VICAP_D0_M1/AVS_ARM/PWM0_M0/GPIO1_A2_d	VENCIOC_GPIO1A_IOMUX_SEL_L[10:8]=3'h3
vicap_data1	I	MIPI/LVDS_D3P/VICAP_D1_M0/GPIO3_B1_d	VIIOC_GPIO3B_IOMUX_SEL_L[6:4]=3'h1
		VICAP_D1_M1/SPI1_CS1n_M0/PWM7_IR_M1/UART4_TX_M0/GPIO1_B1_d	VENCIOC_GPIO1B_IOMUX_SEL_L[6:4]=3'h4
vicap_data2	I	MIPI/LVDS_CK1N/VICAP_D2_M0/GPIO3_B2_d	VIIOC_GPIO3B_IOMUX_SEL_L[10:8]=3'h1
		SDMMC1_D1_M1/SPI0_CS0n_M0/PWM2_M2/VICAP_D2_M1/LCD_D7/GPIO1_C0_d	VENCIOC_GPIO1C_IOMUX_SEL_L[2:0]=3'h2
vicap_data3	I	MIPI/LVDS_CK1P/VICAP_D3_M0/GPIO3_B3_d	VIIOC_GPIO3B_IOMUX_SEL_L[14:12]=3'h1
		SDMMC1_D0_M1/SPI0_CLK_M0/PWM4_M2/VICAP_D3_M1/LCD_D6/GPIO1_C1_d	VENCIOC_GPIO1C_IOMUX_SEL_L[6:4]=3'h2
vicap_data4	I	MIPI/LVDS_D2N/VICAP_D4_M0/GPIO3_B4_d	VIIOC_GPIO3B_IOMUX_SEL_H[2:0]=3'h1
		SPI0_MOSI_M0/SDMMC1_CLK_M1/I2C4_SCL_M1/PWM5_M2/VICAP_D4_M1/LCD_D5/GPIO1_C2_d	VENCIOC_GPIO1C_IOMUX_SEL_L[10:8]=3'h2
vicap_data5	I	MIPI/LVDS_D2P/VICAP_D5_M0/GPIO3_B5_d	VIIOC_GPIO3B_IOMUX_SEL_H[6:4]=3'h1

Module Pin	Dir	Pad Name	IOMUX Setting
		SPI0_MISO_M0/SDMMC1_CMD_M1/I2C4_SDA_M1/PWM6_M2/VICAP_D5_M1/LCD_D4/GPIO1_C3_d	VENCIOC_GPIO1C_IOMUX_SEL_L[14:12]=3'h2
vicap_data6	I	MIPI/LVDS_D1N/VICAP_D6_M0/GPIO3_B6_d	VIIOC_GPIO3B_IOMUX_SEL_H[10:8]=3'h1
		SDMMC1_D3_M1/UART4_RX_M1/PWM8_M1/VICAP_D6_M1/LCD_D3/GPIO1_C4_d	VENCIOC_GPIO1C_IOMUX_SEL_H[2:0]=3'h2
vicap_data7	I	MIPI/LVDS_D1P/VICAP_D7_M0/GPIO3_B7_d	VIIOC_GPIO3B_IOMUX_SEL_H[14:12]=3'h1
		SDMMC1_D2_M1/UART4_TX_M1/PWM9_M1/VICAP_D7_M1/LCD_D2/GPIO1_C5_d	VENCIOC_GPIO1C_IOMUX_SEL_H[6:4]=3'h2
vicap_data8	I	MIPI/LVDS_CK0N/VICAP_D8_M0/GPIO3_C0_d	VIIOC_GPIO3C_IOMUX_SEL_L[2:0]=3'h1
		UART4_RTSN_M1/PWM10_M1/VICAP_D8_M1/LCD_D1/GPIO1_C6_d	VENCIOC_GPIO1C_IOMUX_SEL_H[10:8]=3'h2
vicap_data9	I	MIPI/LVDS_CK0P/VICAP_D9_M0/GPIO3_C1_d	VIIOC_GPIO3C_IOMUX_SEL_L[6:4]=3'h1
		UART4_CTSN_M1/PWM11_IR_M1/VICAP_D9_M1/LCD_D0/GPIO1_C7_d	VENCIOC_GPIO1C_IOMUX_SEL_H[14:12]=3'h2
vicap_data10	I	MIPI_REFCLK_OUT1/PWM7_IR_M2/VICAP_D10/GPIO3_C6_d	VIIOC_GPIO3C_IOMUX_SEL_H[10:8]=3'h1
vicap_data11	I	I2C4_SCL_M2/UART5_TX_M2/VICAP_D11/GPIO3_C7_d	VIIOC_GPIO3C_IOMUX_SEL_H[14:12]=3'h1
vicap_data12	I	I2C4_SDA_M2/UART5_RX_M2/VICAP_D12/GPIO3_D0_d	VIIOC_GPIO3D_IOMUX_SEL_L[2:0]=3'h1
vicap_data13	I	I2C3_SCL_M2/UART5_RTSN_M2/VICAP_D13/GPIO3_D1_d	VIIOC_GPIO3D_IOMUX_SEL_L[6:4]=3'h1
vicap_data14	I	I2C3_SDA_M2/UART5_CTSN_M2/VICAP_D14/GPIO3_D2_d	VIIOC_GPIO3D_IOMUX_SEL_L[10:8]=3'h1
vicap_data15	I	PWM1_M2/VICAP_D15/GPIO3_D3_d	VIIOC_GPIO3D_IOMUX_SEL_L[14:12]=3'h1

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.

26.6 Application Notes

26.6.1 DVP Interface

- The MSB 4 bits of SAV/EAV of BT.656/BT.1120 are configurable, and the default values of SAV/EAV abide by the agreement.
- The sw_only_sav_mode for BT.656/BT.1120 is more robust for the case that the EAV is wrong.
- User can read the register of ro_sync_header_0~ro_sync_header_7 to obtain the recent 8 SAV/EAV.
- The register sw_yuyv_in_order should be configured correctly. In the BT.1120 case 2'h0/2'h2 are the same for UV sequence and 2'h1/2'h3 are the same for VU sequence.
- In the dual-edge sampling case, the GRF_VI_CIF_CON[2] must be set to 1, and the register sw_dualedge_en must be set to 1. The clock phase could be adjusted by setting the GRF_VI_CIF_CON[10:3].
- User can select the M1 by setting the register GRF_VI_CIF_CON[0]=1.

User can invert the clkinm0 by setting the register GRF_VI_CIF_CON[1]=1. User can invert the clkinm1 by setting the register VENC_CIF_IO_WRAPPER[0]=1.

26.6.2 MIPI Interface

- The register sw_crop_en is suggested to be set to 1. The register sw_width/sw_start_x should be aligned to 4.
- In the sony identification code hdr mode, the register sw_sony_effect_code0/1 should be configured correctly according to the camera manual.
- In the line counter hdr mode, the register sw_on_pad_value should be configured correctly according to the camera manual.

- In the VC hdr mode, the register sw_vc_hdr_protect is suggested to be set to 1, and the register sw_vc_hdr_id_main should be configured as the first coming id that is the long exposure frame id.
- The register sw_cap_en_idx is recommended for static configuration, and the register sw_cap_en is recommended for dynamic configuration.
- The MIPI interface can receive DSI data, but there are following restrictions on DSI TX: (1) one row of data could not be split into multiple long packets; (2) multiple packets could not be concatenated within a single HS transmission.

26.6.3 LVDS Interface

- The sync code register (sw_lvds_sav_act0_id0,sw_lvds_eav_act0_id0...) must be configured correctly by the user.
- There are four LVDS sync mode can be selected by sw_lvds_mode_idx. Mode0: ls-le...fs-fe or sav_act-eav_act...sav_blk-eav_blk. Mode1:fs-le...ls-fe. Mode2: sony dol hdr pattern 1. Mode3:sony dol hdr pattern 2. And if the mode is mode2, the user also need configure the register sw_lvds_hdr_frame_idx. If the mode is mode3, the user also need configure the register sw_lvds_fid_idx.

26.6.4 Scale down

- The register sw_scl_chx_black_00/01/10/11 is the black level value for bayer pattern 00/01/10/11 position.
- The bit width of scale down output data is 16. When the max bit width of the scale down output data exceed 16, output data will be clipped. For example, 8 times scale down for RAW12, the bit width is 18, then the result will be clipped to 16 bits.
- When the width is not a multiple of double scale down times,then the result will be rounded up to an integer. For example, the image width is 4656 and 32 times scaling down is performed, the result width equals to (ceil(4656/64)*64)/32.
- When the height is not a multiple of double scale down times, then the result will be rounded down to an integer. For example, the image height is 1080 and 8 times scaling down is performed, the result height equals to (floor(1080/16)*16)/8.

26.6.5 TOISP

- The register sw_width/sw_start_x should be aligned to 4.
- Although there are three channels to ISP, user just need enable channel 0 and select the short exposure frame id.

26.6.6 DMA

- The start address/virtual stride should be aligned to 8 at least. And being aligned to 64 is good for DDR.
- User could shut down DMA transferring by setting the register sw_dma_en to 0.
- User could set sw_hurry_en/sw_press_en to 1 for dynamic adjusting the priority of VICAP. And the register sw_water_line is the triggering condition of pulling up hurry/press signal.
- The register sw_yuyv_out_order is effective when the register sw_wrddr_type is NV16/NV12/YUYV. And in the NV16/NV12 case, 2'h0/2'h2 are the same for UV sequence and 2'h1/2'h3 are the same for VU sequence.
- There are two modes for line counter interrupt.In one-time mode, if line_int_num =100, the line_intst will be 1 when receive 100th line. In circular mode, if line_int_num =100, the line_intst will be 1 when receive 100th line200th line300th line.....
- The DMA has two transfer strategies :1. transfer image according to the actual height of reception. 2. transfer image according to the configured height. Which strategy you choose depends the register sw_dma_adapt_en.

26.6.7 Reset

- DMA will be auto-reset when DMA goes wrong, for example fifo overflow. And DMA will transfer data from the next frame.
- DMA also could be soft-reset by the register sw_dma_rst.
- User could reset the entire data path (interface/dma) by setting the register sw_soft_rst to 1. And the scale/toisp module will be reset if the sw_scl_mux/sw_toisp_mux is setted to choose the path.

Chapter 27 Mailbox

27.1 Overview

The Mailbox module is a simple APB peripheral that allows CPU, MCU core to communicate with each other by writing operation to generate interrupt. The registers are accessible via APB interface.

The Mailbox has the following main features:

- Support APB interface
- Support four mailbox elements, each element includes one data word, one command word register and one flag bit that can represent one interrupt
- Support interrupts to CPU, and MCU core
- Provide 32 lock registers for software to use to indicate whether mailbox is occupied

27.2 Block Diagram

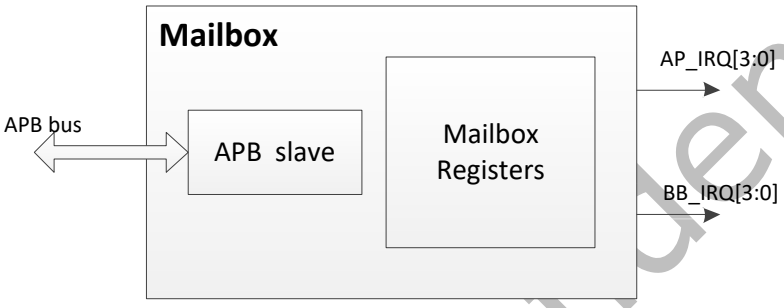


Fig. 27-1 Mailbox Block Diagram

27.3 Function Description

27.3.1 Mailbox

- Regard CPU as the "AP" side of the Mailbox. The four elements interrupt to CPU is:
  - Enabled when MAILBOX_B2A_INTEN is set to 1 and the responding IRQs is enabled in GIC.
  - Generated when there are writing operation to corresponding MAILBOX_B2A_CMD_i and MAILBOX_B2A_DAT_i orderly.
  - Cleared when writing 1 to corresponding MAILBOX_B2A_STATUS.
- Regard MCU core as the "BB" side of the Mailbox. The four elements interrupt to MCU core is:
  - Enabled when MAILBOX_A2B_INTEN is set to 1 and the responding IRQs is enabled in INTC of MCU.
  - Generated when there are writing operation to corresponding MAILBOX_A2B_CMD_i and MAILBOX_A2B_DAT_i orderly.
  - Cleared when writing 1 to corresponding MAILBOX_A2B_STATUS.
- You can also regard CPU as the "BB" side of the Mailbox and regard MCU core as the "AP" side of the Mailbox. The configuration flow is similar.

27.4 Register Description

27.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MAILBOX_A2B_INTEN	0x0000	W	0x00000000	AP to BB Interrupt Enable Register
MAILBOX_A2B_STATUS	0x0004	W	0x00000000	AP to BB Interrupt Status Register
MAILBOX_A2B_CMD_0	0x0008	W	0x00000000	AP to BB Command 0 Register
MAILBOX_A2B_DAT_0	0x000C	W	0x00000000	AP to BB Data 0 Register
MAILBOX_A2B_CMD_1	0x0010	W	0x00000000	AP to BB Command 1 Register
MAILBOX_A2B_DAT_1	0x0014	W	0x00000000	AP to BB Data 1 Register
MAILBOX_A2B_CMD_2	0x0018	W	0x00000000	AP to BB Command 2 Register
MAILBOX_A2B_DAT_2	0x001C	W	0x00000000	AP to BB Data 2 Register

Name	Offset	Size	Reset Value	Description
MAILBOX_A2B_CMD_3	0x0020	W	0x00000000	AP to BB Command 3 Register
MAILBOX_A2B_DAT_3	0x0024	W	0x00000000	AP to BB Data 3 Register
MAILBOX_B2A_INTEN	0x0028	W	0x00000000	BB to AP Interrupt Enable Register
MAILBOX_B2A_STATUS	0x002C	W	0x00000000	BB to AP Interrupt Status Register
MAILBOX_B2A_CMD_0	0x0030	W	0x00000000	BB to AP Command 0 Register
MAILBOX_B2A_DAT_0	0x0034	W	0x00000000	BB to AP Data 0 Register
MAILBOX_B2A_CMD_1	0x0038	W	0x00000000	BB to AP Command 1 Register
MAILBOX_B2A_DAT_1	0x003C	W	0x00000000	BB to AP Data 1 Register
MAILBOX_B2A_CMD_2	0x0040	W	0x00000000	BB to AP Command 2 Register
MAILBOX_B2A_DAT_2	0x0044	W	0x00000000	BB to AP Data 2 Register
MAILBOX_B2A_CMD_3	0x0048	W	0x00000000	BB to AP Command 3 Register
MAILBOX_B2A_DAT_3	0x004C	W	0x00000000	BB to AP Data 3 Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, **DW**-Double WORD (64 bits) access*

27.4.2 Detail Register Description

MAILBOX_A2B_INTEN

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX_A2B_STATUS

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	W1C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

Bit	Attr	Reset Value	Description
1	W1C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX_A2B_CMD_0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_A2B_DAT_0

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_A2B_CMD_1

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_A2B_DAT_1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_A2B_CMD_2

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_A2B_DAT_2

Address: Operational Base + offset (0x001C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_A2B_CMD_3

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_A2B_DAT_3

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_B2A_INTEN

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX_B2A_STATUS

Address: Operational Base + offset (0x002C)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3	W1C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
1	W1C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX_B2A_CMD_0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_B2A_DAT_0

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_B2A_CMD_1

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_B2A_DAT_1

Address: Operational Base + offset (0x003C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_B2A_CMD_2

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_B2A_DAT_2

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX_B2A_CMD_3

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX_B2A_DAT_3

Address: Operational Base + offset (0x004C)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

27.5 Application Notes

- It is recommended to read one ATOMIC_LOCK register first when using the Mailbox. The read value is 0 means it is available, and 1 means it has been automatically locked. Writing to the ATOMIC_LOCK register will clear this bit.
- Write to the CMD register before writing to the DAT register. If wrong order is used, then the interrupt cannot be generated successfully.
- If you want to clear the interrupt, you can read out the STATUS register and writing 1 to corresponding bit.

Chapter 28 Watchdog Timer (WDT)

28.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that caused by conflicting parts or programs in a SOC. The WDT would generate interrupt or reset signal when its counter reaches zero, then a reset controller would reset the system.

WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock can be chosen from 24MHz or 32KHz clock
- 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
- There are 3 WDTs: WDT, WDTS, WDT_PMU
- All the five WDTs can drive CRU to generate global software reset

28.2 Block Diagram

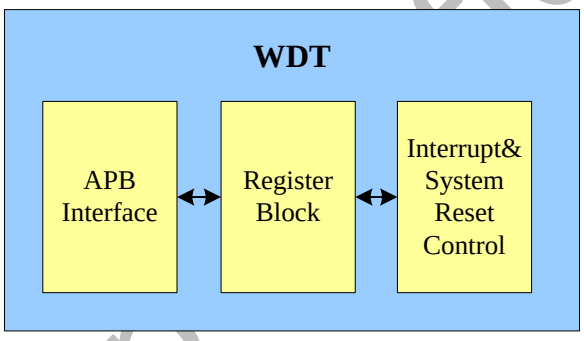


Fig. 28-1 WDT Block Diagram

WDT comprises with:

- 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.

28.3 Function Description

28.3.1 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

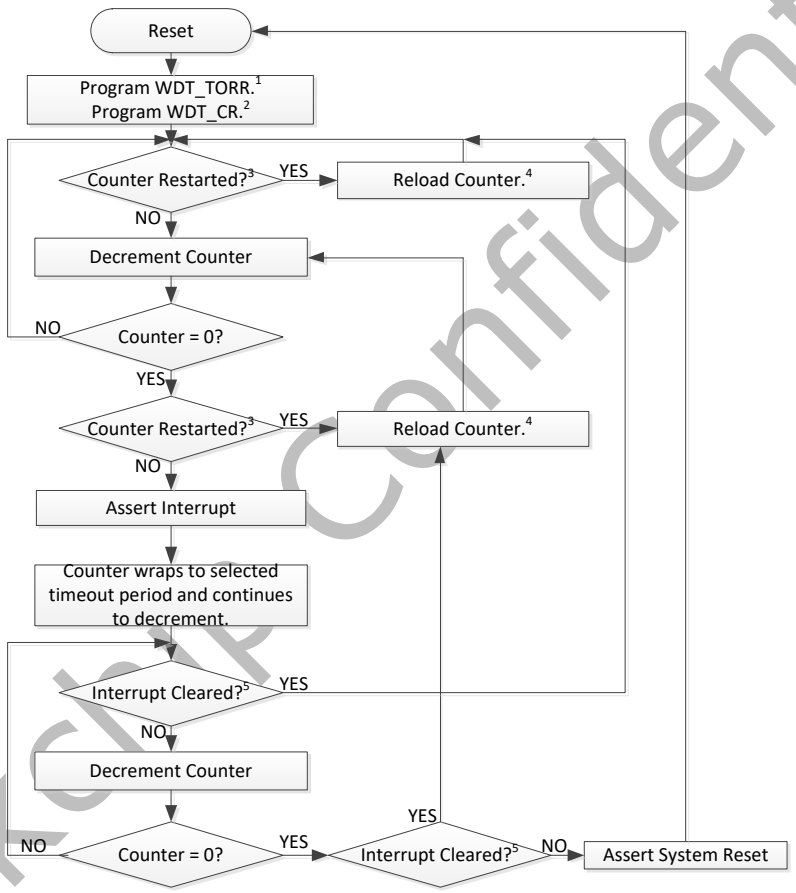
The WDT can be programmed to generate an interrupt (and then a system reset) when a

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.

28.3.2 Programming Sequence



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. 28-2 WDT Operation Flow (RMOD=1)

28.4 Register Description

28.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x00000008	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout Range Register
WDT_CCVR	0x0008	W	0x0000FFFF	Current Counter Value Register

Name	Offset	Size	Reset Value	Description
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, **DW**-Double WORD (64 bits) access*

28.4.2 Detail Registers Description

WDT_CR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:5	RO	0x0000000	reserved
4:2	RW	0x2	rst_pluse_length This is used to select the number of pclk cycles for which the system reset stays asserted. 3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles
1	RW	0x0	resp_mode Selects the output response generated to a timeout. 1'b0: Generate a system reset. 1'b1: 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	en This bit is used to enable and disable the WDT.When disabled, the counter dose not decrement .Thus, no interrupt or system reset are generated. Once this bit has been enabled, it can be cleared only by a system reset. 1'b0: WDT disabled. 1'b1: WDT enabled.

WDT_TORR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0000000	reserved
3:0	RW	0x0	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: 4'b0000: 0x0000ffff 4'b0001: 0x0001ffff 4'b0010: 0x0003ffff 4'b0011: 0x0007ffff 4'b0100: 0x000fffff 4'b0101: 0x001fffff 4'b0110: 0x003fffff 4'b0111: 0x007fffff 4'b1000: 0x00ffffff 4'b1001: 0x01ffffff 4'b1010: 0x03ffffff 4'b1011: 0x07ffffff

Bit	Attr	Reset Value	Description
			4'b1100: 0x0fffffff 4'b1101: 0x1fffffff 4'b1110: 0x3fffffff 4'b1111: 0x7fffffff

WDT_CCVR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000ffff	cur_cnt This register, when read, is the current value of the internal counter. This value is read coherently whenever it is read.

WDT_CRR

Address: Operational Base + offset (0x000C)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	reserved
7:0	WO	0x00	cnt_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	0x00000000	reserved
0	RO	0x0	status This register shows the interrupt status of the WDT. 1'b1: Interrupt is active regardless of polarity. 1'b0: Interrupt is inactive.

WDT_EOI

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:1	RO	0x00000000	reserved
0	RO	0x0	int_clr This can be used to clear the interrupt without restarting the watchdog counter.