Rockchip
RK3328
Technical Reference Manual Part1

Revision 1.1

Mar. 2017

Revision History

Date	Revision	Description
2017-3-20	1.1	Update
2017-02-06	1.0	Initial Release

Table of Content

Figure Index

Table Index

NOTICE

1. By using this document, you hereby unequivocally acknowledge that you have read and agreed to be bound by the contents of this notice.
2. Fuzhou Rockchip Electronics Co., Ltd. (“Rockchip”) may make changes to any information in this document at any time without any prior notice. The information herein is subject to change without notice. Do not finalize a design with this information.
3. Information in this document is provided in connection with Rockchip products.
4. THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OR CONDITION WITH RESPECT TO MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR NON-INFRINGEMENT.ROCKCHIP DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE NOR FOR ANY INFRINGEMENT OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE.
5. Rockchip products described in this document are not designed, intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility application.
6. Rockchip and Rockchip logo are trademarks or registered trademarks of Rockchip in China and other countries. All referenced brands, product names, service names and trademarks in this document are the property by their respective owners.

Chapter 1 System Overview

1.1 Address Mapping

RK3328 supports to boot from internal bootrom, which supports remap function by software programming. Remap is controlled by SGRF_SOC_CON2[10]. When remap is set to 1, the bootrom is mapped to address 0Xff080000 and internal memory is mapped to address 0Xffff0000.

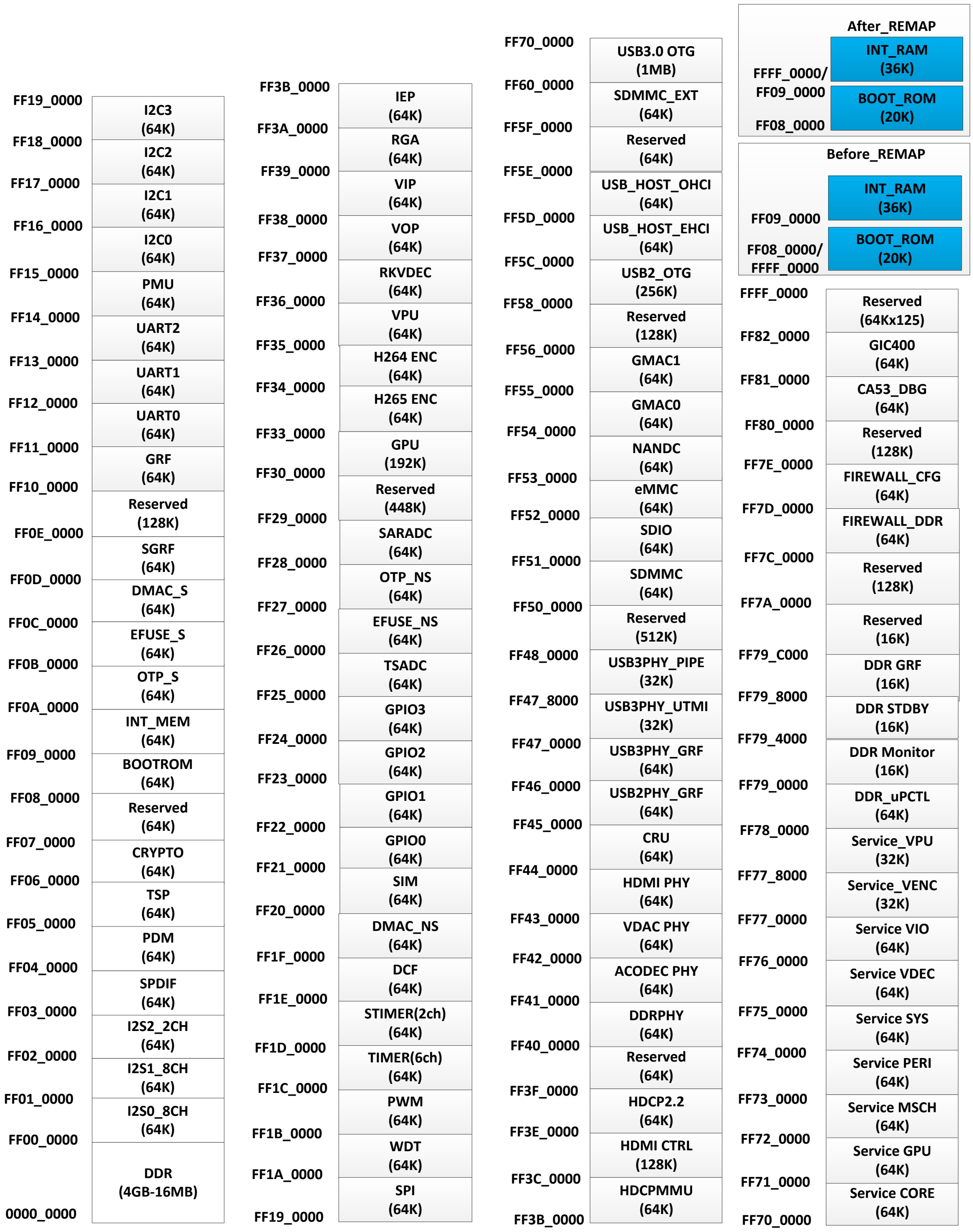


Fig. 1-1 RK3328 Address Mapping

1.2 System Boot

RK3328 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 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 supports.

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

Following figure shows RK3328 boot procedure flow.

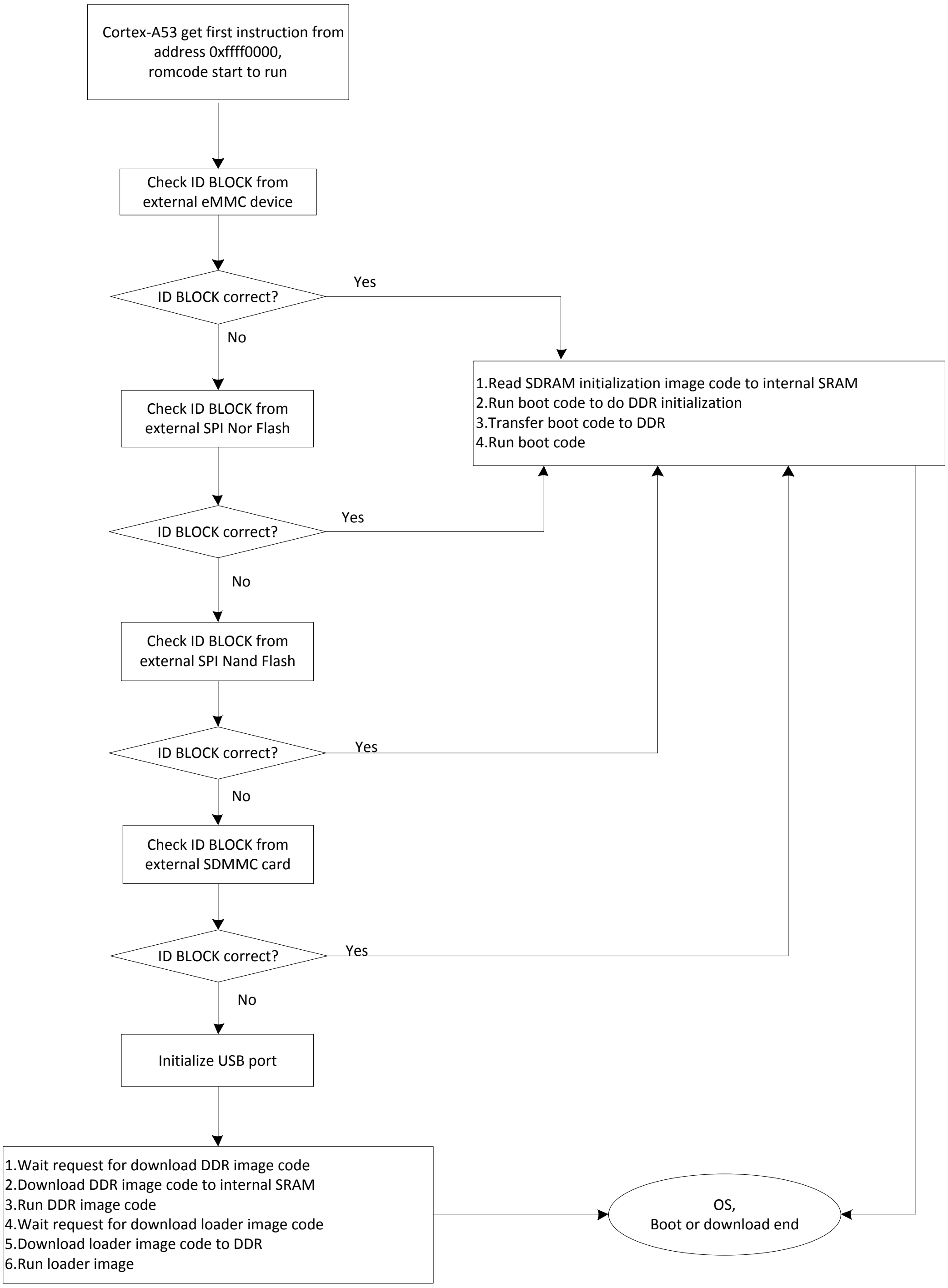


Fig. 1-2 RK3328 boot procedure flow

1.3 System Interrupt connection

RK3328 provides an general interrupt controller(GIC) for CPU, which has 128 SPI (shared peripheral interrupts) interrupt sources and 3 PPI(Private peripheral interrupt) interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to Chapter 9.

Table 1-1 RK3328 Interrupt connection list

IRQ Type	IRQ ID	Source(spi)	Polarity
SPI	32	(|bus_dmac_irq)	High level
	33	bus_dmac_irq_abort	High level
	34	dfi_alert_err_intr	High level
	35	upctl_awpoison_intr	High level
	36	sdmmc_ext_int	High level
	37	vop_intr_ddr	High level
	38	sdmmc_ext_dectn_in	High level
	39	rkvdec_m_dec_irq	High level
	40	upctl_arpoison_intr	High level
	41	vpu_xintdec_irq	High level
	42	sdmmc_ext_detectn_irq	High level
	43	vpu_mmu_irq	High level
	44	sdmmc_int	High level
	45	sdio_int	High level
	46	emmc_int	High level
	47	otp_int_ns	High level
	48	host0_ehci_int	High level
	49	host0_ohci_int	High level
	50	host0_arb_int	High level
	51	otp_int_s	High level
	52	ddrmon_int	High level
	53	gmac2phy_int	High level
	54	gmac2phy_pmt_int	High level
	55	otg_int	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	56	gmac2io_int	High level
	57	gmac2io_pmt_int	High level
	58	i2s0_8ch_intr	High level
	59	i2s1_8ch_intr	High level
	60	i2s2_2ch_intr	High level
	61	spdif_8ch_intr	High level
	62	crypto_int	High level
	63	iep_intr	High level
	64	vop_intr	High level
	65	rga_intr	High level
	66	hdcp_intr	High level
	67	hdmi_intr	High level
	68	rki2c0_int	High level
	69	rki2c1_int	High level
	70	rki2c2_int	High level
	71	rki2c3_int	High level
	72	wdt_intr	High level
	73	stimer_intr0	High level
	74	stimer_intr1	High level
	75	timer_intr0	High level
	76	timer_intr1	High level
	77	timer_intr2	High level
	78	timer_intr3	High level
	79	timer_intr4	High level
	80	timer_intr5	High level
	81	spi0_intr	High level
	82	rkpwm_int	High level
	83	gpio0_intr	High level
	84	gpio1_intr	High level
	85	gpio2_intr	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	86	gpio3_intr	High level
	87	uart0_intr	High level
	88	uart1_intr	High level
	89	uart2_intr	High level
	90	tsadc_int	High level
	91	usbphy_otg_bvalid_irq	High level
	92	usbphy_otg_id_irq	High level
	93	usbphy_otg_linestate_irq	High level
	94	usbphy_host_linestate_irq	High level
	95	sdmmc_detectn_irq	High level
	96	cif_intr	High level
	97	sdmmc_dectn_in_flt	High level
	98	usb3otg_host_legacy_smi_interrupt	High level
	99	usb3otg_int	High level
	100	usb3otg_host_sys_err	High level
	101	usb3otg_pme_generation	High level
	102	macphy_int	High level
	103	hdmi_intr_wakeup	High level
	104	tsp_int	High level
	105	sim_int	High level
	106	rkvdec_m_mmu_irq	High level
	107	usb3phy_bvalid_irq	High level
	108	usb3phy_id_irq	High level
	109	usb3phy_linestate_irq	High level
	110	usb3phy_rxdet_irq	High level
	111	efuse_int	High level
	112	saradc_int	High level
	113	tsp_int_mmu	High level
	114	pdm_int	High level
	115	hdmiphy_irq	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	116	dcf_done_int	High level
	117	dcf_error_int	High level
	118	pmu_int	High level
	119	irq_gpu_gpmmu	High level
	120	irq_gpu_pp0	High level
	121	irq_gpu_ppmmu0	High level
	122	irq_gpu_gp	High level
	123	irq_gpu_pp1	High level
	124	irq_gpu_ppmmu1	High level
	125	irq_gpu_pp	High level
	126	irq_gpu_pmu	High level
	127	rkvenc_h265_int	High level
	128	rkvenc_h265_mmu_int	High level
	129	rkvenc_h264_enc_int	High level
	130	rkvenc_h264_mmu_int	High level
	131	Reserved	High level
	132	npmuirq[0]	High level
	133	npmuirq[1]	High level
	134	npmuirq[2]	High level
	135	npmuirq[3]	High level
	136	nvcpumntirq[0]	High level
	137	nvcpumntirq[1]	High level
	138	nvcpumntirq[2]	High level
	139	nvcpumntirq[3]	High level
	140	ncommirq[0]	High level
	141	ncommirq[1]	High level
	142	ncommirq[2]	High level
	143	ncommirq[3]	High level
	144	naxierrirq	High level

1.4 System DMA hardware request connection

RK3328 provides one DMA controller inside the system. The trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC, please refer to Chapter 8.

Table 1-2 RK3328 DMAC Hardware request connection list

Req Number	Source	Polarity
0	I2S2_2ch tx	High level
1	I2S2_2ch rx	High level
2	Uart0 tx	High level
3	Uart0 rx	High level
4	Uart1 tx	High level
5	Uart1 rx	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	SPI tx	High level
9	SPI rx	High level
10	SPDIF	High level
11	I2S0_8ch tx	High level
12	I2S0_8ch rx	High level
13	pwm_tx	High level
14	I2S1_8ch_tx	High level
15	I2S1_8ch_rx	High level
16	pdm	High level

Chapter 2 Clock & Reset Unit (CRU)

2.1 Overview

The CRU is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. CRU generates system clocks from PLL output clock or external clock source, and generates system reset from external power-on-reset, watchdog timer reset or software reset.

CRU supports the following features:

- Compliance to the AMBA APB interface
- Embedded 5 PLLs
- Flexible selection of clock source
- Supports the respective gating of all clocks
- Supports the respective software reset of all modules

2.2 Block Diagram

CRU comprises with:

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

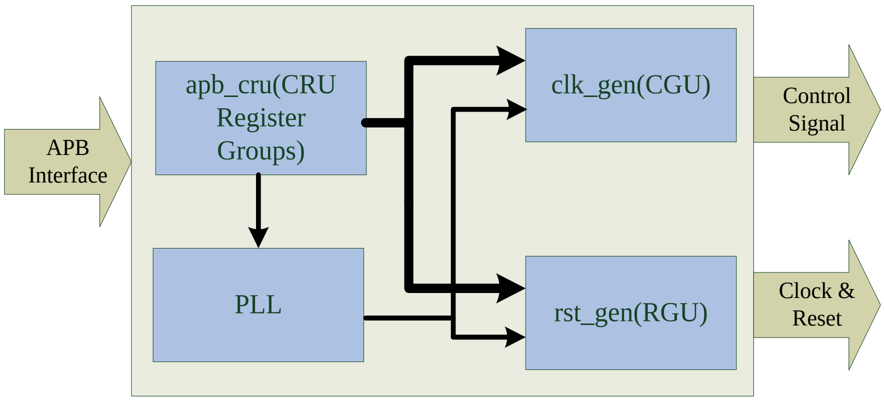


Fig. 2-1 CRU Block Diagram

2.3 System Reset Solution

The following diagram shows reset architecture.

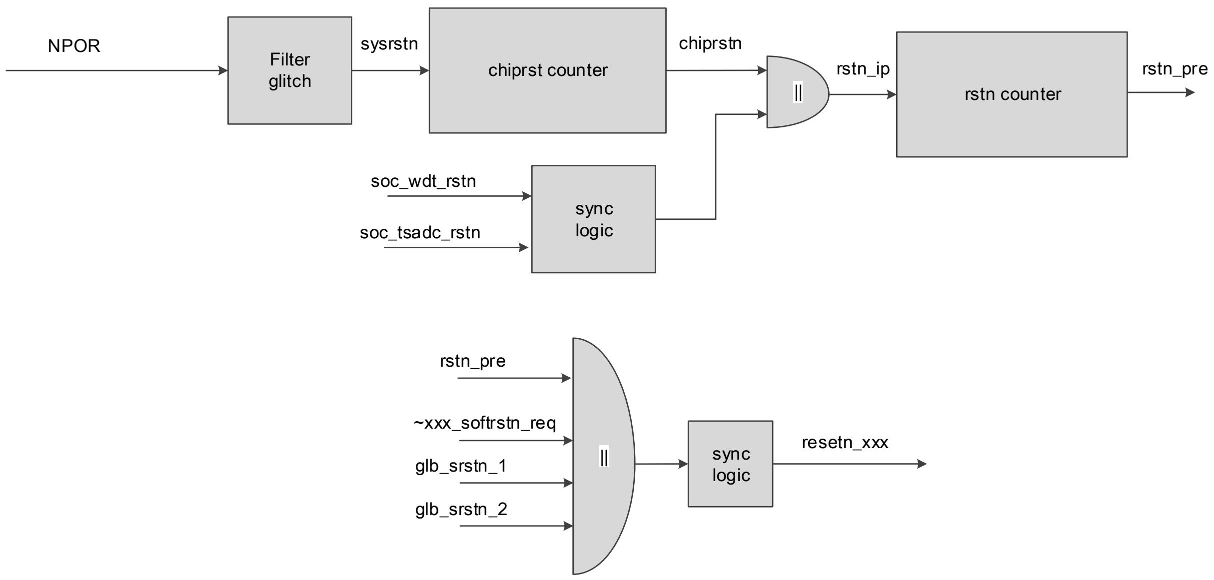


Fig. 2-2 Reset Architecture Diagram

Reset source of each reset signal includes hardware reset(NPOR), SoC watch dog reset(soc_wdt_rstn), SoC tsadc reset(soc_tsadc_rstn), software reset request(xxx_softrstn_req), global software reset1(glb_srstn_1), global software reset2(glb_srstn_2).
The 'xxx' of resetn_xxx and xxx_softrstn_req is the module name.
soc_wdt_rstn is the reset from watch-dog IP in the SoC.
glb_srstn_1 and glb_srstn_2 are the global software reset by programming CRU register. When writing register CRU_GLB_SRST_FST_VALUE as 0xfdb9, glb_srstn_1 will be asserted, and when writing register CRU_GLB_SRST_SND_VALUE as 0xeca8, glb_srstn_2 will be asserted. The two software resets will be self-cleared by hardware. glb_srstn_1 will reset the all logic, and glb_srstn_2 will reset the all logic except GRF and all GPIOs.

2.4 Function Description

There are 5 PLLs in the chip: ARM PLL, NEW PLL, DDR PLL, CODEC PLL and GENERAL PLL, and it supports only one crystal oscillator: 24MHz. Each PLL can only receive 24MHz oscillator.
These 5 PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz. When power on or changing PLL setting, we must force PLL into slow mode to ensure output stable clock.
To maximize the flexibility, some of clocks can select divider source from 5 PLLs. (Note: It's recommended to use NEW PLL instead of ARM PLL as arm clock source, because NEW PLL is near to ARM. And it's jitter is better than ARM PLL).
To provide some specific frequency, another solution is integrated: fractional divider. In order to guarantee the performance for divided clock, there is some usage limit, we can only get low frequency and divider factor must be larger than 20.
All clocks can be software gated and all resets can be software generated.

2.5 PLL Introduction

2.5.1 Overview

The chip uses 3.2GHz PLL for all the PLLs. The 3.2GHz PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter

performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments. By combining ultra-low jitter output clocks into a low power, low area, widely programmable design, we can greatly simplify an SoC by enabling a single macro to be used for all clocking applications in the system.
3.2GHz PLL supports the following features:

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

2.5.2 Block diagram

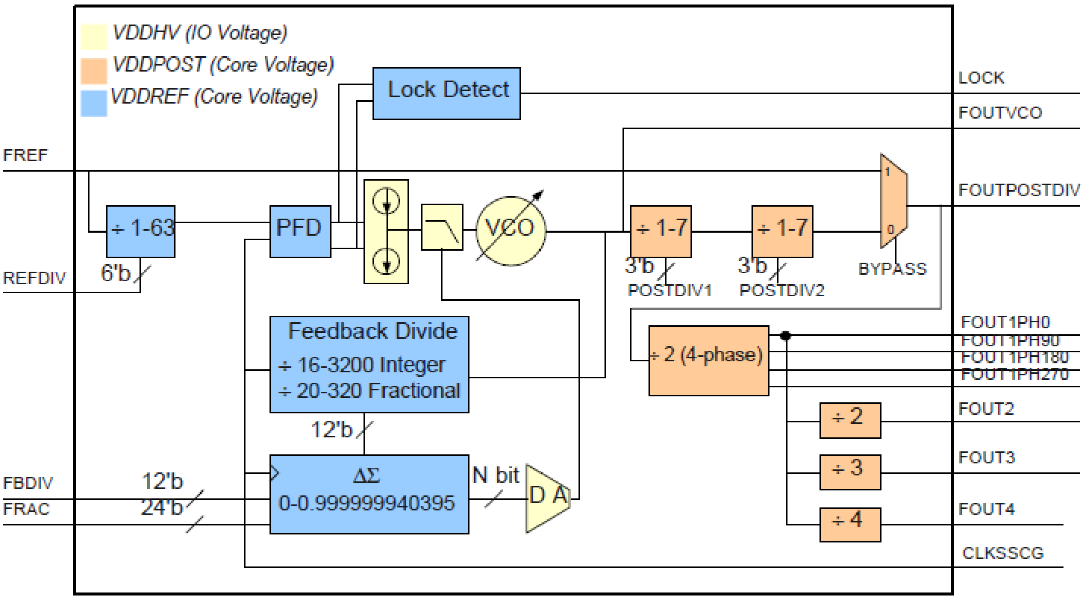


Fig. 2-3 PLL Block Diagram

How to calculate the PLL

The Fractional PLL output frequency can be calculated using some simple formulas. These formulas also embedded within the Fractional PLL Verilog model:
If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = FREF / REFDIV * FBDIV
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 224)
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
Where:
FOUTVCO = Fractional PLL non-divided output frequency
FOUTPOSTDIV = Fractional PLL divided output frequency (output of second post divider)
FREF = Fractional PLL input reference frequency
REFDIV = Fractional PLL input reference clock divider

FVCO = Frequency of internal VCO
FBDIV = Integer value programmed into feedback divide
FRAC = Fractional value programmed into DSM

Changing the PLL Programming

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

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

2.6 Register Description

2.6.1 Internal Address Mapping

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

2.6.2 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x00003064	APLL configuration register0
CRU_APLL_CON1	0x0004	W	0x00001041	APLL configuration register1
CRU_APLL_CON2	0x0008	W	0x00000001	APLL configuration register2
CRU_APLL_CON3	0x000c	W	0x00000007	APLL configuration register3
CRU_APLL_CON4	0x0010	W	0x00007f00	APLL configuration register4
CRU_DPLL_CON0	0x0020	W	0x00001096	DPLL configuration register0
CRU_DPLL_CON1	0x0024	W	0x00001042	DPLL configuration register1
CRU_DPLL_CON2	0x0028	W	0x00000001	DPLL configuration register2
CRU_DPLL_CON3	0x002c	W	0x00000007	DPLL configuration register3
CRU_DPLL_CON4	0x0030	W	0x00007f00	DPLL configuration register4
CRU_CPLL_CON0	0x0040	W	0x000020c8	CPLL configuration register0
CRU_CPLL_CON1	0x0044	W	0x00001043	CPLL configuration register1
CRU_CPLL_CON2	0x0048	W	0x00000001	CPLL configuration register2
CRU_CPLL_CON3	0x004c	W	0x00000007	CPLL configuration register3
CRU_CPLL_CON4	0x0050	W	0x00007f00	CPLL configuration register4
CRU_GPLL_CON0	0x0060	W	0x00001051	GPLL configuration register0

Name	Offset	Size	Reset Value	Description
CRU_GPLL_CON1	0x0064	W	0x00000042	GPLL configuration register1
CRU_GPLL_CON2	0x0068	W	0x00eb84f8	GPLL configuration register2
CRU_GPLL_CON3	0x006c	W	0x00000007	GPLL configuration register3
CRU_GPLL_CON4	0x0070	W	0x00007f00	GPLL configuration register4
CRU_CRU_MODE	0x0080	W	0x00000000	CRU_MODE
CRU_CRU_MISC	0x0084	W	0x0000a000	CRU_MISC
CRU_CRU_GLB_CNT_TH	0x0090	W	0x3a980064	CRU_GLB_CNT_TH
CRU_GLB_RST_ST	0x0094	W	0x00000000	GLB_RST_ST
CRU_GLB_SRST_SND_VALUE	0x0098	W	0x00000000	GLB_SRST_SND_VALUE
CRU_GLB_SRST_FST_VALUE	0x009c	W	0x00000000	GLB_SRST_FST_VALUE
CRU_NPLL_CON0	0x00a0	W	0x00003064	NPLL configuration register0
CRU_NPLL_CON1	0x00a4	W	0x00001041	NPLL configuration register1
CRU_NPLL_CON2	0x00a8	W	0x00000001	NPLL configuration register2
CRU_NPLL_CON3	0x00ac	W	0x00000007	NPLL configuration register3
CRU_NPLL_CON4	0x00b0	W	0x00007f00	NPLL configuration register4
CRU_CLKSEL_CON0	0x0100	W	0x00000300	Internal clock select and divide register0
CRU_CLKSEL_CON1	0x0104	W	0x00001113	Internal clock select and divide register1
CRU_CLKSEL_CON2	0x0108	W	0x00000003	Internal clock select and divide register2
CRU_CLKSEL_CON3	0x010c	W	0x00000000	Internal clock select and divide register3
CRU_CLKSEL_CON4	0x0110	W	0x00000780	Internal clock select and divide register4
CRU_CLKSEL_CON5	0x0114	W	0x00008000	Internal clock select and divide register5
CRU_CLKSEL_CON6	0x0118	W	0x0000000f	Internal clock select and divide register6
CRU_CLKSEL_CON7	0x011c	W	0x0bb8ea60	Internal clock select and divide register7
CRU_CLKSEL_CON8	0x0120	W	0x0000000f	Internal clock select and divide register8
CRU_CLKSEL_CON9	0x0124	W	0x0bb8ea60	Internal clock select and divide register9
CRU_CLKSEL_CON10	0x0128	W	0x0000000f	Internal clock select and divide register10
CRU_CLKSEL_CON11	0x012c	W	0x0bb8ea60	Internal clock select and divide register11
CRU_CLKSEL_CON12	0x0130	W	0x0000000f	Internal clock select and divide register12
CRU_CLKSEL_CON13	0x0134	W	0x0bb8ea60	Internal clock select and divide register13

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON14	0x0138	W	0x00000007	Internal clock select and divide register14
CRU_CLKSEL_CON15	0x013c	W	0x0bb8ea60	Internal clock select and divide register15
CRU_CLKSEL_CON16	0x0140	W	0x00000007	Internal clock select and divide register16
CRU_CLKSEL_CON17	0x0144	W	0x0bb8ea60	Internal clock select and divide register17
CRU_CLKSEL_CON18	0x0148	W	0x00000007	Internal clock select and divide register18
CRU_CLKSEL_CON19	0x014c	W	0x0bb8ea60	Internal clock select and divide register19
CRU_CLKSEL_CON20	0x0150	W	0x00008f04	Internal clock select and divide register20
CRU_CLKSEL_CON21	0x0154	W	0x00000400	Internal clock select and divide register21
CRU_CLKSEL_CON22	0x0158	W	0x000001e0	Internal clock select and divide register22
CRU_CLKSEL_CON23	0x015c	W	0x000001e0	Internal clock select and divide register23
CRU_CLKSEL_CON24	0x0160	W	0x00000707	Internal clock select and divide register24
CRU_CLKSEL_CON25	0x0164	W	0x00000242	Internal clock select and divide register25
CRU_CLKSEL_CON26	0x0168	W	0x0000000f	Internal clock select and divide register26
CRU_CLKSEL_CON27	0x016c	W	0x00000705	Internal clock select and divide register27
CRU_CLKSEL_CON28	0x0170	W	0x00000042	Internal clock select and divide register28
CRU_CLKSEL_CON29	0x0174	W	0x00000022	Internal clock select and divide register29
CRU_CLKSEL_CON30	0x0178	W	0x00000003	Internal clock select and divide register30
CRU_CLKSEL_CON31	0x017c	W	0x00000001	Internal clock select and divide register31
CRU_CLKSEL_CON32	0x0180	W	0x00000001	Internal clock select and divide register32
CRU_CLKSEL_CON33	0x0184	W	0x0000030b	Internal clock select and divide register33
CRU_CLKSEL_CON34	0x0188	W	0x00000707	Internal clock select and divide register34
CRU_CLKSEL_CON35	0x018c	W	0x00000707	Internal clock select and divide register35
CRU_CLKSEL_CON36	0x0190	W	0x00004242	Internal clock select and divide register36
CRU_CLKSEL_CON37	0x0194	W	0x00000242	Internal clock select and divide register37
CRU_CLKSEL_CON38	0x0198	W	0x0000c2dc	Internal clock select and divide register38
CRU_CLKSEL_CON39	0x019c	W	0x00000001	Internal clock select and divide register39
CRU_CLKSEL_CON40	0x01a0	W	0x00003113	Internal clock select and divide register40
CRU_CLKSEL_CON41	0x01a4	W	0x0bb8ea60	Internal clock select and divide register41
CRU_CLKSEL_CON42	0x01a8	W	0x00000013	Internal clock select and divide register42

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON43	0x01ac	W	0x00000003	Internal clock select and divide register43
CRU_CLKSEL_CON44	0x01b0	W	0x00000042	Internal clock select and divide register44
CRU_CLKSEL_CON45	0x01b4	W	0x0000001f	Internal clock select and divide register45
CRU_CLKSEL_CON46	0x01b8	W	0x00000000	Internal clock select and divide register46
CRU_CLKSEL_CON47	0x01bc	W	0x00000000	Internal clock select and divide register47
CRU_CLKSEL_CON48	0x01c0	W	0x00004201	Internal clock select and divide register48
CRU_CLKSEL_CON49	0x01c4	W	0x00000042	Internal clock select and divide register49
CRU_CLKSEL_CON50	0x01c8	W	0x00000042	Internal clock select and divide register50
CRU_CLKSEL_CON51	0x01cc	W	0x00000203	Internal clock select and divide register51
CRU_CLKSEL_CON52	0x01d0	W	0x0000021e	Internal clock select and divide register52
CRU_CLKGATE_CON0	0x0200	W	0x00000000	Internal clock gating register0
CRU_CLKGATE_CON1	0x0204	W	0x00000000	Internal clock gating register1
CRU_CLKGATE_CON2	0x0208	W	0x00000000	Internal clock gating register2
CRU_CLKGATE_CON3	0x020c	W	0x00000000	Internal clock gating register3
CRU_CLKGATE_CON4	0x0210	W	0x00000000	Internal clock gating register4
CRU_CLKGATE_CON5	0x0214	W	0x00000000	Internal clock gating register5
CRU_CLKGATE_CON6	0x0218	W	0x00000000	Internal clock gating register6
CRU_CLKGATE_CON7	0x021c	W	0x00000000	Internal clock gating register7
CRU_CLKGATE_CON8	0x0220	W	0x00000000	Internal clock gating register8
CRU_CLKGATE_CON9	0x0224	W	0x00000000	Internal clock gating register9
CRU_CLKGATE_CON10	0x0228	W	0x00000000	Internal clock gating register10
CRU_CLKGATE_CON11	0x022c	W	0x00000000	Internal clock gating register11
CRU_CLKGATE_CON12	0x0230	W	0x00000000	Internal clock gating register12
CRU_CLKGATE_CON13	0x0234	W	0x00000000	Internal clock gating register13
CRU_CLKGATE_CON14	0x0238	W	0x00000000	Internal clock gating register14
CRU_CLKGATE_CON15	0x023c	W	0x00000000	Internal clock gating register15
CRU_CLKGATE_CON16	0x0240	W	0x00000000	Internal clock gating register16
CRU_CLKGATE_CON17	0x0244	W	0x00000000	Internal clock gating register17
CRU_CLKGATE_CON18	0x0248	W	0x00000000	Internal clock gating register18

Name	Offset	Size	Reset Value	Description
CRU_CLKGATE_CON19	0x024c	W	0x00000000	Internal clock gating register19
CRU_CLKGATE_CON20	0x0250	W	0x00000000	Internal clock gating register20
CRU_CLKGATE_CON21	0x0254	W	0x00000000	Internal clock gating register21
CRU_CLKGATE_CON22	0x0258	W	0x00000000	Internal clock gating register22
CRU_CLKGATE_CON23	0x025c	W	0x00000000	Internal clock gating register23
CRU_CLKGATE_CON24	0x0260	W	0x00000000	Internal clock gating register24
CRU_CLKGATE_CON25	0x0264	W	0x00000000	Internal clock gating register25
CRU_CLKGATE_CON26	0x0268	W	0x00000000	Internal clock gating register26
CRU_CLKGATE_CON27	0x026c	W	0x00000000	Internal clock gating register27
CRU_CLKGATE_CON28	0x0270	W	0x00000000	Internal clock gating register28
CRU_SSGTBL0_3	0x0280	W	0x00000000	SSMOD external wave table register0
CRU_SSGTBL4_7	0x0284	W	0x00000000	SSMOD external wave table register1
CRU_SSGTBL8_11	0x0288	W	0x00000000	SSMOD external wave table register2
CRU_SSGTBL12_15	0x028c	W	0x00000000	SSMOD external wave table register3
CRU_SSGTBL16_19	0x0290	W	0x00000000	SSMOD external wave table register4
CRU_SSGTBL20_23	0x0294	W	0x00000000	SSMOD external wave table register5
CRU_SSGTBL24_27	0x0298	W	0x00000000	SSMOD external wave table register6
CRU_SSGTBL28_31	0x029c	W	0x00000000	SSMOD external wave table register7
CRU_SSGTBL32_35	0x02a0	W	0x00000000	SSMOD external wave table register8
CRU_SSGTBL36_39	0x02a4	W	0x00000000	SSMOD external wave table register9
CRU_SSGTBL40_43	0x02a8	W	0x00000000	SSMOD external wave table register10
CRU_SSGTBL44_47	0x02ac	W	0x00000000	SSMOD external wave table register11
CRU_SSGTBL48_51	0x02b0	W	0x00000000	SSMOD external wave table register12
CRU_SSGTBL52_55	0x02b4	W	0x00000000	SSMOD external wave table register13
CRU_SSGTBL56_59	0x02b8	W	0x00000000	SSMOD external wave table register14
CRU_SSGTBL60_63	0x02bc	W	0x00000000	SSMOD external wave table register15
CRU_SSGTBL64_67	0x02c0	W	0x00000000	SSMOD external wave table register16
CRU_SSGTBL68_71	0x02c4	W	0x00000000	SSMOD external wave table register17
CRU_SSGTBL72_75	0x02c8	W	0x00000000	SSMOD external wave table register18

Name	Offset	Size	Reset Value	Description
CRU_SSGTBL76_79	0x02cc	W	0x00000000	SSMOD external wave table register19
CRU_SSGTBL80_83	0x02d0	W	0x00000000	SSMOD external wave table register20
CRU_SSGTBL84_87	0x02d4	W	0x00000000	SSMOD external wave table register21
CRU_SSGTBL88_91	0x02d8	W	0x00000000	SSMOD external wave table register22
CRU_SSGTBL92_95	0x02dc	W	0x00000000	SSMOD external wave table register23
CRU_SSGTBL96_99	0x02e0	W	0x00000000	SSMOD external wave table register24
CRU_SSGTBL100_103	0x02e4	W	0x00000000	SSMOD external wave table register25
CRU_SSGTBL104_107	0x02e8	W	0x00000000	SSMOD external wave table register26
CRU_SSGTBL108_111	0x02ec	W	0x00000000	SSMOD external wave table register27
CRU_SSGTBL112_115	0x02f0	W	0x00000000	SSMOD external wave table register28
CRU_SSGTBL116_119	0x02f4	W	0x00000000	SSMOD external wave table register29
CRU_SSGTBL120_123	0x02f8	W	0x00000000	SSMOD external wave table register30
CRU_SSGTBL124_127	0x02fc	W	0x00000000	SSMOD external wave table register31
CRU_SOFTRST_CON0	0x0300	W	0x00000000	Internal software reset control register0
CRU_SOFTRST_CON1	0x0304	W	0x00000000	Internal software reset control register1
CRU_SOFTRST_CON2	0x0308	W	0x00000000	Internal software reset control register2
CRU_SOFTRST_CON3	0x030c	W	0x00000000	Internal software reset control register3
CRU_SOFTRST_CON4	0x0310	W	0x00000000	Internal software reset control register4
CRU_SOFTRST_CON5	0x0314	W	0x00000000	Internal software reset control register5
CRU_SOFTRST_CON6	0x0318	W	0x00000000	Internal software reset control register6
CRU_SOFTRST_CON7	0x031c	W	0x00000000	Internal software reset control register7
CRU_SOFTRST_CON8	0x0320	W	0x00000000	Internal software reset control register8
CRU_SOFTRST_CON9	0x0324	W	0x00000000	Internal software reset control register9
CRU_SOFTRST_CON10	0x0328	W	0x00000000	Internal software reset control register10
CRU_SOFTRST_CON11	0x032c	W	0x00000000	Internal software reset control register11
CRU_CRU_SDMMC_CON0	0x0380	W	0x00000004	sdmmc control0
CRU_CRU_SDMMC_CON1	0x0384	W	0x00000000	sdmmc control1
CRU_CRU_SDIO_CON0	0x0388	W	0x00000004	SDIO control0
CRU_CRU_SDIO_CON1	0x038c	W	0x00000000	SDIO control1

Name	Offset	Size	Reset Value	Description
CRU_CRU_EMMC_CON0	0x0390	W	0x00000004	EMMC control0
CRU_CRU_EMMC_CON1	0x0394	W	0x00000000	EMMC control1
CRU_CRU_SDMMC_EXT_CON0	0x0398	W	0x00000004	SDMMC_EXT control0
CRU_CRU_SDMMC_EXT_CON1	0x039c	W	0x00000000	SDMMC_EXT control1

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

2.6.3 Detail Register Description

CRU_APLL_CON0

Address: Operational Base + offset (0x0000)

APLL configuration register0

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

CRU_APLL_CON1

Address: Operational Base + offset (0x0004)

APLL configuration register1

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

Bit	Attr	Reset Value	Description
15	RW	0x0	pllpdsel PLL global power down source selection "If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b2: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable " 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_APLL_CON2

Address: Operational Base + offset (0x0008)

APLL configuration register2

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

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

CRU_APLL_CON3

Address: Operational Base + offset (0x000c)

APLL configuration register3

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

CRU_APLL_CON4
Address: Operational Base + offset (0x0010)
APLL configuration register4

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

CRU_DPLL_CON0
Address: Operational Base + offset (0x0020)
DPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 First Post Divide Value (1-7)
11:0	RW	0x096	fbdiv Feedback Divide Value "Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_DPLL_CON1
Address: Operational Base + offset (0x0024)
DPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection "If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b2: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable " 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_DPLL_CON2

Address: Operational Base + offset (0x0028)

DPLL configuration register2

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	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_DPLL_CON3

Address: Operational Base + offset (0x002c)

DPLL configuration register3

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

Bit	Attr	Reset Value	Description
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_DPLL_CON4

Address: Operational Base + offset (0x0030)

DPLL configuration register4

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

CRU_CPLL_CON0

Address: Operational Base + offset (0x0040)

CPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x2	postdiv1 First Post Divide Value (1-7)
11:0	RW	0x0c8	fbdiv Feedback Divide Value "Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_CPLL_CON1
Address: Operational Base + offset (0x0044)
CPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection "If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b2: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable " 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x03	refdiv Reference Clock Divide Value (1-63)

CRU_CPLL_CON2
Address: Operational Base + offset (0x0048)
CPLL configuration register2

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

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

CRU_CPLL_CON3
Address: Operational Base + offset (0x004c)
CPLL configuration register3

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

Bit	Attr	Reset Value	Description
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_CPLL_CON4

Address: Operational Base + offset (0x0050)

CPLL configuration register4

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

CRU_GPLL_CON0

Address: Operational Base + offset (0x0060)

GPLL configuration register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass PLL Bypass. FREF bypasses PLL to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 First Post Divide Value (1-7)

Bit	Attr	Reset Value	Description
11:0	RW	0x051	fbdiv Feedback Divide Value "Valid divider settings are: [16, 3200] in integer mode [20, 320] in fractional mode Tips: no plus one operation

CRU_GPLL_CON1

Address: Operational Base + offset (0x0064)

GPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection "If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b2: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x0	dsmpd PLL delta sigma modulator enable " 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x02	refdiv Reference Clock Divide Value (1-63)

CRU_GPLL_CON2

Address: Operational Base + offset (0x0068)

GPLL configuration register2

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	0xeb84f8	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_GPLL_CON3

Address: Operational Base + offset (0x006c)

GPLL configuration register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread spread amplitude % = 0.1 * SPREAD[4:0]
7:4	WO	0x0	ssmod_divval Divider required to set the modulation frequency Divider required to set the modulation frequency
3	WO	0x0	ssmod_downspread Selects center spread or downs pread 1'b0: down spread 1'b1: center spread

Bit	Attr	Reset Value	Description
2	WO	0x1	ssmod_reset Reset modulator state 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Bypass SSMOD by module 1'b0: no bypass 1'b1: bypass
0	WO	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_GPLL_CON4
Address: Operational Base + offset (0x0070)
GPLL configuration register4

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

CRU_CRU_MODE
Address: Operational Base + offset (0x0080)
CRU_MODE

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	gpll_work_mode PLL work mode select 1'b0: Slow mode, clock from external 24MHz/26MHz OSC (default) 1'b1: Normal mode, clock from PLL output
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	cpll_work_mode PLL work mode select 1'b0: Slow mode, clock from external 24MHz/26MHz OSC (default) 1'b1: Normal mode, clock from PLL output
7:5	RO	0x0	reserved
4	RW	0x0	dpll_work_mode PLL work mode select 1'b0: Slow mode, clock from external 24MHz/26MHz OSC (default) 1'b1: Normal mode, clock from PLL output
3:2	RO	0x0	reserved
1	RW	0x0	npll_work_mode PLL work mode select 1'b0: Slow mode, clock from external 24MHz/26MHz OSC (default) 1'b1: Normal mode, clock from PLL output
0	RW	0x0	apll_work_mode PLL work mode select 1'b0: Slow mode, clock from external 24MHz/26MHz OSC (default) 1'b1: Normal mode, clock from PLL output

CRU_CRU_MISC

Address: Operational Base + offset (0x0084)

CRU_MISC

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	usb480m_24m_sel USB PHY select 1'b1: when using USB480M as clock source, clock source freq is set to 24Mhz. 1'b0: when using USB480M as clock source, clock source freq is set to USBPHY480M output.
14	RO	0x0	reserved
13	RW	0x1	hdmiphy_24m_sel HDMI PHY select 1'b1: when using HDMIPHY as clock source, clock source freq is set to 24Mhz. 1'b0: when using HDMIPHY as clock source, clock source freq is set to HDMIPHY pixel output.

Bit	Attr	Reset Value	Description
12:8	RW	0x00	testclk_sel Test clock out select 5'd00: clk_wifi 5'd01: clk_hdmi_cec 5'd02: clk_core 5'd03: clk_ddrphy 5'd04: aclk_rkvdec 5'd05: aclk_rkvenc 5'd06: aclk_vpu 5'd07: aclk_rga 5'd08: aclk_vio 5'd09: aclk_vop 5'd10: aclk_gpu 5'd11: aclk_bus 5'd12: aclk_peri 5'd13: aclk_gmac 5'd14: dclk_vop 5'd15: clk_pdm 5'd16: clk_rga 5'd17: clk_vdec_core 5'd18: clk_venc_core 5'd19: clk_tsp 5'd20: clk_ddrphy1x 5'd21: usb3otg_pipe3_pclk 5'd22: otp_ips_osc_out 5'd23: clk_24m default: buf_clk_wifi
7:3	RO	0x0	reserved
2	RW	0x0	core_wrst_wfien CPU warm reset by wfi enable 1'b1: cpu warm reset is valid when only when wfi is asserted. 1'b0: cpu warm reset is not
1	RW	0x0	core_srst_wfien CPU wfi reset enable 1'b1: cpu reseted when wfi and softrst0[4] are both asserted. 1'b0: cpu reseted only by softrst0[4]
0	RW	0x0	warmrst_en CPU warm reset enable 1'b1: enable cpu warm reset. 1'b0: disable cpu warm reset.

CRU_GLB_CNT_TH

Address: Operational Base + offset (0x0090)

CRU_GLB_CNT_TH

Bit	Attr	Reset Value	Description
31:16	RW	0x3a98	pll_lockperiod Measured in OSC clock cycles.
15	RW	0x0	wdt_glb_srst_ctrl watch_dog trigger global soft reset select 1'b0: watch_dog trigger second global reset 1'b1: watch_dog trigger first global reset
14	RW	0x0	tsadc_glb_srst_ctrl TSADC trigger global soft reset select 1'b0: tsadc trigger second global reset 1'b1: tsadc trigger first global reset
31:0	RW	0x064	global_reset_counter_threshold Global soft reset counter threshold Global soft reset counter threshold

CRU_GLB_RST_ST

Address: Operational Base + offset (0x0094)

GLB_RST_ST

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	W1C	0x0	snd_glb_tsadc_rst_st sencond global TSADC triggered reset flag 1'b0: last hot reset is not sencond global TSADC triggered reset 1'b1: last hot reset is sencond global TSADC triggered reset
4	W1C	0x0	fst_glb_tsadc_rst_st first global TSADC triggered reset flag 1'b0: last hot reset is not first global TSADC triggered reset 1'b1: last hot reset is first global TSADC triggered reset
3	W1C	0x0	snd_glb_wdt_rst_st sencond global WDT triggered reset flag 1'b0: last hot reset is not sencond global WDT triggered reset 1'b1: last hot reset is sencond global WDT triggered reset
2	W1C	0x0	fst_glb_wdt_rst_st first global WDT triggered reset flag 1'b0: last hot reset is not first global WDT triggered reset 1'b1: last hot reset is first global WDT triggered reset
1	W1C	0x0	snd_glb_rst_st second global rst flag 1'b0: last hot reset is not sencond global reset 1'b1: last hot reset is sencond global reset
0	W1C	0x0	fst_glb_rst_st first global rst flag 1'b0: last hot reset is not first global reset 1'b1: last hot reset is first global reset

CRU_GLB_SRST_SND_VALUE

Address: Operational Base + offset (0x0098)

GLB_SRST_SND_VALUE

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

CRU_GLB_SRST_FST_VALUE

Address: Operational Base + offset (0x009c)

GLB_SRST_FST_VALUE

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

CRU_NPLL_CON0

Address: Operational Base + offset (0x00a0)

NPLL configuration register0

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

CRU_NPLL_CON1

Address: Operational Base + offset (0x00a4)

NPLL configuration register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel PLL global power down source selection "If pllpdsel == 1, PLL can be power down only by pllpd1, otherwise pll is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted.
14	RW	0x0	pllpd1 PLL global power down request 1'b0: no power down 1'b2: power down
13	RW	0x0	pllpd0 PLL global power down request 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd PLL delta sigma modulator enable " 1'b0: modulator is enable, 1'b1: modulator is disabled
11	RO	0x0	reserved
10	RO	0x0	pll_lock PLL lock status 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 Second Post Divide Value (1-7)
5:0	RW	0x01	refdiv Reference Clock Divide Value (1-63)

CRU_NPLL_CON2

Address: Operational Base + offset (0x00a8)

NPLL configuration register2

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	0x000001	fracdiv Fractional part of feedback divide (fraction = FRAC/2^24)

CRU_NPLL_CON3
Address: Operational Base + offset (0x00ac)
NPLL configuration register3

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

Bit	Attr	Reset Value	Description
0	RW	0x1	ssmod_bp Bypass SSMOD by integration 1'b0: no bypass 1'b1: bypass

CRU_NPLL_CON4

Address: Operational Base + offset (0x00b0)

NPLL configuration register4

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

CRU_CLKSEL_CON0

Address: Operational Base + offset (0x0100)

Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:13	RW	0x0	bus_aclk_pll_sel bus_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMIPHY 2'b11:reserved
12:8	RW	0x03	bus_aclk_div_con bus_aclk integer divider control register clk=clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	core_clk_pll_sel core_clk pll source selection register 2'b00:APLL 2'b01:GPLL 2'b10:DPLL 2'b11:NPLL
5	RO	0x0	reserved
4:0	RW	0x00	clk_core_div_con Core A53 clock divider frequency clk=clk_src/(div_con+1)

CRU_CLKSEL_CON1

Address: Operational Base + offset (0x0104)

Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	bus_pclk_div_con bus_pclk integer divider control register clk=clk_src/(div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x1	bus_hclk_div_con bus_hclk integer divider control register clk=clk_src/(div_con+1)
7	RO	0x0	reserved
6:4	RW	0x1	aclk_core_div_con aclk_core integer divider control register clk=clk_src/(div_con+1)
3:0	RW	0x3	clk_core_dbg_div_con clk_core_dbg integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON2

Address: Operational Base + offset (0x0108)

Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x00	func_24m_div_con func_24m integer divider control register clk=clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x03	test_div_con test integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON3

Address: Operational Base + offset (0x010c)

Internal clock select and divide register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	ddr_clk_pll_sel ddr_clk pll source selection register 2'b00:DPLL 2'b01:APLL 2'b10:CPLL 2'b11:reserved
7:3	RO	0x0	reserved
2:0	RW	0x0	ddr_div_cnt ddrphy reference clock divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON4

Address: Operational Base + offset (0x0110)

Internal clock select and divide register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:13	RW	0x0	ddrpdclk_clk_pll_sel pd_ddr pclk source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMIPHY 2'b11:reserved

Bit	Attr	Reset Value	Description
12:8	RW	0x07	pd_ddr_div_con pd_ddr pclk divider control register clk=clk_src/(div_con+1)
7:6	RW	0x2	otp_pll_sel otp pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:reserved
5:0	RW	0x00	otp_div_con otp integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON5

Address: Operational Base + offset (0x0114)

Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	efuse_pll_sel efuse pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC
13	RO	0x0	reserved
12:8	RW	0x00	efuse_div_con efuse integer divider control register clk=clk_src/(div_con+1)
7:0	RO	0x0	reserved

CRU_CLKSEL_CON6

Address: Operational Base + offset (0x0118)

Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2s0_pll_sel i2s0 pll source selection register 1'b0:CPLL 1'b1:GPLL
14:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	i2s0_clk_sel i2s0 clk source selection register 2'b00: divout 2'b01: frac_divout 2'b10: 12M clkin 2'b11: 12M clkin
7	RO	0x0	reserved
6:0	RW	0x0f	i2s0_pll_div_con i2s0 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON7

Address: Operational Base + offset (0x011c)

Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s0_frac_div_con i2s0 fraction divider control register High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON8

Address: Operational Base + offset (0x0120)

Internal clock select and divide register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2s1_pll_sel i2s1 pll source selection register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12	RW	0x0	i2s1_out_sel i2s1 output clock selection register 1'b0: clk_i2s1 1'b1: 12M
11:10	RO	0x0	reserved
9:8	RW	0x0	i2s1_clk_sel i2s1 clk source selection register 2'b00: divout 2'b01: frac_divout 2'b10: IO I2S1 clkin 2'b11: 12M clkin
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x0f	i2s1_pll_div_con i2s1 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON9

Address: Operational Base + offset (0x0124)

Internal clock select and divide register9

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s1_frac_div_con i2s1 fraction divider control register High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON10

Address: Operational Base + offset (0x0128)

Internal clock select and divide register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2s2_pll_sel i2s2 pll source selection register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12	RW	0x0	i2s2_out_sel i2s2 output clock selection register 1'b0: clk_i2s2 1'b1: 12M
11:10	RO	0x0	reserved
9:8	RW	0x0	i2s2_clk_sel i2s2 clk source selection register 2'b00: divout 2'b01: frac_divout 2'b10: IO I2S2 clkin 2'b11: 12M clkin
7	RO	0x0	reserved
6:0	RW	0x0f	i2s2_pll_div_con i2s2 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON11

Address: Operational Base + offset (0x012c)

Internal clock select and divide register11

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s2_frac_div_con i2s2 fraction divider control register High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL_CON12
Address: Operational Base + offset (0x0130)
Internal clock select and divide register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	spdif_pll_sel spdif pll source selection register 1'b0:CPLL 1'b1:GPLL
14:10	RO	0x0	reserved
9:8	RW	0x0	spdif_clk_sel spdif clock source selection register 2'b00: divout 2'b01: frac_divout 2'b10: 12M clkin 2'b11: 12M clkin
7	RO	0x0	reserved
6:0	RW	0x0f	spdif_pll_div_con spdif pll divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON13
Address: Operational Base + offset (0x0134)
Internal clock select and divide register13

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	spdif_frac_div_con spdif fraction divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON14
Address: Operational Base + offset (0x0138)
Internal clock select and divide register14

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

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13:12	RW	0x0	uart0_pll_sel clk_uart0 pll source select control register 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select USBPHY 480M clock
11:10	RO	0x0	reserved
9:8	RW	0x0	uart0_clk_sel clk_uart0 clock source select control register 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc input 2'b11: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x07	uart0_pll_div_con clk_uart0 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON15

Address: Operational Base + offset (0x013c)

Internal clock select and divide register15

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart0_frac_div_con Control UART0 fraction divider frequency. High 16-bit for numerator Low 16-bit for denominator "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

CRU_CLKSEL_CON16

Address: Operational Base + offset (0x0140)

Internal clock select and divide register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:12	RW	0x0	uart1_pll_sel clk_uart1 pll source select control register 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select USBPHY 480M clock

Bit	Attr	Reset Value	Description
11:10	RO	0x0	reserved
9:8	RW	0x0	uart1_clk_sel clk_uart1 clock source select control register 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc input 2'b11: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x07	uart1_pll_div_con clk_uart1 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON17

Address: Operational Base + offset (0x0144)

Internal clock select and divide register17

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart1_frac_div_con Control uart1 fraction divider frequency. High 16-bit for numerator Low 16-bit for denominator "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

CRU_CLKSEL_CON18

Address: Operational Base + offset (0x0148)

Internal clock select and divide register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:12	RW	0x0	uart2_pll_sel clk_uart2 pll source select control register 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select USBPHY 480M clock
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	uart2_clk_sel clk_uart2 clock source select control register 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc input 2'b11: select 24MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x07	uart2_pll_div_con clk_uart2 divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON19

Address: Operational Base + offset (0x014c)

Internal clock select and divide register19

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart2_frac_div_con Control uart2 fraction divider frequency. High 16-bit for numerator Low 16-bit for denominator "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

CRU_CLKSEL_CON20

Address: Operational Base + offset (0x0150)

Internal clock select and divide register20

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	pdm_pll_sel pdm pll source selection register 2'd0: CPLL 2'd1: GPLL 2'd2: APLL 2'd3: Reserved
13	RO	0x0	reserved
12:8	RW	0x0f	pdm_div_con pdm integer divider control register clk=clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7	RW	0x0	crypto_pll_sel crypto pll source selection register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x04	crypto_div_con crypto integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON21
Address: Operational Base + offset (0x0154)
Internal clock select and divide register21

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	tsp_pll_sel tsp pll source selection register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x04	tsp_div_con tsp integer divider control register clk=clk_src/(div_con+1)
7:0	RO	0x0	reserved

CRU_CLKSEL_CON22
Address: Operational Base + offset (0x0158)
Internal clock select and divide register22

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:0	RW	0x1e0	tsadc_div_con tsadc integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON23
Address: Operational Base + offset (0x015c)

Internal clock select and divide register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:0	RW	0x1e0	saradc_div_con saradc integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON24

Address: Operational Base + offset (0x0160)

Internal clock select and divide register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clkpwm_pll_sel clkpwm pll source selection register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	pwm0_div_con pwm0 integer divider control register clk=clk_src/(div_con+1)
7	RW	0x0	clkspi_pll_sel clkspi pll source selection register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x07	spi0_div_con spi0 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON25

Address: Operational Base + offset (0x0164)

Internal clock select and divide register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x2	gmac_pclk_div_con gmac_pclk integer divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	gmac_aclk_pll_sel gmac_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMIPHY 2'b11:reserved
5	RO	0x0	reserved
4:0	RW	0x02	gmac_aclk_div_con gmac_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON26

Address: Operational Base + offset (0x0168)

Internal clock select and divide register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	clk_gmac2phy_div_con clk_gmac2phy integer divider control register clk=clk_src/(div_con+1)
7	RW	0x0	gmac2phy_pll_sel gmac2phy pll source selection register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x0f	gmac2phy_div_con gmac2phy integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON27

Address: Operational Base + offset (0x016c)

Internal clock select and divide register27

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

Bit	Attr	Reset Value	Description
15	RW	0x0	gmac2io_out_pll_sel gmac2io_out pll source selection register 1'b0:CPLL 1'b1:GPLL
14:13	RO	0x0	reserved
12:8	RW	0x07	gmac2io_out_div_con gmac2io_out integer divider control register clk=clk_src/(div_con+1)
7	RW	0x0	gmac2io_pll_sel gmac2io pll source selection register 1'b0:CPLL 1'b1:GPLL
6:5	RO	0x0	reserved
4:0	RW	0x05	gmac2io_div_con gmac2io integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON28
Address: Operational Base + offset (0x0170)
Internal clock select and divide register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	periph_pll_sel periph pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMIPHY 2'b11:reserved
5	RO	0x0	reserved
4:0	RW	0x02	periph_aclk_div_con periph_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON29
Address: Operational Base + offset (0x0174)
Internal clock select and divide register29

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:7	RO	0x0	reserved
6:4	RW	0x2	periph_pclk_div_con periph_pclk integer divider control register clk=clk_src/(div_con+1)
3:2	RO	0x0	reserved
1:0	RW	0x2	periph_hclk_div_con periph_hclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON30
Address: Operational Base + offset (0x0178)
Internal clock select and divide register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	clksdmmc_pll_sel clksdmmc pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:USBPHY 480M
7:0	RW	0x03	sdmmc0_div_con sdmmc0 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON31
Address: Operational Base + offset (0x017c)
Internal clock select and divide register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	clksdio_pll_sel clksdio pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:USBPHY 480M
7:0	RW	0x01	sdio_div_con sdio integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON32

Address: Operational Base + offset (0x0180)

Internal clock select and divide register32

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x0	clkemmc_pll_sel clkemmc pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:USBPHY 480M
7:0	RW	0x01	emmc_div_con emmc integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0184)

Internal clock select and divide register33

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb3_otg0_suspend_src_sel clk_usb3_otg0_suspend pll source selection register 1'b0: OSC input 1'b1: 32k clock
14:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:0	RW	0x30b	clk_usb3_otg0_suspend_div_con clk_usb3_otg0_suspend integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON34

Address: Operational Base + offset (0x0188)

Internal clock select and divide register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2c1_pll_sel i2c1 pll source selection register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	i2c1_div_con i2c1 integer divider control register clk=clk_src/(div_con+1)
7	RW	0x0	i2c0_pll_sel i2c0 pll source selection register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x07	i2c0_div_con i2c0 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON35

Address: Operational Base + offset (0x018c)

Internal clock select and divide register35

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2c3_pll_sel i2c3 pll source selection register 1'b0:CPLL 1'b1:GPLL
14:8	RW	0x07	i2c3_div_con i2c3 integer divider control register clk=clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7	RW	0x0	i2c2_pll_sel i2c2 pll source selection register 1'b0:CPLL 1'b1:GPLL
6:0	RW	0x07	i2c2_div_con i2c2 integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON36

Address: Operational Base + offset (0x0190)

Internal clock select and divide register36

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x1	rga_aclk_pll_sel rga_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
13	RO	0x0	reserved
12:8	RW	0x02	rga_aclk_div_con rga_aclk integer divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	rga_clk_pll_sel rga_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x02	rga_clk_div_con rga_clk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON37

Address: Operational Base + offset (0x0194)

Internal clock select and divide register37

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x02	hclk_vio_div_con hclk_vio integer divider control register clk=clk_src/(div_con+1)
7:6	RW	0x1	vio_aclk_pll_sel vio_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x02	vio_aclk_div_con vio_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON38

Address: Operational Base + offset (0x0198)

Internal clock select and divide register38

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x3	rtc32k_clk_pll_sel rtc32k_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:Reserved
13:0	RW	0x02dc	rtc32k_clk_div_con rtc32k_clk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON39

Address: Operational Base + offset (0x019c)

Internal clock select and divide register39

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	vop_aclk_pll_sel vop_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x01	vop_aclk_div_con vop_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON40

Address: Operational Base + offset (0x01a0)

Internal clock select and divide register40

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x31	vop_dclk_div_con vop_dclk integer divider control register clk=clk_src/(div_con+1)
7:6	RO	0x0	reserved
5:3	RW	0x2	hdmiphy_div_con hdmiphy integer divider control register clk=clk_src/(div_con+1)
2	RW	0x0	vop_dclk_frac_sel vop divider source selection register 1'b0: divout 1'b1: frac_divout
1	RW	0x1	vop_dclk_src_sel vop dclk source selection register 1'b0:HDMIPHY 1'b2:PLL
0	RW	0x1	vop_dclk_pll_src_sel vop dclk pll source selection register 1'b0:GPLL 1'b1:CPLL

CRU_CLKSEL_CON41
Address: Operational Base + offset (0x01a4)
Internal clock select and divide register41

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	dclk_vop_frac_div_con dclk_vop fraction divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON42
Address: Operational Base + offset (0x01a8)
Internal clock select and divide register42

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	cif_pll_sel cif pll source selection register 1'b0:HDMIPLL 1'b1:GPLL
6	RO	0x0	reserved
5	RW	0x0	cif_clk_sel cif clk source selection register 1'b0:PLL 1'b1:OSC input
4:0	RW	0x13	cif_div_con cif integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON43
Address: Operational Base + offset (0x01ac)
Internal clock select and divide register43

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	clksdmmcext_pll_sel clksdmmcext pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:OSC input 2'b11:USBPHY 480M
7:0	RW	0x03	sdmmcext_div_con sdmmcext integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON44

Address: Operational Base + offset (0x01b0)

Internal clock select and divide register44

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	gpu_aclk_pll_sel gpu_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x02	gpu_aclk_div_con gpu_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON45

Address: Operational Base + offset (0x01b4)

Internal clock select and divide register45

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	clk_usb3phy_ref_sel usb3phy_ref clock source selection register 1'b0:OSC input 1'b1:PLL
7	RW	0x0	usb3phy_ref_pll_sel usb3phy_ref pll source selection register 1'b0:CPLL 1'b1:GPLL

Bit	Attr	Reset Value	Description
6:0	RW	0x1f	usb3phy_ref_div_con usb3phy_ref integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON46

Address: Operational Base + offset (0x01b8)

Internal clock select and divide register46

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	Reserve write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	RO	0x0	reserved

CRU_CLKSEL_CON48

Address: Operational Base + offset (0x01c0)

Internal clock select and divide register48

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x1	cabac_clk_pll_sel cabac_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
13	RO	0x0	reserved
12:8	RW	0x02	cabac_clk_div_con cabac_clk integer divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	rkvdec_aclk_pll_sel rkvdec_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x01	rkvdec_aclk_div_con rkvdec_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON49
Address: Operational Base + offset (0x01c4)
Internal clock select and divide register49

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	vdec_clk_pll_sel vdec_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x02	vdec_clk_div_con vdec_clk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON50
Address: Operational Base + offset (0x01c8)
Internal clock select and divide register50

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x1	vpu_aclk_pll_sel vpu_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x02	vpu_aclk_div_con vpu_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON51
Address: Operational Base + offset (0x01cc)
Internal clock select and divide register51

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	h265_core_clk_pll_sel h265_core_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
13	RO	0x0	reserved
12:8	RW	0x02	h265_core_clk_div_con h265_core_clk integer divider control register clk=clk_src/(div_con+1)
7:6	RW	0x0	rkvenc_aclk_pll_sel rkvenc_aclk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
5	RO	0x0	reserved
4:0	RW	0x03	rkvenc_aclk_div_con rkvenc_aclk integer divider control register clk=clk_src/(div_con+1)

CRU_CLKSEL_CON52
Address: Operational Base + offset (0x01d0)
Internal clock select and divide register52

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	h265_dsp_clk_pll_sel h265_dsp_clk pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b10:HDMI PHY 2'b11:USBPHY 480M
13	RO	0x0	reserved
12:8	RW	0x02	h265_dsp_clk_div_con h265_dsp_clk integer divider control register clk=clk_src/(div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	wifi_pll_sel wifi pll source selection register 2'b00:CPLL 2'b01:GPLL 2'b11:USBPHY 480M 2'b11:Reserved
5:0	RW	0x1e	wifi_div_con wifi integer divider control register clk=clk_src/(div_con+1)

CRU_CLKGATE_CON0

Address: Operational Base + offset (0x0200)

Internal clock gating register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	core_npll_clk_en core_npll clk gate enable register When HIGH, disable clock
11	RW	0x0	clk_rtc32k_src_en clk_rtc32k clk gate enable register "When HIGH, disable clock
10	RW	0x0	clk_wifi_src_en clk_wifi clk gate enable register "When HIGH, disable clock
9	RW	0x0	testclk_en tes clk gate enable register "When HIGH, disable clock
8:7	RO	0x0	reserved
6	RW	0x0	clk_ddrmon_en clk_ddrmon clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_ddrpd_src_en clk_ddrpd clk gate enable register When HIGH, disable clock
4	RW	0x0	clk_ddrphy_src_en clk_ddrphy clk gate enable register When HIGH, disable clock
3	RW	0x0	bus_src_clk_en bus_src clk gate enable register When HIGH, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	core_gpll_clk_en core_gpll clk gate enable register When HIGH, disable clock
1	RW	0x0	core_dpll_clk_en core_dpll clk gate enable register When HIGH, disable clock
0	RW	0x0	core_apll_clk_en core_apll clk gate enable register When HIGH, disable clock

CRU_CLKGATE_CON1
Address: Operational Base + offset (0x0204)
Internal clock gating register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart0_frac_src_en clk_uart0_frac clk gate enable register "When HIGH, disable clock
14	RW	0x0	clk_uart0_src_en clk_uart0 clk gate enable register "When HIGH, disable clock
13	RW	0x0	clk_spdif_frac_src_en clk_spdif_frac clk gate enable register "When HIGH, disable clock
12	RW	0x0	clk_spdif_src_en clk_spdif clk gate enable register "When HIGH, disable clock
11	RW	0x0	clk_i2s2_out_en clk_i2s2_out clk gate enable register "When HIGH, disable clock
10	RW	0x0	clk_i2s2_en clk_i2s2 clk gate enable register "When HIGH, disable clock
9	RW	0x0	clk_i2s2_frac_src_en clk_i2s2_frac clk gate enable register "When HIGH, disable clock
8	RW	0x0	clk_i2s2_src_en clk_i2s2 clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_i2s1_out_en clk_i2s1_out clk gate enable register "When HIGH, disable clock
6	RW	0x0	clk_i2s1_en clk_i2s1 clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_i2s1_frac_src_en clk_i2s1_frac clk gate enable register "When HIGH, disable clock
4	RW	0x0	clk_i2s1_src_en clk_i2s1 clk gate enable register "When HIGH, disable clock
3	RW	0x0	clk_i2s0_en clk_i2s0 clk gate enable register "When HIGH, disable clock
2	RW	0x0	clk_i2s0_frac_src_en clk_i2s0_frac clk gate enable register "When HIGH, disable clock
1	RW	0x0	clk_i2s0_src_en clk_i2s0 clk gate enable register "When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE_CON2

Address: Operational Base + offset (0x0208)

Internal clock gating register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_pdm_src_en clk_pdm clk gate enable register "When HIGH, disable clock
14	RW	0x0	clk_saradc_src_en clk_saradc clk gate enable register "When HIGH, disable clock
13	RW	0x0	clk_efuse_src_en clk_efuse clk gate enable register "When HIGH, disable clock
12	RW	0x0	clk_i2c3_src_en clk_i2c3 clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	clk_i2c2_src_en clk_i2c2 clk gate enable register "When HIGH, disable clock
10	RW	0x0	clk_i2c1_src_en clk_i2c1 clk gate enable register "When HIGH, disable clock
9	RW	0x0	clk_i2c0_src_en clk_i2c0 clk gate enable register "When HIGH, disable clock
8	RW	0x0	clk_pwm0_src_en clk_pwm0 clk gate enable register "When HIGH, disable clock
7	RW	0x0	clk_spi0_src_en clk_spi0 clk gate enable register "When HIGH, disable clock
6	RW	0x0	clk_tsadc_src_en clk_tsadc clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_tsp_src_en clk_tsp clk gate enable register "When HIGH, disable clock
4	RW	0x0	clk_crypto_src_en clk_crypto clk gate enable register "When HIGH, disable clock
3	RW	0x0	clk_uart2_frac_src_en clk_uart2_frac clk gate enable register "When HIGH, disable clock
2	RW	0x0	clk_uart2_src_en clk_uart2 clk gate enable register "When HIGH, disable clock
1	RW	0x0	clk_uart1_frac_src_en clk_uart1_frac clk gate enable register "When HIGH, disable clock
0	RW	0x0	clk_uart1_src_en clk_uart1 clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON3

Address: Operational Base + offset (0x020c)

Internal clock gating register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	clk_otp_src_en clk_otp clk gate enable register "When HIGH, disable clock
7:6	RO	0x0	reserved
5	RW	0x0	clk_gmac2io_out_en clk_gmac2io_out clk gate enable register "When HIGH, disable clock
4	RW	0x0	gmac_vpll_src_en gmac_vpll clk gate enable register "When HIGH, disable clock
3	RW	0x0	gmac_gpll_src_en gmac_gpll clk gate enable register "When HIGH, disable clock
2	RW	0x0	gmac_cpll_src_en gmac_cpll clk gate enable register "When HIGH, disable clock
1	RW	0x0	clk_gmac2io_src_en clk_gmac2io clk gate enable register "When HIGH, disable clock
0	RW	0x0	clk_gmac2phy_src_en clk_gmac2phy clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON4

Address: Operational Base + offset (0x0210)

Internal clock gating register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	clk_sdmmcext_src_en clk_sdmmcext clk gate enable register "When HIGH, disable clock
9	RW	0x0	clk_usb3phy_ref_25m_en clk_usb3phy_ref_25m clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	clk_usb3_otg0_suspend_en clk_usb3_otg0_suspend clk gate enable register "When HIGH, disable clock
7	RW	0x0	clk_usb3_otg0_ref_en clk_usb3_otg0_ref clk gate enable register "When HIGH, disable clock
6	RW	0x0	clk_otgphy0_en clk_otgphy0 clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_emmc_src_en clk_emmc clk gate enable register "When HIGH, disable clock
4	RW	0x0	clk_sdio_src_en clk_sdio clk gate enable register "When HIGH, disable clock
3	RW	0x0	clk_mmc0_src_en clk_mmc0 clk gate enable register "When HIGH, disable clock
2	RW	0x0	periph_vclk_src_en periph_vclk clk gate enable register "When HIGH, disable clock
1	RW	0x0	periph_cclk_src_en periph_cclk clk gate enable register "When HIGH, disable clock
0	RW	0x0	periph_gclk_src_en periph_gclk clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON5
Address: Operational Base + offset (0x0214)
Internal clock gating register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:7	RO	0x0	reserved
6	RW	0x0	dclk_vop_src_en dclk_vop clk gate enable register "When HIGH, disable clock
5	RW	0x0	aclk_vop_src_en aclk_vop clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	clk_hdmi_sfr_en clk_hdmi_sfr clk gate enable register "When HIGH, disable clock
3	RW	0x0	clk_cif_out_src_en clk_cif_out clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_vio_src_en aclk_vio clk gate enable register "When HIGH, disable clock
1	RW	0x0	clk_rga_src_en clk_rga clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_rga_src_en aclk_rga clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON6

Address: Operational Base + offset (0x0218)

Internal clock gating register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	clk_venc_dsp_src_en clk_venc_dsp clk gate enable register "When HIGH, disable clock
6	RW	0x0	aclk_gpu_src_en aclk_gpu clk gate enable register "When HIGH, disable clock
5	RW	0x0	aclk_vpu_src_en aclk_vpu clk gate enable register "When HIGH, disable clock
4	RW	0x0	clk_venc_core_src_en clk_venc_core clk gate enable register "When HIGH, disable clock
3	RW	0x0	aclk_rkvenc_src_en aclk_rkvenc clk gate enable register "When HIGH, disable clock
2	RW	0x0	clk_vdec_core_src_en clk_vdec_core clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	clk_cabac_src_en clk_cabac clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_rkvdec_src_en aclk_rkvdec clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON7

Address: Operational Base + offset (0x021c)

Internal clock gating register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:5	RO	0x0	reserved
4	RW	0x0	pclk_ddr_en pclk_ddr clk gate enable register "When HIGH, disable clock
3	RO	0x0	reserved
2	RW	0x0	clk_jtag_en core jtag clock enable "When HIGH, disable clock
1	RW	0x0	clk_core_periph_en clk_core_periph clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_core_en aclk_core clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON8

Address: Operational Base + offset (0x0220)

Internal clock gating register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	clk_timer5_en clk_timer5 clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_timer4_en clk_timer4 clk gate enable register "When HIGH, disable clock
8	RW	0x0	clk_timer3_en clk_timer3 clk gate enable register "When HIGH, disable clock
7	RW	0x0	clk_timer2_en clk_timer2 clk gate enable register "When HIGH, disable clock
6	RW	0x0	clk_timer1_en clk_timer1 clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_timer0_en clk_timer0 clk gate enable register "When HIGH, disable clock
4	RW	0x0	pclk_phy_en pclk_phy clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_bus_en pclk_bus clk gate enable register "When HIGH, disable clock
2	RW	0x0	pclk_bus_src_en pclk_bus clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_bus_en hclk_bus clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_bus_en aclk_bus clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON9

Address: Operational Base + offset (0x0224)

Internal clock gating register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	clk_gmac2io_ref_en clk_gmac2io_ref clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_gmac2io_refout_en clk_gmac2io_refout clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_gmac2io_tx_en clk_gmac2io_tx clk gate enable register "When HIGH, disable clock
4	RW	0x0	clk_gmac2io_rx_en clk_gmac2io_rx clk gate enable register "When HIGH, disable clock
3	RW	0x0	clk_gmac2phy_ref_en clk_gmac2phy_ref clk gate enable register "When HIGH, disable clock
2	RW	0x0	clk_macphy_en clk_macphy clk gate enable register "When HIGH, disable clock
1	RW	0x0	clk_gmac2phy_rx_en clk_gmac2phy_rx clk gate enable register "When HIGH, disable clock
0	RW	0x0	pclk_gmac_en pclk_gmac clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON10

Address: Operational Base + offset (0x0228)

Internal clock gating register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:3	RO	0x0	reserved
2	RW	0x0	pclk_periph_en pclk_periph clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_periph_en hclk_periph clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_periph_en aclk_periph clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON11

Address: Operational Base + offset (0x022c)

Internal clock gating register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	hclk_vpu_en hclk_vpu clk gate enable register "When HIGH, disable clock
7:5	RO	0x0	reserved
4	RW	0x0	hclk_rkvenc_en hclk_rkvenc clk gate enable register "When HIGH, disable clock
3:1	RO	0x0	reserved
0	RW	0x0	hclk_rkvdec_en hclk_rkvdec clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON12

Address: Operational Base + offset (0x0230)

Internal clock gating register12

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Reserve write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

CRU_CLKGATE_CON13

Address: Operational Base + offset (0x0234)

Internal clock gating register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:2	RO	0x0	reserved
1	RW	0x0	aclk_gic400_en aclk_gic400 clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_core_niu_en aclk_core_niu clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON14
Address: Operational Base + offset (0x0238)
Internal clock gating register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:2	RO	0x0	reserved
1	RW	0x0	aclk_gpu_niu_en aclk_gpu_niu clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_gpu_en aclk_gpu clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON15
Address: Operational Base + offset (0x023c)
Internal clock gating register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_phy_niu_en pclk_phy_niu clk gate enable register "When HIGH, disable clock
14	RW	0x0	pclk_bus_niu_en pclk_bus_niu clk gate enable register "When HIGH, disable clock
13	RW	0x0	hclk_bus_niu_en hclk_bus_niu clk gate enable register "When HIGH, disable clock
12	RW	0x0	aclk_bus_niu_en aclk_bus_niu clk gate enable register "When HIGH, disable clock
11	RW	0x0	aclk_dcf_en aclk_dcf clk gate enable register "When HIGH, disable clock
10	RW	0x0	pclk_i2c0_en pclk_i2c0 clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	pclk_efuse_1024_en pclk_efuse_1024 clk gate enable register "When HIGH, disable clock
8	RW	0x0	sclk_crypto_en sclk_crypto clk gate enable register "When HIGH, disable clock
7	RW	0x0	mclk_crypto_en mclk_crypto clk gate enable register "When HIGH, disable clock
6	RW	0x0	hclk_spdif_8ch_en hclk_spdif_8ch clk gate enable register "When HIGH, disable clock
5	RW	0x0	hclk_i2s2_2ch_en hclk_i2s2_2ch clk gate enable register "When HIGH, disable clock
4	RW	0x0	hclk_i2s1_8ch_en hclk_i2s1_8ch clk gate enable register "When HIGH, disable clock
3	RW	0x0	hclk_i2s0_8ch_en hclk_i2s0_8ch clk gate enable register "When HIGH, disable clock
2	RW	0x0	hclk_rom_en hclk_rom clk gate enable register "When HIGH, disable clock
1	RW	0x0	aclk_dmac_bus_en aclk_dmac_bus clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_intmem_en aclk_intmem clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON16

Address: Operational Base + offset (0x0240)

Internal clock gating register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_dcf_en pclk_dcf clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
14	RW	0x0	pclk_tsadc_en pclk_tsadc clk gate enable register "When HIGH, disable clock
13	RW	0x0	pclk_uart2_en pclk_uart2 clk gate enable register "When HIGH, disable clock
12	RW	0x0	pclk_uart1_en pclk_uart1 clk gate enable register "When HIGH, disable clock
11	RW	0x0	pclk_uart0_en pclk_uart0 clk gate enable register "When HIGH, disable clock
10	RW	0x0	pclk_gpio3_en pclk_gpio3 clk gate enable register "When HIGH, disable clock
9	RW	0x0	pclk_gpio2_en pclk_gpio2 clk gate enable register "When HIGH, disable clock
8	RW	0x0	pclk_gpio1_en pclk_gpio1 clk gate enable register "When HIGH, disable clock
7	RW	0x0	pclk_gpio0_en pclk_gpio0 clk gate enable register "When HIGH, disable clock
6	RW	0x0	pclk_rk_pwm_en pclk_rk_pwm clk gate enable register "When HIGH, disable clock
5	RW	0x0	pclk_spi0_en pclk_spi0 clk gate enable register "When HIGH, disable clock
4	RW	0x0	pclk_stimer_en pclk_stimer clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_timer0_en pclk_timer0 clk gate enable register "When HIGH, disable clock
2	RW	0x0	pclk_i2c3_en pclk_i2c3 clk gate enable register "When HIGH, disable clock
1	RW	0x0	pclk_i2c2_en pclk_i2c2 clk gate enable register "When HIGH, disable clock
0	RW	0x0	pclk_i2c1_en pclk_i2c1 clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON17
Address: Operational Base + offset (0x0244)
Internal clock gating register17

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_saradc_en pclk_saradc clk gate enable register "When HIGH, disable clock
14	RW	0x0	pclk_usb_grf_en pclk_usb_grf clk gate enable register "When HIGH, disable clock
13	RW	0x0	clk_hsadc_0_tsp_en clk_hsadc_0_tsp clk gate enable register "When HIGH, disable clock
12	RW	0x0	aclk_tsp_en aclk_tsp clk gate enable register "When HIGH, disable clock
11	RW	0x0	hclk_tsp_en hclk_tsp clk gate enable register "When HIGH, disable clock
10	RW	0x0	pclk_scr_en pclk_scr clk gate enable register "When HIGH, disable clock
9	RO	0x0	reserved
8	RW	0x0	pclk_vdacphy_en pclk_vdacphy clk gate enable register "When HIGH, disable clock
7	RW	0x0	pclk_hdmiphy_en pclk_hdmiphy clk gate enable register "When HIGH, disable clock
6	RW	0x0	pclk_sgrf_en pclk_sgrf clk gate enable register "When HIGH, disable clock
5	RW	0x0	pclk_acodecphy_en pclk_acodecphy clk gate enable register "When HIGH, disable clock
4	RW	0x0	pclk_cru_en pclk_cru clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
3	RW	0x0	pclk_ddrphy_en pclk_ddrphy clk gate enable register "When HIGH, disable clock
2	RW	0x0	pclk_usb3grf_en pclk_usb3grf clk gate enable register "When HIGH, disable clock
1	RO	0x0	reserved
0	RW	0x0	pclk_grf_en pclk_grf clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON18

Address: Operational Base + offset (0x0248)

Internal clock gating register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	pclk_ddrstdby_en pclk_ddrstdby clk gate enable register "When HIGH, disable clock
6	RW	0x0	clk_ddr_msch_en clk_ddr_msch clk gate enable register "When HIGH, disable clock
5	RW	0x0	clk_ddr_upctl_en clk_ddr_upctl clk gate enable register "When HIGH, disable clock
4	RW	0x0	aclk_ddr_upctl_en aclk_ddr_upctl clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_ddr_mon_en pclk_ddr_mon clk gate enable register "When HIGH, disable clock
2	RW	0x0	pclk_ddr_msch_en pclk_ddr_msch clk gate enable register "When HIGH, disable clock
1	RW	0x0	pclk_ddr_upctl_en pclk_ddr_upctl clk gate enable register "When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE_CON19

Address: Operational Base + offset (0x024c)

Internal clock gating register19

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_sdmmc_ext_en hclk_sdmmc_ext hclk gate enable register "When HIGH, disable clock
14	RW	0x0	aclk_usb3otg_en aclk_usb3otg clk gate enable register "When HIGH, disable clock
13	RW	0x0	pclk_peri_niu_en pclk_peri_niu clk gate enable register "When HIGH, disable clock
12	RW	0x0	hclk_peri_niu_en hclk_peri_niu clk gate enable register "When HIGH, disable clock
11	RW	0x0	aclk_peri_niu_en aclk_peri_niu clk gate enable register "When HIGH, disable clock
10	RO	0x0	reserved
9	RW	0x0	hclk_otg_pmu_en hclk_otg_pmu clk gate enable register "When HIGH, disable clock
8	RW	0x0	hclk_otg_en hclk_otg clk gate enable register "When HIGH, disable clock
7	RW	0x0	hclk_host0_arb_en hclk_host0_arb clk gate enable register "When HIGH, disable clock
6	RW	0x0	hclk_host0_en hclk_host0 clk gate enable register "When HIGH, disable clock
5:3	RO	0x0	reserved
2	RW	0x0	hclk_emmc_en hclk_emmc clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_sdio_en hclk_sdio clk gate enable register "When HIGH, disable clock
0	RW	0x0	hclk_sdmmc_en hclk_sdmmc clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON20
Address: Operational Base + offset (0x0250)
Internal clock gating register20

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Reserve write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

CRU_CLKGATE_CON21
Address: Operational Base + offset (0x0254)
Internal clock gating register21

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	aclk_hdcp_en aclk_hdcp clk gate enable register "When HIGH, disable clock
14	RW	0x0	hclk_h2p_en hclk_h2p clk gate enable register "When HIGH, disable clock
13	RW	0x0	pclk_h2p_en pclk_h2p clk gate enable register "When HIGH, disable clock
12	RW	0x0	hclk_ahb1tom_en hclk_ahb1tom clk gate enable register "When HIGH, disable clock
11	RW	0x0	hclk_rga_en hclk_rga clk gate enable register "When HIGH, disable clock
10	RW	0x0	aclk_rga_en aclk_rga clk gate enable register "When HIGH, disable clock
9	RW	0x0	hclk_cif_en hclk_cif clk gate enable register "When HIGH, disable clock
8	RW	0x0	aclk_cif_en aclk_cif clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
7	RW	0x0	hclk_iep_en hclk_iep clk gate enable register "When HIGH, disable clock
6	RW	0x0	aclk_iep_en aclk_iep clk gate enable register "When HIGH, disable clock
5	RW	0x0	hclk_vop_niu_en hclk_vop_niu clk gate enable register "When HIGH, disable clock
4	RW	0x0	aclk_vop_niu_en aclk_vop_niu clk gate enable register "When HIGH, disable clock
3	RW	0x0	hclk_vop_en hclk_vop clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_vop_en aclk_vop clk gate enable register "When HIGH, disable clock
1:0	RO	0x0	reserved

CRU_CLKGATE_CON22

Address: Operational Base + offset (0x0258)

Internal clock gating register22

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	pclk_hdcp_ctrl_en pclk_hdcp_ctrl clk gate enable register "When HIGH, disable clock
4	RW	0x0	pclk_hdmi_ctrl_en pclk_hdmi_ctrl clk gate enable register "When HIGH, disable clock
3	RW	0x0	aclk_rga_niu_en aclk_rga_niu clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_vio_niu_en aclk_vio_niu clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_vio_niu_en hclk_vio_niu clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	hclk_hdcp_en hclk_hdcp clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON23

Address: Operational Base + offset (0x025c)

Internal clock gating register23

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:4	RO	0x0	reserved
3	RW	0x0	hclk_vpu_niu_en hclk_vpu_niu clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_vpu_niu_en aclk_vpu_niu clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_vpu_en hclk_vpu clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_vpu_en aclk_vpu clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON24

Address: Operational Base + offset (0x0260)

Internal clock gating register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:4	RO	0x0	reserved
3	RW	0x0	hclk_rkvdec_niu_en hclk_rkvdec_niu clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_rkvdec_niu_en aclk_rkvdec_niu clk gate enable register "When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	hclk_rkvdec_en hclk_rkvdec clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_rkvdec_en aclk_rkvdec clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON25

Address: Operational Base + offset (0x0264)

Internal clock gating register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:7	RO	0x0	reserved
6	RW	0x0	aclk_axi2sram_en axi2sram clk gate enable register "When HIGH, disable clock
5	RW	0x0	hclk_h264_en hclk_h264 clk gate enable register "When HIGH, disable clock
4	RW	0x0	aclk_h264_en aclk_h264 clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_h265_en pclk_h265 clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_h265_en aclk_h265 clk gate enable register "When HIGH, disable clock
1	RW	0x0	hclk_rkvenc_niu_en hclk_rkvenc_niu clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_rkvenc_niu_en aclk_rkvenc_niu clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON26

Address: Operational Base + offset (0x0268)

Internal clock gating register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	pclk_gmac_niu_en pclk_gmac_niu clk gate enable register "When HIGH, disable clock
4	RW	0x0	aclk_gmac_niu_en aclk_gmac_niu clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_gmac2io_en pclk_gmac2io clk gate enable register "When HIGH, disable clock
2	RW	0x0	aclk_gmac2io_en aclk_gmac2io clk gate enable register "When HIGH, disable clock
1	RW	0x0	pclk_gmac2phy_en pclk_gmac2phy clk gate enable register "When HIGH, disable clock
0	RW	0x0	aclk_gmac2phy_en aclk_gmac2phy clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON27

Address: Operational Base + offset (0x026c)

Internal clock gating register27

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:2	RO	0x0	reserved
1	RW	0x0	clk4x_ddrphy_en clk4x_ddrphy clk gate enable register "When HIGH, disable clock
0	RW	0x0	clk_ddrphy_en clk_ddrphy clk gate enable register "When HIGH, disable clock

CRU_CLKGATE_CON28

Address: Operational Base + offset (0x0270)

Internal clock gating register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:5	RO	0x0	reserved
4	RW	0x0	pclk_otp_en pclk_otp clk gate enable register "When HIGH, disable clock
3	RW	0x0	pclk_pmu_en pclk_pmu clk gate enable register "When HIGH, disable clock
2	RW	0x0	pclk_usb3phy_pipe_en pclk_usb3phy_pipe clk gate enable register "When HIGH, disable clock
1	RW	0x0	pclk_usb3phy_otg_en pclk_usb3phy_otg clk gate enable register "When HIGH, disable clock
0	RW	0x0	hclk_pdm_en hclk_pdm clk gate enable register "When HIGH, disable clock

CRU_SSGTBL0_3

Address: Operational Base + offset (0x0280)

SSMOD external wave table register0

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

CRU_SSGTBL4_7

Address: Operational Base + offset (0x0284)

SSMOD external wave table register1

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

CRU_SSGTBL8_11
Address: Operational Base + offset (0x0288)
SSMOD external wave table register2

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

CRU_SSGTBL12_15
Address: Operational Base + offset (0x028c)
SSMOD external wave table register3

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

CRU_SSGTBL16_19
Address: Operational Base + offset (0x0290)
SSMOD external wave table register4

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

CRU_SSGTBL20_23
Address: Operational Base + offset (0x0294)
SSMOD external wave table register5

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

CRU_SSGTBL24_27

Address: Operational Base + offset (0x0298)

SSMOD external wave table register6

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

CRU_SSGTBL28_31

Address: Operational Base + offset (0x029c)

SSMOD external wave table register7

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

CRU_SSGTBL32_35

Address: Operational Base + offset (0x02a0)

SSMOD external wave table register8

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

CRU_SSGTBL36_39

Address: Operational Base + offset (0x02a4)

SSMOD external wave table register9

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

CRU_SSGTBL40_43

Address: Operational Base + offset (0x02a8)

SSMOD external wave table register10

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

CRU_SSGTBL44_47

Address: Operational Base + offset (0x02ac)

SSMOD external wave table register11

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

CRU_SSGTBL48_51

Address: Operational Base + offset (0x02b0)

SSMOD external wave table register12

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

CRU_SSGTBL52_55

Address: Operational Base + offset (0x02b4)
SSMOD external wave table register13

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

CRU_SSGTBL56_59
Address: Operational Base + offset (0x02b8)
SSMOD external wave table register14

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

CRU_SSGTBL60_63
Address: Operational Base + offset (0x02bc)
SSMOD external wave table register15

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

CRU_SSGTBL64_67
Address: Operational Base + offset (0x02c0)
SSMOD external wave table register16

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

CRU_SSGTBL68_71

Address: Operational Base + offset (0x02c4)

SSMOD external wave table register17

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

CRU_SSGTBL72_75

Address: Operational Base + offset (0x02c8)

SSMOD external wave table register18

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

CRU_SSGTBL76_79

Address: Operational Base + offset (0x02cc)

SSMOD external wave table register19

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

CRU_SSGTBL80_83

Address: Operational Base + offset (0x02d0)

SSMOD external wave table register20

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ssgtbl80_83 Extern wave table 80-83 7-0: table80 15-8: table81 23-16: table82 31-24: table83

CRU_SSGTBL84_87

Address: Operational Base + offset (0x02d4)

SSMOD external wave table register21

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

CRU_SSGTBL88_91

Address: Operational Base + offset (0x02d8)

SSMOD external wave table register22

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

CRU_SSGTBL92_95

Address: Operational Base + offset (0x02dc)

SSMOD external wave table register23

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

CRU_SSGTBL96_99

Address: Operational Base + offset (0x02e0)

SSMOD external wave table register24

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

CRU_SSGTBL100_103

Address: Operational Base + offset (0x02e4)

SSMOD external wave table register25

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

CRU_SSGTBL104_107

Address: Operational Base + offset (0x02e8)

SSMOD external wave table register26

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

CRU_SSGTBL108_111

Address: Operational Base + offset (0x02ec)

SSMOD external wave table register27

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

CRU_SSGTBL112_115
Address: Operational Base + offset (0x02f0)
SSMOD external wave table register28

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

CRU_SSGTBL116_119
Address: Operational Base + offset (0x02f4)
SSMOD external wave table register29

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

CRU_SSGTBL120_123
Address: Operational Base + offset (0x02f8)
SSMOD external wave table register30

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

CRU_SSGTBL124_127
Address: Operational Base + offset (0x02fc)
SSMOD external wave table register31

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

CRU_SOFTRST_CON0

Address: Operational Base + offset (0x0300)

Internal software reset control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	l2_srstn_req l2 reset request bit "When HIGH, reset relative logic
14	RW	0x0	strc_sys_asrstn_req bus niu aresetn request bit "When HIGH, reset relative logic
13	RW	0x0	core_niu_srstn_req core_niu reset request bit "When HIGH, reset relative logic
12	RW	0x0	topdbg_srstn_req dap presetn request bit "When HIGH, reset relative logic
11	RW	0x0	core3_dbg_srstn_req core3_dbg reset request bit "When HIGH, reset relative logic
10	RW	0x0	core2_dbg_srstn_req core2_dbg reset request bit "When HIGH, reset relative logic
9	RW	0x0	core1_dbg_srstn_req core1_dbg reset request bit "When HIGH, reset relative logic
8	RW	0x0	core0_dbg_srstn_req core0_dbg reset request bit "When HIGH, reset relative logic
7	RW	0x0	core3_srstn_req core3 reset request bit "When HIGH, reset relative logic
6	RW	0x0	core2_srstn_req core2 reset request bit "When HIGH, reset relative logic
5	RW	0x0	core1_srstn_req core1 reset request bit "When HIGH, reset relative logic
4	RW	0x0	core0_srstn_req core0 reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	corepo3_srstn_req corepo3 reset request bit "When HIGH, reset relative logic
2	RW	0x0	corepo2_srstn_req corepo2 reset request bit "When HIGH, reset relative logic
1	RW	0x0	corepo1_srstn_req corepo1 reset request bit "When HIGH, reset relative logic
0	RW	0x0	corepo0_srstn_req corepo0 reset request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON1

Address: Operational Base + offset (0x0304)

Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	gpio3_srstn_req gpio3 reset request bit "When HIGH, reset relative logic
14	RW	0x0	gpio2_srstn_req gpio2 reset request bit "When HIGH, reset relative logic
13	RW	0x0	gpio1_srstn_req gpio1 reset request bit "When HIGH, reset relative logic
12	RW	0x0	gpio0_srstn_req gpio0 reset request bit "When HIGH, reset relative logic
11	RW	0x0	rom_srstn_req rom reset request bit "When HIGH, reset relative logic
10	RW	0x0	intmem_srstn_req intmem reset request bit "When HIGH, reset relative logic
9	RW	0x0	spdif_srstn_req spdif reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
8	RW	0x0	bussys_psrstn_req bus niu presetn request bit "When HIGH, reset relative logic
7	RW	0x0	bussys_hsrstn_req bus niu hresetn request bit "When HIGH, reset relative logic
6	RW	0x0	efuse_srstn_req efuse reset request bit "When HIGH, reset relative logic
5	RW	0x0	pmu_psrstn_req pmu presetn request bit "When HIGH, reset relative logic
4	RO	0x0	Reserved
3	RW	0x0	dap_srstn_req dap reset request bit "When HIGH, reset relative logic
2	RW	0x0	a53_gic_srstn_req a53_gic reset request bit "When HIGH, reset relative logic
1:0	RO	0x0	Reserved

CRU_SOFTRST_CON2

Address: Operational Base + offset (0x0308)

Internal software reset control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2c3_srstn_req i2c3 reset request bit "When HIGH, reset relative logic
14	RW	0x0	i2c2_srstn_req i2c2 reset request bit "When HIGH, reset relative logic
13	RW	0x0	i2c1_srstn_req i2c1 reset request bit "When HIGH, reset relative logic
12	RW	0x0	i2c0_srstn_req i2c0 reset request bit "When HIGH, reset relative logic
11	RW	0x0	uart2_psrstn_req uart2 presetn request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	uart1_psrstn_req uart1 presetn request bit "When HIGH, reset relative logic
9	RW	0x0	uart0_psrstn_req uart0 presetn request bit "When HIGH, reset relative logic
8	RW	0x0	uart2_srstn_req uart2 reset request bit "When HIGH, reset relative logic
7	RW	0x0	uart1_srstn_req uart1 reset request bit "When HIGH, reset relative logic
6	RW	0x0	uart0_srstn_req uart0 reset request bit "When HIGH, reset relative logic
5	RW	0x0	i2s2_hsrstn_req i2s2 hresetn request bit "When HIGH, reset relative logic
4	RW	0x0	i2s1_hsrstn_req i2s1 hresetn request bit "When HIGH, reset relative logic
3	RW	0x0	i2s0_hsrstn_req i2s0 hresetn request bit "When HIGH, reset relative logic
2	RW	0x0	i2s2_srstn_req i2s2 reset request bit "When HIGH, reset relative logic
1	RW	0x0	i2s1_srstn_req i2s1 reset request bit "When HIGH, reset relative logic
0	RW	0x0	i2s0_srstn_req i2s0 reset request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON3

Address: Operational Base + offset (0x030c)

Internal software reset control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	dcf_psrstn_req dcf presetn request bit "When HIGH, reset relative logic
13	RW	0x0	dcf_asrstn_req dcf aresetn request bit "When HIGH, reset relative logic
12	RW	0x0	tsp_hsadc_srstn_req tsp_hsadc reset request bit "When HIGH, reset relative logic
11	RW	0x0	tsp_srstn_req tsp reset request bit "When HIGH, reset relative logic
10	RW	0x0	tsp_hsrstn_req tsp hresetn request bit "When HIGH, reset relative logic
9	RW	0x0	tsp_asrstn_req tsp aresetn request bit "When HIGH, reset relative logic
8	RW	0x0	dma_srstn_req dma reset request bit "When HIGH, reset relative logic
7	RW	0x0	pwm0_psrstn_req pwm0 presetn request bit "When HIGH, reset relative logic
6	RW	0x0	pwm0_srstn_req pwm0 reset request bit "When HIGH, reset relative logic
5	RW	0x0	efuse_ns_psrstn_req efuse_ns presetn request bit "When HIGH, reset relative logic
4	RW	0x0	efuse_se_psrstn_req efuse_se presetn request bit "When HIGH, reset relative logic
3	RW	0x0	i2c3_psrstn_req i2c3 presetn request bit "When HIGH, reset relative logic
2	RW	0x0	i2c2_psrstn_req i2c2 presetn request bit "When HIGH, reset relative logic
1	RW	0x0	i2c1_psrstn_req i2c1 presetn request bit "When HIGH, reset relative logic
0	RW	0x0	i2c0_psrstn_req i2c0 presetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON4
Address: Operational Base + offset (0x0310)
Internal software reset control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb3grf_srstn_req usb3grf reset request bit "When HIGH, reset relative logic
14	RW	0x0	timer5_srstn_req timer5 reset request bit "When HIGH, reset relative logic
13	RW	0x0	timer4_srstn_req timer4 reset request bit "When HIGH, reset relative logic
12	RW	0x0	timer3_srstn_req timer3 reset request bit "When HIGH, reset relative logic
11	RW	0x0	timer2_srstn_req timer2 reset request bit "When HIGH, reset relative logic
10	RW	0x0	timer1_srstn_req timer1 reset request bit "When HIGH, reset relative logic
9	RW	0x0	timer0_srstn_req timer0 reset request bit "When HIGH, reset relative logic
8	RW	0x0	timer_6ch_psrstn_req timer_6ch presetn request bit "When HIGH, reset relative logic
7	RW	0x0	usb_grf_srstn_req usb_grf reset request bit "When HIGH, reset relative logic
6	RW	0x0	grf_srstn_req grf reset request bit "When HIGH, reset relative logic
5	RW	0x0	sgrf_srstn_req sgrf reset request bit "When HIGH, reset relative logic
4	RW	0x0	crypto_srstn_req crypto reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
3	RW	0x0	tsadc_psrstn_req tsadc presetn request bit "When HIGH, reset relative logic
2	RW	0x0	tsadc_srstn_req tsadc reset request bit "When HIGH, reset relative logic
1	RW	0x0	spi0_srstn_req spi0 reset request bit "When HIGH, reset relative logic
0	RW	0x0	scr_srstn_req scr reset request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON5

Address: Operational Base + offset (0x0314)

Internal software reset control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	ddrphy_psrstn_req ddrphy presetn request bit "When HIGH, reset relative logic
14	RW	0x0	ddrphy_srstn_req ddrphy reset request bit "When HIGH, reset relative logic
13	RW	0x0	ddrctrl_psrstn_req ddrctrl presetn request bit "When HIGH, reset relative logic
12	RW	0x0	ddrctrl_srstn_req ddrctrl reset request bit "When HIGH, reset relative logic
11	RW	0x0	ddrmsch_srstn_req ddrmsch reset request bit "When HIGH, reset relative logic
10	RO	0x0	reserved
9	RW	0x0	msch_srstn_req msch reset request bit "When HIGH, reset relative logic
8	RW	0x0	dfimon_srstn_req dfimon reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
7	RW	0x0	grf_ddr_srstn_req grf_ddr reset request bit "When HIGH, reset relative logic
6	RW	0x0	saradc_psrstn_req saradc presetn request bit "When HIGH, reset relative logic
5	RW	0x0	saradc_srstn_req saradc reset request bit "When HIGH, reset relative logic
4	RO	0x0	reserved
3	RW	0x0	acodec_psrstn_req acodec presetn request bit "When HIGH, reset relative logic
2	RW	0x0	vdac_srstn_req vdac reset request bit "When HIGH, reset relative logic
1	RW	0x0	hdmiphy_srstn_req hdmiphy reset request bit "When HIGH, reset relative logic
0	RW	0x0	phyniu_srstn_req phyniu reset request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON6
Address: Operational Base + offset (0x0318)
Internal software reset control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	emmc_srstn_req emmc reset request bit "When HIGH, reset relative logic
14	RW	0x0	sdio_srstn_req sdio reset request bit "When HIGH, reset relative logic
13	RW	0x0	mmc0_srstn_req mmc0 reset request bit "When HIGH, reset relative logic
12	RW	0x0	periphsys_hsrstn_req periph_niu hresetn request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
11	RW	0x0	periph_niu_psrstn_req periph_niu presetn request bit "When HIGH, reset relative logic
10	RW	0x0	periph_niu_hsrstn_req periph_niu hresetn request bit "When HIGH, reset relative logic
9	RW	0x0	periph_niu_asrstn_req periph_niu aresetn request bit "When HIGH, reset relative logic
8	RW	0x0	sdmmcext_srstn_req sdmmcext reset request bit "When HIGH, reset relative logic
7	RW	0x0	gpu_niu_asrstn_req gpu_niu aresetn request bit "When HIGH, reset relative logic
6	RW	0x0	gpu_asrstn_req gpu aresetn request bit "When HIGH, reset relative logic
5	RW	0x0	otp_phy_srstn_req otp_phy reset request bit "When HIGH, reset relative logic
4	RW	0x0	macphy_srstn_req macphy reset request bit "When HIGH, reset relative logic
3	RW	0x0	gmac2io_asrstn_req gmac2io aresetn request bit "When HIGH, reset relative logic
2	RW	0x0	gmac2phy_asrstn_req gmac2phy aresetn request bit "When HIGH, reset relative logic
1	RW	0x0	gmac_niu_psrstn_req gmac_niu presetn request bit "When HIGH, reset relative logic
0	RW	0x0	gmac_niu_asrstn_req gmac_niu aresetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON7

Address: Operational Base + offset (0x031c)

Internal software reset control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb3phy_pipe_srstn_req usb3phy_pipe reset request bit "When HIGH, reset relative logic
14	RW	0x0	usb3phy_u3_srstn_req usb3phy_u3 reset request bit "When HIGH, reset relative logic
13	RW	0x0	usb3phy_u2_srstn_req usb3phy_u2 reset request bit "When HIGH, reset relative logic
12	RW	0x0	usb3otg_utmi_srst_req usb3otg_utmi reset request bit "When HIGH, reset relative logic
11	RW	0x0	usb2host_utmi_srst_req usb2host_utmi reset request bit "When HIGH, reset relative logic
10	RW	0x0	usb2otg_utmi_srst_req usb2otg_utmi reset request bit "When HIGH, reset relative logic
9	RW	0x0	usbpor_srst_req usbpor reset request bit "When HIGH, reset relative logic
8	RW	0x0	usb3otg_srstn_req usb3otg reset request bit "When HIGH, reset relative logic
7	RW	0x0	usb2host_utmi_srstn_req usb2host_utmi reset request bit "When HIGH, reset relative logic
6	RW	0x0	usb2host_ehciphy_srstn_req usb2host_ehciphy reset request bit "When HIGH, reset relative logic
5	RW	0x0	usb2host_aux_srstn_req usb2host_aux reset request bit "When HIGH, reset relative logic
4	RW	0x0	usb2host_arb_srstn_req usb2host_arb reset request bit "When HIGH, reset relative logic
3	RW	0x0	usb2host_hsrstn_req usb2host hresetn request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	usb2otg_adp_srstn_req usb2otg_adp reset request bit "When HIGH, reset relative logic
1	RW	0x0	usb2otg_srstn_req usb2otg reset request bit "When HIGH, reset relative logic
0	RW	0x0	usb2otg_hsrstn_req usb2otg hresetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON8

Address: Operational Base + offset (0x0320)

Internal software reset control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hdmi_psrstn_req hdmi presetn request bit "When HIGH, reset relative logic
14	RW	0x0	hdmi_srstn_req hdmi reset request bit "When HIGH, reset relative logic
13	RW	0x0	iep_hsrstn_req iep hresetn request bit "When HIGH, reset relative logic
12	RW	0x0	iep_asrstn_req iep aresetn request bit "When HIGH, reset relative logic
11	RW	0x0	rga_hsrstn_req rga hresetn request bit "When HIGH, reset relative logic
10	RW	0x0	rga_asrstn_req rga aresetn request bit "When HIGH, reset relative logic
9	RW	0x0	rga_niu_asrstn_req rga_niu aresetn request bit "When HIGH, reset relative logic
8	RW	0x0	rga_srstn_req rga reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
7	RW	0x0	vop_dsrstn_req vop dresetn request bit "When HIGH, reset relative logic
6	RW	0x0	vop_hsrstn_req vop hresetn request bit "When HIGH, reset relative logic
5	RW	0x0	vop_asrstn_req vop aresetn request bit "When HIGH, reset relative logic
4	RW	0x0	vop_niu_asrstn_req vop_niu aresetn request bit "When HIGH, reset relative logic
3	RW	0x0	vio_arbi_hsrstn_req vio_arbi hresetn request bit "When HIGH, reset relative logic
2	RW	0x0	vio_h2p_hsrstn_req vio_h2p hresetn request bit "When HIGH, reset relative logic
1	RW	0x0	vio_bus_hsrstn_req vio_bus hresetn request bit "When HIGH, reset relative logic
0	RW	0x0	vio_asrstn_req vio aresetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON9

Address: Operational Base + offset (0x0324)

Internal software reset control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb3phy_pipe_psrstn_req usb3phy_pipe presetn request bit "When HIGH, reset relative logic
14	RW	0x0	usb3phy_otg_psrstn_req usb3phy_otg presetn request bit "When HIGH, reset relative logic
13	RW	0x0	pdm_srstn_req pdm reset request bit "When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	pdm_hsrstn_req pdm hresetn request bit "When HIGH, reset relative logic
11	RW	0x0	ddrstdy_srstn_req ddrstdy reset request bit "When HIGH, reset relative logic
10	RW	0x0	ddrstdy_psrstn_req ddrstdy presetn request bit "When HIGH, reset relative logic
9	RW	0x0	ddrctrl_asrstn_req ddrctrl aresetn request bit "When HIGH, reset relative logic
8	RW	0x0	otp_user_srstn_req otp_user reset request bit "When HIGH, reset relative logic
7	RW	0x0	otp_sbpi_srstn_req otp_sbpi reset request bit "When HIGH, reset relative logic
6	RW	0x0	otp_psrstn_req otp presetn request bit "When HIGH, reset relative logic
5	RW	0x0	cif_psrstn_req cif presetn request bit "When HIGH, reset relative logic
4	RW	0x0	cif_hsrstn_req cif hresetn request bit "When HIGH, reset relative logic
3	RW	0x0	cif_asrstn_req cif aresetn request bit "When HIGH, reset relative logic
2	RW	0x0	hdcp_hsrstn_req hdcp hresetn request bit "When HIGH, reset relative logic
1	RW	0x0	hdcp_srstn_req hdcp reset request bit "When HIGH, reset relative logic
0	RW	0x0	hdcp_asrstn_req hdcp aresetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON10
Address: Operational Base + offset (0x0328)
Internal software reset control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	ddrphydiv_srstn_req ddrphydiv reset request bit "When HIGH, reset relative logic
14:10	RO	0x0	reserved
9	RW	0x0	vdec_cabac_srstn_req vdec_cabac reset request bit "When HIGH, reset relative logic
8	RW	0x0	vdec_core_srstn_req vdec_core reset request bit "When HIGH, reset relative logic
7	RW	0x0	vdec_niu_hsrstn_req vdec_niu hresetn request bit "When HIGH, reset relative logic
6	RW	0x0	vdec_hsrstn_req vdec hresetn request bit "When HIGH, reset relative logic
5	RW	0x0	vdec_niu_asrstn_req vdec_niu aresetn request bit "When HIGH, reset relative logic
4	RW	0x0	vdec_asrstn_req vdec aresetn request bit "When HIGH, reset relative logic
3	RW	0x0	vcodec_niu_hsrstn_req vcodec_niu hresetn request bit "When HIGH, reset relative logic
2	RW	0x0	vcodec_hsrstn_req vcodec hresetn request bit "When HIGH, reset relative logic
1	RW	0x0	vcodec_niu_asrstn_req vcodec_niu aresetn request bit "When HIGH, reset relative logic
0	RW	0x0	vcodec_asrstn_req vcodec aresetn request bit "When HIGH, reset relative logic

CRU_SOFTRST_CON11
Address: Operational Base + offset (0x032c)
Internal software reset control register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	rkvenc_intmem_srstn_req rkvenc_intmem reset request bit "When HIGH, reset relative logic
7	RW	0x0	rkvenc_h264_hsrstn_req rkvenc_h264 hresetn request bit "When HIGH, reset relative logic
6	RW	0x0	rkvenc_h264_asrstn_req rkvenc_h264 aresetn request bit "When HIGH, reset relative logic
5	RW	0x0	rkvenc_h265_dsp_srstn_req rkvenc_h265_dsp reset request bit "When HIGH, reset relative logic
4	RW	0x0	rkvenc_h265_core_srstn_req rkvenc_h265_core reset request bit "When HIGH, reset relative logic
3	RW	0x0	rkvenc_h265_psrstn_req rkvenc_h265 presetn request bit "When HIGH, reset relative logic
2	RW	0x0	rkvenc_h265_asrstn_req rkvenc_h265 aresetn request bit "When HIGH, reset relative logic
1	RW	0x0	rkvenc_niu_hsrstn_req rkvenc_niu hresetn request bit "When HIGH, reset relative logic
0	RW	0x0	rkvenc_niu_asrstn_req rkvenc_niu aresetn request bit "When HIGH, reset relative logic

CRU_CRU_SDMMC_CON0

Address: Operational Base + offset (0x0380)

sdmmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	drv_sel drive select drive select
10:3	RW	0x00	drv_delaynum drive delay number drive delay number
2:1	RW	0x2	drv_degree drive degree drive degree
0	RW	0x0	init_state initial state initial state

CRU_CRU_SDMMC_CON1

Address: Operational Base + offset (0x0384)

sdmmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	sample_sel sample select sample select
9:2	RW	0x00	sample_delaynum sample delay number sample delay number
1:0	RW	0x0	sample_degree sample degree sample degree

CRU_CRU_SDIO_CON0

Address: Operational Base + offset (0x0388)

SDIO control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	drv_sel drive select drive select
10:3	RW	0x00	drv_delaynum drive delay number drive delay number
2:1	RW	0x2	drv_degree drive degree drive degree
0	RW	0x0	init_state initial state initial state

CRU_CRU_SDIO_CON1

Address: Operational Base + offset (0x038c)

SDIO control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	sample_sel sample select sample select
9:2	RW	0x00	sample_delaynum sample delay number sample delay number
1:0	RW	0x0	sample_degree sample degree sample degree

CRU_CRU_EMMC_CON0

Address: Operational Base + offset (0x0390)

EMMC control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	drv_sel drive select drive select
10:3	RW	0x00	drv_delaynum drive delay number drive delay number
2:1	RW	0x2	drv_degree drive degree drive degree
0	RW	0x0	init_state initial state initial state

CRU_CRU_EMMC_CON1

Address: Operational Base + offset (0x0394)

EMMC control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	sample_sel sample select sample select
9:2	RW	0x00	sample_delaynum sample delay number sample delay number
1:0	RW	0x0	sample_degree sample degree sample degree

CRU_CRU_SDMMC_EXT_CON0

Address: Operational Base + offset (0x0398)

SDMMC_EXT control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11	RW	0x0	drv_sel drive select drive select
10:3	RW	0x00	drv_delaynum drive delay number drive delay number
2:1	RW	0x2	drv_degree drive degree drive degree
0	RW	0x0	init_state initial state initial state

CRU_CRU_SDMMC_EXT_CON1

Address: Operational Base + offset (0x039c)

SDMMC_EXT control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask bits "When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	sample_sel sample select sample select
9:2	RW	0x00	sample_delaynum sample delay number sample delay number
1:0	RW	0x0	sample_degree sample degree sample degree

2.7 Timing Diagram

Power on reset timing is shown as follow:

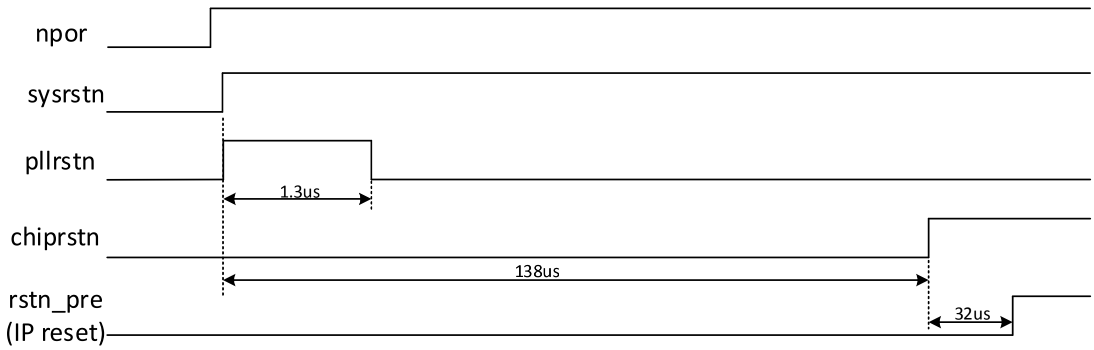


Fig. 2-4 Chip Power On Reset Timing Diagram

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

2.8 Application Notes

2.8.1 PLL usage

A. PLL output frequency configuration

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_APLL_CON0, CRU_DPLL_CON0 and CRU_GPLL_CON0.
DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_APLL_CON1, CRU_DPLL_CON1 and CRU_GPLL_CON1.
FRAC can be configured by programming CRU_APLL_CON2, CRU_DPLL_CON2 and CRU_GPLL_CON2.
If DSMPD = 1 (DSM is disabled, "integer mode")
FOUTVCO = FREF / REFDIV * FBDIV
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2
When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

DSMPD = 1
REFDIV = 6
FBDIV = 175
POSTDIV1=1
POSTDIV2=1

And then
FOUTVCO = FREF / REFDIV * FBDIV = 24/6*175=700
FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2=700/1/1=700
If DSMPD = 0 (DSM is enabled, "fractional mode")
FOUTVCO = FREF / REFDIV * (FBDIV + FRAC / 224)
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 / 224) = 24/1*(40+24'hf5c28f /224)= 983.04

FOUTPOSTDIV = FOUTVCO / POSTDIV1 / POSTDIV2=983.04/2/1=491.52

B. PLL setting consideration

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

2.8.2 PLL frequency change and lock check

The PLL programming supports changed on-the-fly and the PLL will simply slew to the new frequency.

PLL lock state can be checked in CRU_APLL_CON1[10], CRU_DPLL_CON1[10], CRU_CPLL_CON1[10], CRU_GPLL_CON1[10] register. The lock state is high when both original hardware PLL lock and PLL counter lock are high. The PLL counter lock initial value is CRU_GLB_CNT_TH[31:16].

The max delay time is 500 REF_CLK.

PLL locking consists of three phases.

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

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

2.8.3 Fractional divider usage

To get specific frequency, clocks of I2S, SPDIF, UART can be generated by fractional divider. Generally you must set that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock like I2S, UART.

2.8.4 Global software reset

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST_VALUE[15:0] as 0xfdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND_VALUE[15:0] as 0xeca8 to assert the second global software reset glb_srstn_2. These two software resets are self-deasserted by hardware.

Glb_srstn_1 resets almost all logic.

Glb_srstn_2 resets almost all logic except GRF and GPIOs.

2.8.5 Restriction

a The HDMI controller apb bus is connected to NIU (Network interface Unit) through a h2p bridge. So if HDMI is needed, make sure hclk_h2p_en and pclk_h2p_en (cru_clkgate_con21 bit 13 and bit 14) is disabled to open the clock for h2p bridge.

b The AXI bus of RGA/IEP/HDCP /VIP share same logic in niu of pd_vio. Please make sure the rga_aclk_niu is opened (aclk_rga_niu_en, cru_clkgate_con22 bit 3 is disabled) if either of these controllers is inuse.

c There is a sram shared between H265 and H264. H265 can access this sram by an axi2sram bridge. So if H265 or H264 is enabled, make sure the clock of axi2sram is opend (aclk_axi2sram_en, cru_clkgate_con25 bit6 should be set to disable).

Chapter 3 General Register Files (GRF)

3.1 Overview

The general register file will be used to do static set by software, which is composed of many registers for system control. The GRF is divided into four sections,

- GRF, used for general non-secure system,
- DDR_GRF, used for always on system
- USB2PHY_GRF, used for USB2 PHY control and query
- USB3PHY_GRF, used for USB3 PHY control and query

3.2 Function Description

The function of general register file is:

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

3.3 GRF Register Description

3.3.1 Internal Address Mapping

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

3.3.2 Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0A_IOMUX	0x0000	W	0x00000000	GPIO0A iomux control
GRF_GPIO0B_IOMUX	0x0004	W	0x00000000	GPIO0B iomux control
GRF_GPIO0C_IOMUX	0x0008	W	0x00000000	GPIO0C iomux control
GRF_GPIO0D_IOMUX	0x000c	W	0x00000000	GPIO0D iomux control
GRF_GPIO1A_IOMUX	0x0010	W	0x000004aa	GPIO1A iomux control
GRF_GPIO1B_IOMUX	0x0014	W	0x00000000	GPIO1B iomux control
GRF_GPIO1C_IOMUX	0x0018	W	0x00000000	GPIO1C iomux control
GRF_GPIO1D_IOMUX	0x001c	W	0x00000000	GPIO1D iomux control
GRF_GPIO2A_IOMUX	0x0020	W	0x00000000	GPIO2A iomux control
GRF_GPIO2BL_IOMUX	0x0024	W	0x00000200	GPIO2BL iomux control
GRF_GPIO2BH_IOMUX	0x0028	W	0x00000000	GPIO2BH iomux control

Name	Offset	Size	Reset Value	Description
GRF_GPIO2CL_IOMUX	0x002c	W	0x00000000	GPIO2CL iomux control
GRF_GPIO2CH_IOMUX	0x0030	W	0x00000000	GPIO2CH iomux control
GRF_GPIO2D_IOMUX	0x0034	W	0x00000000	GPIO2D iomux control
GRF_GPIO3AL_IOMUX	0x0038	W	0x00000000	GPIO3AL iomux control
GRF_GPIO3AH_IOMUX	0x003c	W	0x00000000	GPIO3AH iomux control
GRF_GPIO3BL_IOMUX	0x0040	W	0x00000000	GPIO3BL iomux control
GRF_GPIO3BH_IOMUX	0x0044	W	0x00000000	GPIO3BH iomux control
GRF_GPIO3C_IOMUX	0x0048	W	0x00000000	GPIO3C iomux control
GRF_GPIO3D_IOMUX	0x004c	W	0x00000000	GPIO3D iomux control
GRF_COM_IOMUX	0x0050	W	0x00000000	GRF common iomux control
GRF_GPIO0A_P	0x0100	W	0x0000566a	GPIO0A PU/PD control
GRF_GPIO0B_P	0x0104	W	0x0000aa6a	GPIO0B PU/PD control
GRF_GPIO0C_P	0x0108	W	0x0000aa6a	GPIO0C PU/PD control
GRF_GPIO0D_P	0x010c	W	0x0000aaaa	GPIO0D PU/PD control
GRF_GPIO1A_P	0x0110	W	0x0000a555	GPIO1A PU/PD control
GRF_GPIO1B_P	0x0114	W	0x000056a5	GPIO1B PU/PD control
GRF_GPIO1C_P	0x0118	W	0x00009a65	GPIO1C PU/PD control
GRF_GPIO1D_P	0x011c	W	0x0000aaaa	GPIO1D PU/PD control
GRF_GPIO2A_P	0x0120	W	0x00009556	GPIO2A PU/PD control
GRF_GPIO2B_P	0x0124	W	0x0000959a	GPIO2B PU/PD control
GRF_GPIO2C_P	0x0128	W	0x00005565	GPIO2C PU/PD control
GRF_GPIO2D_P	0x012c	W	0x000055a5	GPIO2D PU/PD control
GRF_GPIO3A_P	0x0130	W	0x000055a5	GPIO3A PU/PD control
GRF_GPIO3B_P	0x0134	W	0x00005aaa	GPIO3B PU/PD control
GRF_GPIO3C_P	0x0138	W	0x00006555	GPIO3C PU/PD control
GRF_GPIO3D_P	0x013c	W	0x0000555a	GPIO3D PU/PD control
GRF_GPIO0A_E	0x0200	W	0x00008011	GPIO0A drive strength control
GRF_GPIO0B_E	0x0204	W	0x0000aa2a	GPIO0B drive strength control
GRF_GPIO0C_E	0x0208	W	0x0000aa0a	GPIO0C drive strength control

Name	Offset	Size	Reset Value	Description
GRF_GPIO0D_E	0x020c	W	0x0000005a	GPIO0D drive strength control
GRF_GPIO1A_E	0x0210	W	0x0000aaaa	GPIO1A drive strength control
GRF_GPIO1B_E	0x0214	W	0x0000aa2a	GPIO1B drive strength control
GRF_GPIO1C_E	0x0218	W	0x0000a88a	GPIO1C drive strength control
GRF_GPIO1D_E	0x021c	W	0x0000005a	GPIO1D drive strength control
GRF_GPIO2A_E	0x0220	W	0x00000000	GPIO2A drive strength control
GRF_GPIO2B_E	0x0224	W	0x00004145	GPIO2B drive strength control
GRF_GPIO2C_E	0x0228	W	0x00005515	GPIO2C drive strength control
GRF_GPIO2D_E	0x022c	W	0x0000aa01	GPIO2D drive strength control
GRF_GPIO3A_E	0x0230	W	0x0000aa22	GPIO3A drive strength control
GRF_GPIO3B_E	0x0234	W	0x00000000	GPIO3B drive strength control
GRF_GPIO3C_E	0x0238	W	0x0000aaaa	GPIO3C drive strength control
GRF_GPIO3D_E	0x023c	W	0x0000aaaa	GPIO3D drive strength control
GRF_GPIO0L_SR	0x0300	W	0x00000000	GPIO0 A/B SR control
GRF_GPIO0H_SR	0x0304	W	0x00000000	GPIO0 C/D SR control
GRF_GPIO1L_SR	0x0308	W	0x00000000	GPIO1 A/B SR control
GRF_GPIO1H_SR	0x030c	W	0x00000000	GPIO1 C/D SR control
GRF_GPIO2L_SR	0x0310	W	0x00000000	GPIO2 A/B SR control
GRF_GPIO2H_SR	0x0314	W	0x00000000	GPIO2 C/D SR control
GRF_GPIO3L_SR	0x0318	W	0x00000000	GPIO3 A/B SR control
GRF_GPIO3H_SR	0x031c	W	0x00000000	GPIO3 C/D SR control
GRF_GPIO0L_SMT	0x0380	W	0x00000000	GPIO0 A/B smitter control register
GRF_GPIO0H_SMT	0x0384	W	0x00000000	GPIO0 C/D smitter control register
GRF_GPIO1L_SMT	0x0388	W	0x00000000	GPIO1 A/B smitter control register
GRF_GPIO1H_SMT	0x038c	W	0x00000000	GPIO1 C/D smitter control register
GRF_GPIO2L_SMT	0x0390	W	0x00000000	GPIO2 A/B smitter control register
GRF_GPIO2H_SMT	0x0394	W	0x00000000	GPIO2 C/D smitter control register
GRF_GPIO3L_SMT	0x0398	W	0x00000000	GPIO3 A/B smitter control register
GRF_GPIO3H_SMT	0x039c	W	0x00000000	GPIO3 C/D smitter control register

Name	Offset	Size	Reset Value	Description
GRF_SOC_CON0	0x0400	W	0x00000000	SOC control register0
GRF_SOC_CON1	0x0404	W	0x00000000	SOC control register1
GRF_SOC_CON2	0x0408	W	0x00001000	SOC control register2
GRF_SOC_CON3	0x040c	W	0x00000000	SOC control register3
GRF_SOC_CON4	0x0410	W	0x00000000	SOC control register4
GRF_SOC_CON5	0x0414	W	0x00000000	SOC control register5
GRF_SOC_CON6	0x0418	W	0x00000000	SOC control register6
GRF_SOC_CON7	0x041c	W	0x00000000	SOC control register7
GRF_SOC_CON8	0x0420	W	0x00000000	SOC control register8
GRF_SOC_CON9	0x0424	W	0x00000000	SOC control register9
GRF_SOC_CON10	0x0428	W	0x0000f800	SOC control register10
GRF_SOC_STATUS0	0x0480	W	0x00000000	SOC status register0
GRF_SOC_STATUS1	0x0484	W	0x00000000	SOC status register1
GRF_SOC_STATUS2	0x0488	W	0x00000000	SOC status register2
GRF_SOC_STATUS3	0x048c	W	0x00000000	SOC status register3
GRF_SOC_STATUS4	0x0490	W	0x00000000	SOC status register4
GRF_USB3OTG_CON0	0x04c0	W	0x00002000	USB3OTG control register0
GRF_USB3OTG_CON1	0x04c4	W	0x00001100	USB3OTG control register1
GRF_CPU_CON0	0x0500	W	0x00000060	CPU control register0
GRF_CPU_CON1	0x0504	W	0x0000000c	CPU control register1
GRF_CPU_STATUS0	0x0520	W	0x00000000	CPU status register0
GRF_CPU_STATUS1	0x0524	W	0x00000000	CPU status register1
GRF_OS_REG0	0x05c8	W	0x00000000	os register0
GRF_OS_REG1	0x05cc	W	0x00000000	os register1
GRF_OS_REG2	0x05d0	W	0x00000000	os register2
GRF_OS_REG3	0x05d4	W	0x00000000	os register3
GRF_OS_REG4	0x05d8	W	0x00000000	os register4
GRF_OS_REG5	0x05dc	W	0x00000000	os register5
GRF_OS_REG6	0x05e0	W	0x00000000	os register6

Name	Offset	Size	Reset Value	Description
GRF_OS_REG7	0x05e4	W	0x00000000	os register7
GRF_SIG_DETECT_CON	0x0680	W	0x00000000	External signal detect configue register
GRF_SIG_DETECT_STATUS	0x0690	W	0x00000000	External signal detect status register
GRF_SIG_DETECT_STATUS_CLEAR	0x06a0	W	0x00000000	External signal detect status clear register
GRF_SDMMC_DET_COUNTER	0x06b0	W	0x00030100	SDMMC detect counter register
GRF_HOST0_CON0	0x0700	W	0x00000820	host0 control register0
GRF_HOST0_CON1	0x0704	W	0x000004bc	host0 control register1
GRF_HOST0_CON2	0x0708	W	0x00000019	host0 control register2
GRF_OTG_CON0	0x0880	W	0x00000000	OTG control register
GRF_HOST0_STATUS	0x0890	W	0x00000000	HOST0 status register
GRF_MAC_CON0	0x0900	W	0x00000000	MAC control register0
GRF_MAC_CON1	0x0904	W	0x00000000	MAC control register1
GRF_MAC_CON2	0x0908	W	0x00000000	MAC control register2
GRF_MACPHY_CON0	0x0b00	W	0x00002039	MACPHY control register0
GRF_MACPHY_CON1	0x0b04	W	0x00000000	MACPHY control register1
GRF_MACPHY_CON2	0x0b08	W	0x00000000	MACPHY control register2
GRF_MACPHY_CON3	0x0b0c	W	0x00000000	MACPHY control register3
GRF_MACPHY_STATUS	0x0b10	W	0x00000000	MACPHY status register

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

3.3.3 Detail Register Description

GRF_GPIO0A_IOMUX

Address: Operational Base + offset (0x0000)

GPIO0A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio0_a7_sel GPIO0A[7] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d0 2'b11: reserved
13:10	RO	0x0	reserved
9:8	RW	0x0	gpio0_a4_sel GPIO0A[4] iomux select 2'b00: gpio 2'b01: hdmi_hdp 2'b10: reserved 2'b11: reserved
7:6	RO	0x0	reserved
5:4	RW	0x0	gpio0_a2_sel GPIO0A[2] iomux select 2'b00: gpio 2'b01: clk_out_gmacm0 2'b10: spdif_txm2 2'b11: reserved
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio0_a0_sel GPIO0A[0] iomux select 2'b00: gpio 2'b01: clk_out_wifim0 2'b10: reserved 2'b11: reserved

GRF_GPIO0B_IOMUX

Address: Operational Base + offset (0x0004)

GPIO0B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_GPIO0C_IOMUX

Address: Operational Base + offset (0x0008)

GPIO0C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_GPIO0D_IOMUX

Address: Operational Base + offset (0x000c)

GPIO0D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio0_d6_sel GPIO0D[6] iomux select 2'b00: gpio 2'b01: fephyled_speed10 2'b10: fephyled_duplex 2'b11: sdmmc0_pwrenm1
11:8	RO	0x0	reserved
7:6	RW	0x0	gpio0_d3_sel GPIO0D[3] iomux select 2'b00: gpio 2'b01: spdif_txm0 2'b10: reserved 2'b11: reserved
5:0	RO	0x0	reserved

GRF_GPIO1A_IOMUX

Address: Operational Base + offset (0x0010)

GPIO1A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	gpio1_a6_sel GPIO1A[6] iomux select 2'b00: gpio 2'b01: sdmmc0_clkout 2'b10: test_clk0 2'b11: reserved
11:10	RW	0x1	gpio1_a5_sel GPIO1A[5] iomux select 2'b00: gpio 2'b01: sdmmc0_detn 2'b10: reserved 2'b11: reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio1_a4_sel GPIO1A[4] iomux select 2'b00: gpio 2'b01: sdmmc0_cmd 2'b10: reserved 2'b11: reserved
7:6	RW	0x2	gpio1_a3_sel GPIO1A[3] iomux select 2'b00: gpio 2'b01: sdmmc0_d3 2'b10: jtag_tms 2'b11: reserved
5:4	RW	0x2	gpio1_a2_sel GPIO1A[2] iomux select 2'b00: gpio 2'b01: sdmmc0_d2 2'b10: jtag_tck 2'b11: reserved
3:2	RW	0x2	gpio1_a1_sel GPIO1A[1] iomux select 2'b00: gpio 2'b01: sdmmc0_d1 2'b10: uart2dbg_rxm0 2'b11: reserved
1:0	RW	0x2	gpio1_a0_sel GPIO1A[0] iomux select 2'b00: gpio 2'b01: sdmmc0_d0 2'b10: uart2dbg_txm0 2'b11: reserved

GRF_GPIO1B_IOMUX

Address: Operational Base + offset (0x0014)

GPIO1B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio1_b7_sel GPIO1B[7] iomux select 2'b00: gpio 2'b01: sdmmc1_d1 2'b10: gmac_rxd2m1 2'b11: reserved
13:12	RW	0x0	gpio1_b6_sel GPIO1B[6] iomux select 2'b00: gpio 2'b01: sdmmc1_d0 2'b10: gmac_rxd3m1 2'b11: reserved
11:10	RW	0x0	gpio1_b5_sel GPIO1B[5] iomux select 2'b00: gpio 2'b01: sdmmc1_cmd 2'b10: gmac_rxclkm1 2'b11: reserved
9:8	RW	0x0	gpio1_b4_sel GPIO1B[4] iomux select 2'b00: gpio 2'b01: sdmmc1_clkout 2'b10: gmac_txclkm1 2'b11: reserved
7:6	RW	0x0	gpio1_b3_sel GPIO1B[3] iomux select 2'b00: gpio 2'b01: uart0_ctsn 2'b10: gmac_rxd0m1 2'b11: reserved
5:4	RW	0x0	gpio1_b2_sel GPIO1B[2] iomux select 2'b00: gpio 2'b01: uart0_rtsn 2'b10: gmac_rxd1m1 2'b11: reserved
3:2	RW	0x0	gpio1_b1_sel GPIO1B[1] iomux select 2'b00: gpio 2'b01: uart0_tx 2'b10: gmac_txd0m1 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio1_b0_sel GPIO1B[0] iomux select 2'b00: gpio 2'b01: uart0_rx 2'b10: gmac_txd1m1 2'b11: reserved

GRF_GPIO1C_IOMUX

Address: Operational Base + offset (0x0018)

GPIO1C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1_c7_sel GPIO1C[7] iomux select 2'b00: gpio 2'b01: i2s2_lrcktxm0 2'b10: gmac_mdcm1 2'b11: pdm_sdi0m1
13:12	RW	0x0	gpio1_c6_sel GPIO1C[6] iomux select 2'b00: gpio 2'b01: i2s2_sclkm0 2'b10: gmac_rxdvm1 2'b11: pdm_clkm1
11:10	RW	0x0	gpio1_c5_sel GPIO1C[5] iomux select 2'b00: gpio 2'b01: i2s2_mclk 2'b10: gmac_clkm1 2'b11: reserved
9:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio1_c3_sel GPIO1C[3] iomux select 2'b00: gpio 2'b01: sdmmc1_detn 2'b10: gmac_mdiom1 2'b11: pdm_fsyncm1
5:4	RW	0x0	gpio1_c2_sel GPIO1C[2] iomux select 2'b00: gpio 2'b01: sdmmc1_pwren 2'b10: gmac_crsm1 2'b11: reserved
3:2	RW	0x0	gpio1_c1_sel GPIO1C[1] iomux select 2'b00: gpio 2'b01: sdmmc1_d3 2'b10: gmac_txd2m1 2'b11: reserved
1:0	RW	0x0	gpio1_c0_sel GPIO1C[0] iomux select 2'b00: gpio 2'b01: sdmmc1_d2 2'b10: gmac_txd3m1 2'b11: reserved

GRF_GPIO1D_IOMUX

Address: Operational Base + offset (0x001c)

GPIO1D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio1_d4_sel GPIO1D[4] iomux select 2'b00: gpio 2'b01: clk32k_outm1 2'b10: reserved 2'b11: reserved
7:6	RO	0x0	reserved
5:4	RW	0x0	gpio1_d2_sel GPIO1D[2] iomux select 2'b00: gpio 2'b01: i2s2_lrckrxm0 2'b10: clk_out_gmacm2 2'b11: pdm_sdi3m1
3:2	RW	0x0	gpio1_d1_sel GPIO1D[1] iomux select 2'b00: gpio 2'b01: i2s2_sdom0 2'b10: gmac_txenm1 2'b11: pdm_sdi2m1
1:0	RW	0x0	gpio1_d0_sel GPIO1D[0] iomux select 2'b00: gpio 2'b01: i2s2_sdim0 2'b10: gmac_rxerm1 2'b11: pdm_sdi1m1

GRF_GPIO2A_IOMUX

Address: Operational Base + offset (0x0020)

GPIO2A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio2_a6_sel GPIO2A[6] iomux select 2'b00: gpio 2'b01: pwm_2 2'b10: reserved 2'b11: reserved
11:10	RW	0x0	gpio2_a5_sel GPIO2A[5] iomux select 2'b00: gpio 2'b01: pwm_1 2'b10: i2c1_scl 2'b11: reserved
9:8	RW	0x0	gpio2_a4_sel GPIO2A[4] iomux select 2'b00: gpio 2'b01: pwm_0 2'b10: i2c1_sda 2'b11: reserved
7:6	RW	0x0	gpio2_a3_sel GPIO2A[3] iomux select 2'b00: gpio 2'b01: efuse_pwren 2'b10: power_state3 2'b11: reserved
5:4	RW	0x0	gpio2_a2_sel GPIO2A[2] iomux select 2'b00: gpio 2'b01: pwm_ir 2'b10: power_state2 2'b11: reserved
3:2	RW	0x0	gpio2_a1_sel GPIO2A[1] iomux select 2'b00: gpio 2'b01: uart2dbg_rxm1 2'b10: power_state1 2'b11: reserved
1:0	RW	0x0	gpio2_a0_sel GPIO2A[0] iomux select 2'b00: gpio 2'b01: uart2dbg_txm1 2'b10: power_state0 2'b11: reserved

GRF_GPIO2BL_IOMUX

Address: Operational Base + offset (0x0024)

GPIO2BL iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	reserved
11:10	RO	0x0	reserved
9:8	RW	0x2	gpio2_b4_sel GPIO2B[4] iomux select 2'b00: gpio 2'b01: spi_csn1m0 2'b10: flash_vol_sel 2'b11: reserved
7:0	RO	0x0	reserved

GRF_GPIO2BH_IOMUX

Address: Operational Base + offset (0x0028)

GPIO2BH iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio2_b7_sel GPIO2B[7] iomux select 3'b000: gpio 3'b001: i2s1_mclk 3'b010: reserved 3'b011: tsp_syncm1 3'b100: cif_clkoutm1 3'b101: reserved 3'b110: reserved 3'b111: reserved

GRF_GPIO2CL_IOMUX

Address: Operational Base + offset (0x002c)

GPIO2CL iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:12	RW	0x0	gpio2_c4_sel GPIO2C[4] iomux select 3'b000: gpio 3'b001: i2s1_sdio1 3'b010: pdm_sdi1m0 3'b011: card_rstm1 3'b100: reserved 3'b101: reserved 3'b110: reserved 3'b111: reserved
11:9	RW	0x0	gpio2_c3_sel GPIO2C[3] iomux select 3'b000: gpio 3'b001: i2s1_sdi 3'b010: pdm_sdi0m0 3'b011: card_clkm1 3'b100: reserved 3'b101: reserved 3'b110: reserved 3'b111: reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x0	gpio2_c2_sel GPIO2C[2] iomux select 3'b000: gpio 3'b001: i2s1_sclk 3'b010: pdm_clkm0 3'b011: tsp_d7m1 3'b100: cif_data7m1 3'b101: reserved 3'b110: reserved 3'b111: reserved
5:3	RW	0x0	gpio2_c1_sel GPIO2C[1] iomux select 3'b000: gpio 3'b001: i2s1_lrcktx 3'b010: spdif_txm1 3'b011: tsp_d6m1 3'b100: cif_data6m1 3'b101: reserved 3'b110: reserved 3'b111: reserved
2:0	RW	0x0	gpio2_c0_sel GPIO2C[0] iomux select 3'b000: gpio 3'b001: i2s1_lrckrx 3'b010: reserved 3'b011: tsp_d5m1 3'b100: cif_data5m1 3'b101: reserved 3'b110: reserved 3'b111: reserved

GRF_GPIO2CH_IOMUX

Address: Operational Base + offset (0x0030)

GPIO2CH iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio2_c7_sel GPIO2C[7] iomux select 2'b00: gpio 2'b01: i2s1_sdo 2'b10: pdm_fsyncm0 2'b11: reserved
13:6	RO	0x0	reserved
5:3	RW	0x0	gpio2_c6_sel GPIO2C[6] iomux select 3'b000: gpio 3'b001: i2s1_sdio3 3'b010: pdm_sdi3m0 3'b011: card_iom1 3'b100: reserved 3'b101: reserved 3'b110: reserved 3'b111: reserved
2:0	RW	0x0	gpio2_c5_sel GPIO2C[5] iomux select 3'b000: gpio 3'b001: i2s1_sdio2 3'b010: pdm_sdi2m0 3'b011: card_detm1 3'b100: reserved 3'b101: reserved 3'b110: reserved 3'b111: reserved

GRF_GPIO2D_IOMUX

Address: Operational Base + offset (0x0034)

GPIO2D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio2_d7_sel GPIO2D[7] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d4 2'b11: reserved
13:12	RW	0x0	gpio2_d6_sel GPIO2D[6] iomux select 2'b00: gpio 2'b01:reserved 2'b10: emmc_d3 2'b11: reserved
11:10	RW	0x0	gpio2_d5_sel GPIO2D[5] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d2 2'b11: reserved
9:8	RW	0x0	gpio2_d4_sel GPIO2D[4] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d1 2'b11: reserved
7:6	RO	0x0	reserved
5:4	RW	0x0	gpio2_d2_sel GPIO2D[2] iomux select 2'b00: gpio 2'b01: usb2otg_drvbus 2'b10: reserved 2'b11: reserved
3:2	RW	0x0	gpio2_d1_sel GPIO2D[1] iomux select 2'b00: gpio 2'b01: i2c0_sda 2'b10: fephyled_rxm1 2'b11: fephyled_txm1
1:0	RW	0x0	gpio2_d0_sel GPIO2D[0] iomux select 2'b00: gpio 2'b01: i2c0_scl 2'b10: fephy_led_linkm1 2'b11: reserved

GRF_GPIO3AL_IOMUX

Address: Operational Base + offset (0x0038)
GPIO3AL iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14:12	RW	0x0	gpio3_a4_sel GPIO3A[4] iomux select 3'b000: gpio 3'b001: tsp_d0 3'b010: cif_data0 3'b011: sdmmc0ext_d0 3'b100: uart1_tx 3'b101: usb3phy_debug4 3'b110: reserved 3'b111: reserved
11:9	RO	0x0	reserved
8:6	RW	0x0	gpio3_a2_sel GPIO3A[2] iomux select 3'b000: gpio 3'b001: tsp_clk 3'b010: cif_clkin 3'b011: sdmmc0ext_clkout 3'b100: spi_rxdm2 3'b101: usb3phy_debug3 3'b110: i2s2_sdim1 3'b111: reserved
5:3	RW	0x0	gpio3_a1_sel GPIO3A[1] iomux select 3'b000: gpio 3'b001: tsp_fail 3'b010: cif_href 3'b011: sdmmc0ext_det 3'b100: spi_txdm2 3'b101: usb3phy_debug2 3'b110: i2s2_sdom1 3'b111: reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio3_a0_sel GPIO3A[0] iomux select 3'b000: gpio 3'b001: tsp_valid 3'b010: cif_vsync 3'b011: sdmmc0ext_cmd 3'b100: spi_clkm2 3'b101: usb3phy_debug1 3'b110: i2s2_sclkm1 3'b111: reserved

GRF_GPIO3AH_IOMUX

Address: Operational Base + offset (0x003c)

GPIO3AH iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:9	RO	0x0	reserved
8:6	RW	0x0	gpio3_a7_sel GPIO3A[7] iomux select 3'b000: gpio 3'b001: tsp_d3 3'b010: cif_data3 3'b011: sdmmc0ext_d3 3'b100: uart1_ctsn 3'b101: usb3phy_debug7 3'b110: reserved 3'b111: reserved
5:3	RW	0x0	gpio3_a6_sel GPIO3A[6] iomux select 3'b000: gpio 3'b001: tsp_d2 3'b010: cif_data2 3'b011: sdmmc0ext_d2 3'b100: uart1_rx 3'b101: usb3phy_debug6 3'b110: reserved 3'b111: reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	gpio3_a5_sel GPIO3A[5] iomux select 3'b000: gpio 3'b001: tsp_d1 3'b010: cif_data1 3'b011: sdmmc0ext_d1 3'b100: uart1_rtsn 3'b101: usb3phy_debug5 3'b110: reserved 3'b111: reserved

GRF_GPIO3BL_IOMUX

Address: Operational Base + offset (0x0040)

GPIO3BL iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:3	RO	0x0	reserved
2:0	RW	0x0	gpio3_b0_sel GPIO3B[0] iomux select 3'b000: gpio 3'b001: tsp_d4 3'b010: cif_data4 3'b011: spi_csn0m2 3'b100: i2s2_lrcktxm1 3'b101: usb3phy_debug8 3'b110: i2s2_lrckrxm1 3'b111: reserved

GRF_GPIO3BH_IOMUX

Address: Operational Base + offset (0x0044)

GPIO3BH iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_GPIO3C_IOMUX

Address: Operational Base + offset (0x0048)

GPIO3C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:12	RW	0x0	gpio3_c6_sel GPIO3C[6] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_pwren 2'b11: reserved
11:10	RW	0x0	gpio3_c5_sel GPIO3C[5] iomux select 2'b00: gpio 2'b01:reserved 2'b10: emmc_clkout 2'b11: reserved
9:8	RO	0x0	reserved
7:6	RW	0x0	gpio3_c3_sel GPIO3C[3] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_cmd 2'b11: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio3_c2_sel GPIO3C[2] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d7 2'b11: reserved
3:2	RW	0x0	gpio3_c1_sel GPIO3C[1] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d6 2'b11: reserved
1:0	RW	0x0	gpio3_c0_sel GPIO3C[0] iomux select 2'b00: gpio 2'b01: reserved 2'b10: emmc_d5 2'b11: reserved

GRF_GPIO3D_IOMUX

Address: Operational Base + offset (0x004c)

GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0	reserved

GRF_COM_IOMUX

Address: Operational Base + offset (0x0050)

GRF common iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	grf_clk_out_gmacm1_sel gmac m1 io select 0:before optimization 1:after optimization
11	RW	0x0	grf_clk_out_gmacm2_sel clk_out_gmacm2 select 0:before optimization 1:after optimization
10	RW	0x0	grf_gmac_m1_sel gmac m1 io select 0:before optimization 1:after optimization
9	RW	0x0	grf_cif_io_sel cif_io select 0: m0 mux solution 1: m1 mux solution
8	RW	0x0	grf_tsp_io_sel tsp_io select 0: m0 mux solution 1: m1 mux solution
7	RW	0x0	grf_card_io_sel card_io select 0: m0 mux solution 1: m1 mux solution
6	RW	0x0	grf_i2s2_io_sel i2s2_io select 0: m0 mux solution 1: m1 mux solution
5:4	RW	0x0	grf_con_spi_io_sel spi_io_sel bit control 2'b00: m0 mux solution 2'b01: m1 mux solution 2'b10: m2 mux solution 2'b11: reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	grf_con_pdm_iomux_sel pdm_iomux_sel bit control 0: m0 mux solution 1: m1 mux solution
2	RW	0x0	grf_con_gmac_iomux_sel gmac_iomux_sel bit control 0: m0 mux solution 1: m1 mux solution
1:0	RW	0x0	grf_uart_dbg_sel grf_con_iomux_uartdbgsel when grf_con_iomux_uartdbgena is 1, uartdbg source select 2'b00: m0 2'b01: m1 2'b10: usb2phy 2'b11: reserved

GRF_GPIO0A_P
Address: Operational Base + offset (0x0100)
GPIO0A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x566a	gpio0_a_p gpio0a bit control GPIO0A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO0D_P
Address: Operational Base + offset (0x010c)
GPIO0D PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaaaa	gpio0_d_p gpio0d bit control GPIO0D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO1A_P

Address: Operational Base + offset (0x0110)

GPIO1A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xa555	gpio1_a_p gpio1a bit control GPIO1A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO1B_P

Address: Operational Base + offset (0x0114)

GPIO1B PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x56a5	gpio1_b_p gpio1b bit control GPIO1B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO1C_P

Address: Operational Base + offset (0x0118)

GPIO1C PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x9a65	gpio1_c_p gpio1c bit control GPIO1C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO1D_P

Address: Operational Base + offset (0x011c)

GPIO1D PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaaaa	gpio1_d_p gpio1d bit control GPIO1D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO2A_P

Address: Operational Base + offset (0x0120)

GPIO2A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x9556	gpio2_a_p gpio2a bit control GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO2B_P

Address: Operational Base + offset (0x0124)

GPIO2B PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x959a	gpio2_b_p gpio2b bit control GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO2C_P

Address: Operational Base + offset (0x0128)

GPIO2C PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5565	gpio2_c_p gpio2c bit control GPIO2C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO2D_P

Address: Operational Base + offset (0x012c)

GPIO2D PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x55a5	gpio2_d_p gpio2d bit control GPIO2D PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO3A_P

Address: Operational Base + offset (0x0130)

GPIO3A PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x55a5	gpio3_a_p gpio3a bit control GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO3B_P

Address: Operational Base + offset (0x0134)

GPIO3B PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5aaa	gpio3_b_p gpio3b bit control GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO3C_P

Address: Operational Base + offset (0x0138)

GPIO3C PU/PD control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x6555	gpio3_c_p gpio3c bit control GPIO3C PU/PD programmation section, every GPIO bit corresponding to 2bits 2'b00: Z(Normal operation); 2'b01: weak 1(pull-up); 2'b10: weak 0(pull_down); 2'b11: Repeater(Bus keeper)

GRF_GPIO0A_E

Address: Operational Base + offset (0x0200)

GPIO0A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x8011	gpio0_a_e gpio0a bit control GPIO0A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO0D_E

Address: Operational Base + offset (0x020c)

GPIO0D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x005a	gpio0_d_e gpio0d bit control GPIO0D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1A_E

Address: Operational Base + offset (0x0210)

GPIO1A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaaaa	gpio1_a_e gpio1a bit control GPIO1A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1B_E

Address: Operational Base + offset (0x0214)

GPIO1B drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaa2a	gpio1_b_e gpio1b bit control GPIO1B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1C_E

Address: Operational Base + offset (0x0218)

GPIO1C drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xa88a	gpio1_c_e gpio1c bit control GPIO1C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1D_E

Address: Operational Base + offset (0x021c)

GPIO1D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x005a	gpio1_d_e gpio1d bit control GPIO1D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2A_E

Address: Operational Base + offset (0x0220)

GPIO2A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio2_a_e gpio2a bit control GPIO2A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2B_E

Address: Operational Base + offset (0x0224)

GPIO2B drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x4145	gpio2_b_e gpio2b bit control GPIO2B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2C_E

Address: Operational Base + offset (0x0228)

GPIO2C drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5515	gpio2_c_e gpio2c bit control GPIO2C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2D_E

Address: Operational Base + offset (0x022c)

GPIO2D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaa01	gpio2_d_e gpio2d bit control GPIO2D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3A_E

Address: Operational Base + offset (0x0230)

GPIO3A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaa22	gpio3_a_e gpio3a bit control GPIO3A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3B_E
Address: Operational Base + offset (0x0234)
GPIO3B drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	gpio3_b_e gpio3b bit control GPIO3B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3C_E
Address: Operational Base + offset (0x0238)
GPIO3C drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaaaa	gpio3_c_e gpio3c bit control GPIO3C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO0L_SR
Address: Operational Base + offset (0x0300)
GPIO0 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio0b_sr GPIO0B slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio0a_sr GPIO0A slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO0H_SR
Address: Operational Base + offset (0x0304)
GPIO0 C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio0d_sr GPIO0D slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio0c_sr GPIO0C slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1L_SR

Address: Operational Base + offset (0x0308)

GPIO1 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio1b_sr GPIO1B slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio1a_sr GPIO1A slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO1H_SR

Address: Operational Base + offset (0x030c)

GPIO1 C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio1d_sr GPIO1D slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio1c_sr GPIO1C slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO2L_SR

Address: Operational Base + offset (0x0310)

GPIO2 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio2b_sr GPIO2B slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio2a_sr GPIO2A slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO2H_SR

Address: Operational Base + offset (0x0314)

GPIO2 C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio2d_sr GPIO2D slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio2c_sr GPIO2C slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3L_SR

Address: Operational Base + offset (0x0318)

GPIO3 A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio3b_sr GPIO3B slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio3a_sr GPIO3A slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO3H_SR

Address: Operational Base + offset (0x031c)

GPIO3 C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio3d_sr GPIO3D slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast
7:0	RW	0x00	grf_gpio3c_sr GPIO3C slew rate control for each bit 1'b0: slow(half frequency) 1'b1: fast

GRF_GPIO0L_SMT

Address: Operational Base + offset (0x0380)

GPIO0 A/B smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio0b_smt gpio0b_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio0a_smt gpio0a_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO0H_SMT

Address: Operational Base + offset (0x0384)

GPIO0 C/D smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio0d_smt gpio0d_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio0c_smt gpio0c_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO1L_SMT

Address: Operational Base + offset (0x0388)

GPIO1 A/B smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio1b_smt gpio1b_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio1a_smt gpio1a_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO1H_SMT

Address: Operational Base + offset (0x038c)

GPIO1 C/D smitter   control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio1d_smt gpio1d_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio1c_smt gpio1c_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO2L_SMT

Address: Operational Base + offset (0x0390)

GPIO2 A/B smitter   control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio2b_smt gpio2b_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

Bit	Attr	Reset Value	Description
7:0	RW	0x00	grf_gpio2a_smt gpio2a_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO2H_SMT

Address: Operational Base + offset (0x0394)

GPIO2 C/D smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio2d_smt gpio2d_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio2c_smt gpio2c_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO3L_SMT

Address: Operational Base + offset (0x0398)

GPIO3 A/B smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	grf_gpio3b_smt gpio3b_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio3a_smt gpio3a_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_GPIO3H_SMT

Address: Operational Base + offset (0x039c)

GPIO3 C/D smitter control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	grf_gpio3d_smt gpio3d_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.
7:0	RW	0x00	grf_gpio3c_smt gpio3c_smt bit control Schmitt trigger control. 0: No hysteresis 1: Schmitt trigger enabled.

GRF_SOC_CON0

Address: Operational Base + offset (0x0400)

SOC control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RO	0x0000	soc_con0 Reserved reserved

GRF_SOC_CON1
Address: Operational Base + offset (0x0404)
SOC control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	soc_con1 Reserved reserved

GRF_SOC_CON2
Address: Operational Base + offset (0x0408)
SOC control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	grf_con_i2s1_src_sel i2s1_src_sel bit control 1'b1 I2S1 is controll is connected with ACODEC PHY; 1'b0: I2S1 is connected with IO
14	RW	0x0	grf_con_i2s_acodec_en i2s_acodec_en bit control i2s_8ch iomux control 1:connect with acodec 0:connect with external io
13	RW	0x0	grf_con_ddrphy_bufferen_sel ddrphy_bufferen_sel bit control 1'b1: ddrphy_bufferen from grf_con_ddrphy_bufferen_core; 1'b0: ddrphy_bufferen from pmu
12	RW	0x1	grf_con_ddrphy_bufferen_core ddrphy_bufferen_core bit control 1'b1: enable ddrphy_bufferen; 1'b0: disable ddrphy_bufferen
11	RW	0x0	grf_con_hdmi_sdain_msk hdmi_sdain_msk bit control hdmi_sdain mask control 1: mask disable 0: mask enable
10	RW	0x0	grf_con_hdmi_sclin_msk hdmi_sclin_msk bit control hdmi_sclin mask control 1: mask disable 0: mask enable
9	RW	0x0	grf_con_hdmi_cecin_msk hdmi_cecin_msk bit control hdmi_cecin mask control 0: mask disable 1: mask enable
8	RW	0x0	grf_con_saradc_sel saradc_sel bit control SARADC controller selection 1'b1: select saradc auto controller 1'b0: select orignal saradc controller
7	RW	0x0	grf_con_hdmisda5v_gpio_iout hdmisda5v_gpio_iout bit control IO PAD output data 1'b0: set IO output to 0; 1'b1: set IO output to 1;

Bit	Attr	Reset Value	Description
6	RW	0x0	grf_con_hdmisda5v_gpio_ioe_ hdmisda5v_gpio_ioe_ bit control IO Pad output enable bit control 1'b1: set IO as input; 1'b0: set IO as output;
5	RW	0x0	grf_con_hdmiscl5v_gpio_iout hdmiscl5v_gpio_iout bit control IO PAD output data 1'b0: set IO output to 0; 1'b1: set IO output to 1;
4	RW	0x0	grf_con_hdmiscl5v_gpio_ioe_ hdmiscl5v_gpio_ioe_ bit control IO Pad output enable bit control 1'b1: set IO as input; 1'b0: set IO as output;
3	RW	0x0	grf_con_hdmihpd5v_gpio_iout hdmihpd5v_gpio_iout bit control IO PAD output data 1'b0: set IO output to 0; 1'b1: set IO output to 1;
2	RW	0x0	grf_con_hdmihpd5v_gpio_ioe_ hdmihpd5v_gpio_ioe_ bit control IO Pad output enable bit control 1'b1: set IO as input; 1'b0: set IO as output;
1	RW	0x0	grf_con_hdmicec5v_gpio_iout hdmicec5v_gpio_iout bit control IO PAD output data 1'b0: set IO output to 0; 1'b1: set IO output to 1;
0	RW	0x0	grf_con_hdmicec5v_gpio_ioe_ hdmicec5v_gpio_ioe_ bit control IO Pad output enable bit control 1'b1: set IO as input; 1'b0: set IO as output;

GRF_SOC_CON3

Address: Operational Base + offset (0x040c)

SOC control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_con_hdmisda5v_gpio_sel hdmisda5v_gpio_sel bit control "if grf_con_hdmisda5v_gpio_sel == 0 and grf_con_i2c3_sda5v == 0, SDA5V is controlled by HDMI controller; if grf_con_hdmisda5v_gpio_sel == 0 and grf_con_i2c3_sda5v == 1, SDA5V is controlled by I2C3 controller; if grf_con_hdmisda5v_gpio_sel == 1 and no matter grf_con_i2c3_sda5v what is, SDA5V is controlled by grf; "
14	RW	0x0	grf_con_hdmiscl5v_gpio_sel hdmiscl5v_gpio_sel bit control "if grf_con_hdmiscl5v_gpio_sel == 0 and grf_con_i2c3_scl5v == 0, SCL5V is controlled by HDMI controller; if grf_con_hdmiscl5v_gpio_sel == 0 and grf_con_i2c3_scl5v == 1, SCL5V is controlled by I2C3 controller; if grf_con_hdmiscl5v_gpio_sel == 1 and no matter grf_con_i2c3_scl5v what is, SCL5V is controlled by grf; "
13	RW	0x0	grf_con_hdmihpd5v_gpio_sel hdmihpd5v_gpio_sel bit control 1'b1: HPD5V io is controlled by grf_con_hdmihpd5v_gpio_ioe_ and grf_con_hdmihpd5v_gpio_iout; 1'b0: HPD5V is controlled by HDMI controller
12	RW	0x0	grf_con_hdmicec5v_gpio_sel hdmicec5v_gpio_sel bit control 1'b1: CEC5V io is controlled by grf_con_hdmicec5v_gpio_ioe_ and grf_con_hdmicec5v_gpio_iout; 1'b0: CEC5V is controlled by HDMI controller
11	RW	0x0	grf_con_h265enc_work_flag h265enc_work_flag bit control 1'b1: sram is controlled by h265 encoder 1'b0: sram is controlled by h264 encoder

Bit	Attr	Reset Value	Description
10:9	RW	0x0	grf_vop_standby_sel vop_standby select dcf vop standby source 2'b00: from vop standby; 2'b01: from vop aclk en; 2'b10: from vop aclk en or vop standby; 2'b11 Reserved
8	RW	0x0	grf_hdmiphy_pll_pd hdmiphy_pll_pd hdmiphy pll power down, active high
7	RW	0x0	grf_hdmip_pdata_en hdmip_pdata enable hdmiphy input parallel data enable 1:enable 0:disable
6	RO	0x0	reserved
5:3	RW	0x0	grf_uart_rts_sel uart_rts select UART polarity selection for rts_n Every bit for one UART, bit2 is for UART2, bit1 is for UART1, bit0 is for UART0 1:cts_n is high active 0:cts_n is low active
2:0	RW	0x0	grf_uart_cts_sel uart_cts select UART polarity selection for cts_n Every bit for one UART, bit2 is for UART2, bit1 is for UART1, bit0 is for UART0 1:cts_n is high active 0:cts_n is low active

GRF_SOC_CON4
Address: Operational Base + offset (0x0410)
SOC control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15	RW	0x0	cif_pclkin_inv_sel pclkin_inv select 0: inveter disable 1: inverter enable
14	RW	0x0	grf_con_gmac2io_mac_clk_output_en 0: output 1: input
13	RW	0x0	grf_con_hdmi_hpd_src_sel hdmi_hpd source select 0:from gpio of 3.3V or 5V 1:from SARADC CH0
12	RW	0x0	grf_force_jtag force jtag Force select jtag function from sdmmc0 IO 1:IO used for JTAG. 0:IO used for SDMMC
11	RW	0x0	grf_hdmi_cec_vsel grf_hdmi_cec_vsel hdmi cec port 3.3V/5V io select 0:IO is 3.3V 1:IO is 5V
10	RW	0x0	grf_hdmi_sda_vsel grf_hdmi_sda_vsel hdmi sda port 3.3V/5V io select 0:IO is 3.3V 1:IO is 5V
9	RW	0x0	grf_hdmi_scl_vsel grf_hdmi_scl_vsel hdmi scl port 3.3V/5V io select 0:IO is 3.3V 1:IO is 5V
8	RW	0x0	grf_hdmi_hdp_vsel grf_hdmi_hdp_vsel hdmi hpd port 3.3V/5V io select 0:IO is 3.3V 1:IO is 5V
7	RW	0x0	grf_vccio2_vsel_src grf_vccio2_vsel_src 1'b1: vccio2 vsel controlled by grf_vccio2_vsel; 1'b0: vccio2 vsel controlled by GPIO2B4 IO
6	RW	0x0	grf_pmuio_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V

Bit	Attr	Reset Value	Description
5	RW	0x0	grf_vccio6_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V
4	RW	0x0	grf_vccio5_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V
3	RW	0x0	grf_vccio4_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V
2	RW	0x0	grf_vccio3_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V
1	RW	0x0	grf_vccio2_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V
0	RW	0x0	grf_vccio1_vsel VCC IO voltage select 1'b0:3.3V 1'b1:1.8V

GRF_SOC_CON5

Address: Operational Base + offset (0x0414)

SOC control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	vpu_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response

Bit	Attr	Reset Value	Description
14	RW	0x0	vop_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
13	RW	0x0	usb_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
12	RW	0x0	subvio_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
11	RW	0x0	rkvenc_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
10	RW	0x0	rkvdec_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
9	RW	0x0	vpu_pwr_IdleReq send idle request to vpu niu 0:disable 1:enable
8	RW	0x0	vio_pwr_IdleReq send idle request to vio niu 0:disable 1:enable
7	RW	0x0	sys_pwr_IdleReq send idle request to bus niu 0:disable 1:enable
6	RW	0x0	rkvenc_pwr_IdleReq rkvenc_pwr_IdleReq
5	RW	0x0	rkvdec_pwr_IdleReq send idle request to rkvdec niu 0:disable 1:enable
4	RW	0x0	peri_pwr_IdleReq send idle request to peri niu 0:disable 1:enable

Bit	Attr	Reset Value	Description
3	RW	0x0	msch_pwr_IdleReq send idle request to msch niu 0:disable 1:enable
2	RW	0x0	msch_apb_pwr_IdleReq send idle request to mschapb niu 0:disable 1:enable
1	RW	0x0	gpu_pwr_IdleReq send idle request to gpu niu 0:disable 1:enable
0	RW	0x0	core_pwr_IdleReq send idle request to core niu 0:disable 1:enable

GRF_SOC_CON6

Address: Operational Base + offset (0x0418)

SOC control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	peri_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
14	RW	0x0	nv_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
13	RW	0x0	msch_srv_fw_fwr_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response

Bit	Attr	Reset Value	Description
12	RW	0x0	msch_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
11	RW	0x0	gpu_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
10	RW	0x0	gmac_fwr_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
9	RW	0x0	core_fwr_bus_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
8	RW	0x0	vcodec_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
7	RW	0x0	gpu_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
6	RW	0x0	core_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
5	RW	0x0	bus_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
4	RW	0x0	vop_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
3	RW	0x0	vio_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
2	RW	0x0	rkvenc_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response

Bit	Attr	Reset Value	Description
1	RW	0x0	rkvdec_req_link_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response
0	RW	0x0	peri_req_pwrDiscTargPwrStall Response type when NIU is set Idle 0:error response 1:stall response

GRF_SOC_CON7
Address: Operational Base + offset (0x041c)
SOC control register7

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	grf_con_otp_usr_clk_mux otp user mode clock source mux 0: bypass clock 1: divide by 2
1	RW	0x0	grf_con_newpll_clamp_en newpll clamp enable 0: disable 1: enable
0	RW	0x0	grf_con_scr_sim_detect_inv_sel scr detect inveter select 0: inveter disable 1: inveter enable

GRF_SOC_CON8
Address: Operational Base + offset (0x0420)
SOC control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	grf_tsadc_testbit_h tsadc_testbit_h bit register tsadc_testbit_h bit register

GRF_SOC_CON9

Address: Operational Base + offset (0x0424)

SOC control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	grf_tsadc_testbit_l tsadc_testbit_l bit register tsadc_testbit_l bit register

GRF_SOC_CON10

Address: Operational Base + offset (0x0428)

SOC control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	grf_con_hdmi_sda5v_smt hdmi_sda5v_smt bit control hdmi_sda5v_smt bit control
14	RW	0x1	grf_con_hdmi_scl5v_smt hdmi_scl5v_smt bit control hdmi_scl5v_smt bit control
13	RW	0x1	grf_con_hdmi_hpd5v_smt hdmi_hpd5v_smt bit control hdmi_hpd5v_smt bit control
12	RW	0x1	grf_con_hdmi_cec5v_smt hdmi_cec5v_smt bit control hdmi_cec5v_smt bit control

Bit	Attr	Reset Value	Description
11	RW	0x1	grf_con_gpiomut_pmuio_p2 gpiomut_pmuio_p2 bit control gpiomut_pmuio pull bit 2
10	RW	0x0	grf_con_gpiomut_pmuio_p1 gpiomut_pmuio_p1 bit control gpiomut_pmuio pull bit 1
9	RW	0x0	grf_con_sdmmc_pwren_sel iomux select GPIO2A7 sdmmc power selection 1'b0: from sdmmc_ext 1'b1: from sdmmc0
8	RW	0x0	grf_con_i2c3_sda5v iomux select "if grf_con_hdmisda5v_gpio_sel == 0 and grf_con_i2c3_sda5v == 0, SDA5V is controlled by HDMI controller; if grf_con_hdmisda5v_gpio_sel == 0 and grf_con_i2c3_sda5v == 1, SDA5V is controlled by I2C3 controller; if grf_con_hdmisda5v_gpio_sel == 1 and no matter grf_con_i2c3_sda5v what is, SDA5V is controlled by grf; "
7	RW	0x0	grf_con_i2c3_scl5v iomux select "if grf_con_hdmiscl5v_gpio_sel == 0 and grf_con_i2c3_scl5v == 0, SCL5V is controlled by HDMI controller; if grf_con_hdmiscl5v_gpio_sel == 0 and grf_con_i2c3_scl5v == 1, SCL5V is controlled by I2C3 controller; if grf_con_hdmiscl5v_gpio_sel == 1 and no matter grf_con_i2c3_scl5v what is, SCL5V is controlled by grf; "
6	RW	0x0	grf_con_tsadc_ch_inv tsadc_ch_inv bit control The enable signal of the clock inverter for the analog to digital interface 0:invert 1:don't invert
5	RW	0x0	5 RW 0x0 grf_con_clk_wifi_sel clk_wifi_sel bit control clk_wifi (GPIO1D3/GPIO0A0) source selection 1'b0: from clk_wifi; 1'b1: from 24M OSC
4	RW	0x0	grf_con_i2s1_8ch_sdio3_oen i2s1_8ch_sdio3_oen bit control i2s1_8ch_sdio3_oen 1:output disable 2:output enable

Bit	Attr	Reset Value	Description
3	RW	0x0	grf_con_i2s1_8ch_sdio2_oen i2s1_8ch_sdio2_oen bit control i2s1_8ch_sdio2_oen 1:output disable 1:output enable
2	RW	0x0	grf_con_i2s1_8ch_sdio1_oen i2s1_8ch_sdio1_oen bit control i2s1_8ch_sdio1_oen 1:output disable 0:output enable
1	RW	0x0	gpiomut_pmuio_iout gpiomut_pmuio_iout bit register gpiomut output value 1'b1: output 1; 1'b1: output 0
0	RW	0x0	gpiomut_pmuio_ioe_ gpiomut_pmuio_ioe_ bit register gpiomut output enable 1'b1: output disable; 1'b0: output enable

GRF_SOC_STATUS0
Address: Operational Base + offset (0x0480)
SOC status register0

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RO	0x0	h265enc_vpu_idle h265enc_vpu_idle bit register
26	RO	0x0	hdmicec5v_gpio_masked_pin hdmicec5v_gpio_masked_pin IO PAD input status
25	RO	0x0	hdmihpd5v_gpio_masked_pin hdmihpd5v_gpio_masked_pin IO PAD input status
24	RO	0x0	hdmisda5v_gpio_masked_pin IO PAD input status
23	RO	0x0	hdmiscl5v_gpio_masked_pin IO PAD input status
22	RO	0x0	gpiomut_pmuio_pin IO PAD input status
21	RO	0x0	grf_st_acodec_master_en st_acodec_master enable st_acodec_master enable

Bit	Attr	Reset Value	Description
20	RO	0x0	gmac2phy_portselect gmac2phy_port select signal indicating the default PHY interface of MAC 1:MII 0:GMII
19	RO	0x0	grf_stat_vdac_dispdet grf_stat_vdac_dispdet bit register vdac cable detection output status
18	RO	0x0	vop_dma_finish vop_dma_finish bit register vop_dma_finish_status
17	RO	0x0	reserved
16	RO	0x0	timer_en_status5 timer_en_status5 bit register
15	RO	0x0	timer_en_status4 timer_en_status4 bit register
14	RO	0x0	timer_en_status3 timer_en_status3 bit register
13	RO	0x0	timer_en_status2 timer_en_status2 bit register
12	RO	0x0	timer_en_status1 timer_en_status1 bit register
11	RO	0x0	timer_en_status0 timer_en_status0 bit register
10	RO	0x0	gmac2io_portselect gmac2io_port select
9	RO	0x0	opt_sbpi_busy_ns opt_sbpi_busy_ns bit register
8	RO	0x0	opt_user_busy_ns opt_user_busy_ns bit register
7	RO	0x0	opt_sbpi_busy_s opt_sbpi_busy_s bit register
6	RO	0x0	opt_user_busy_s opt_user_busy_s bit register
5	RO	0x0	ddr_plllock ddr_plllock bit register DDRPLL of DDRPHY lock status.
4	RO	0x0	apll_lock pll_lock bit register APLL lock status.
3	RO	0x0	dpll_lock pll_lock bit register DPLL lock status.

Bit	Attr	Reset Value	Description
2	RO	0x0	cpll_lock pll_lock bit register CPLL lock status.
1	RO	0x0	gpll_lock pll_lock bit register GPLL lock status.
0	WO	0x0	npll_lock pll_lock bit register NPLL lock status

GRF_SOC_STATUS1
Address: Operational Base + offset (0x0484)
SOC status register1

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RO	0x0	vpu_pwr_Idle vpu_pwr_Idle bit register idle status of vpu niu 0: idle is asserted 1: idle is de-asserted
18	RO	0x0	vio_pwr_Idle vio_pwr_Idle bit register idle status of vio niu 0: idle is asserted 1: idle is de-asserted
17	RO	0x0	sys_pwr_Idle sys_pwr_Idle bit register idle status of bus niu 0: idle is asserted 1: idle is de-asserted
16	RO	0x0	rkvenc_pwr_Idle rkvenc_pwr_Idle bit register idle status of rkvdec niu 0: idle is asserted 1: idle is de-asserted
15	RO	0x0	rkvdec_pwr_Idle rkvdec_pwr_Idle bit register rkvdec_pwr_Idle bit register 0: idle is asserted 1: idle is de-asserted

Bit	Attr	Reset Value	Description
14	RO	0x0	peri_pwr_Idle peri_pwr_Idle bit register idle status of peri niu 0: idle is asserted 1: idle is de-asserted
13	RO	0x0	msch_pwr_Idle msch_pwr_Idle bit register idle status of msch niu 0: idle is asserted 1: idle is de-asserted
12	RO	0x0	msch_apb_pwr_Idle msch_apb_pwr_Idle bit register idle status of mschapb niu 0: idle is asserted 1: idle is de-asserted
11	RO	0x0	gpu_pwr_Idle gpu_pwr_Idle bit register idle status of gpu niu 0: idle is asserted 1: idle is de-asserted
10	RO	0x0	core_pwr_Idle core_pwr_Idle bit register idle status of core niu 0: idle is asserted 1: idle is de-asserted
9	RO	0x0	vpu_pwr_IdleAck vpu_pwr_IdleAck bit register idle acknowledge status from bus vpu 0: idle_ack asserted 1: idle_ack de-asserted
8	RO	0x0	vio_pwr_IdleAck vio_pwr_IdleAck bit register idle acknowledge status from bus vio 0: idle_ack asserted 1: idle_ack de-asserted
7	RO	0x0	sys_pwr_IdleAck sys_pwr_IdleAck bit register idle acknowledge status from bus niu 0: idle_ack asserted 1: idle_ack de-asserted
6	RO	0x0	rkvenc_pwr_IdleAck rkvenc_pwr_IdleAck bit register rkvenc_pwr_IdleAck bit register 0: idle_ack asserted 1: idle_ack de-asserted

Bit	Attr	Reset Value	Description
5	RO	0x0	rkvdec_pwr_IdleAck rkvdec_pwr_IdleAck bit register idle acknowledge status from rkvdec niu 0: idle_ack asserted 1: idle_ack de-asserted
4	RO	0x0	peri_pwr_IdleAck peri_pwr_IdleAck bit register idle acknowledge status from peri niu 0: idle_ack asserted 1: idle_ack de-asserted
3	RO	0x0	msch_pwr_IdleAck msch_pwr_IdleAck bit register idle acknowledge status from msch niu 0: idle_ack asserted 1: idle_ack de-asserted
2	RO	0x0	msch_apb_pwr_IdleAck msch_apb_pwr_IdleAck bit register idle acknowledge status from mschapb niu 0: idle_ack asserted 1: idle_ack de-asserted
1	RO	0x0	gpu_pwr_IdleAck gpu_pwr_IdleAck bit register idle acknowledge status from gpu niu 0: idle_ack asserted 1: idle_ack de-asserted
0	RO	0x0	core_pwr_IdleAck core_pwr_IdleAck bit register idle acknowledge status from core niu 0: idle_ack asserted 1: idle_ack de-asserted

GRF_SOC_STATUS2

Address: Operational Base + offset (0x0488)

SOC status register2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	grf_sta_usb3otg_logic_analyzer_trace[31:0] usb3otg_logic_analyzer_trace[31:0] bit status

GRF_SOC_STATUS3

Address: Operational Base + offset (0x048c)

SOC status register3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	grf_sta_usb3otg_logic_analyzer_trace[63:32] usb3otg_logic_analyzer_trace[63:32] bit status

GRF_SOC_STATUS4

Address: Operational Base + offset (0x0490)

SOC status register4

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	grf_sta_usb3otg_host_current_belt[11:0] usb3otg_host_current_belt[11:0] bit status

GRF_USB3OTG_CON0

Address: Operational Base + offset (0x04c0)

USB3OTG control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_con_usb3otg_host_u2_port_disable USB2.0 Port Disable control. 0: Port Enabled 1: Port Disabled When 1, this signal stops reporting connect/disconnect events the port and keeps the port in disabled state.
14	RW	0x0	grf_con_usb3otg_host_port_power_control_present This indicates whether the host controller implementation includes port power control. 0: Indicates that the port does not have port power switches. 1: Indicates that the port has port power switches
13:8	RW	0x20	grf_con_usb3otg_fladj_30mhz_reg usb3otg_fladj_30mhz_reg bit control
7:6	RW	0x0	grf_con_usb3otg_hub_port_perm_attach Indicates if the device attached to a downstream port is permanently attached or not. 0: Not permanently attached 1: Permanently attached Bit0 is for USB2.0 port and bit1 are for USB 3.0 SS port.

Bit	Attr	Reset Value	Description
5:4	RW	0x0	grf_con_usb3otg_hub_port_overcurrent This is the per port Overcurrent indication of the root-hub ports: 0: No Overcurrent 1: Overcurrent Bit0 is for USB 2.0 port and bit1 are for USB 3.0 SS port.
3:0	RW	0x0	grf_con_usb3otg_bus_filter_bypass 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. The function of each bit is: bus_filter_bypass[3]: Bypass the filter for utmiotg_iddig bus_filter_bypass[2]: Bypass the filters for utmisrp_bvalid and utmisrp_sessend bus_filter_bypass[1]: Bypass the filter for pipe3_PowerPresent all U3 ports 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 (bypassed)

GRF_USB3OTG_CON1

Address: Operational Base + offset (0x04c4)

USB3OTG control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x1	grf_con_usb3otg_host_num_u3_port usb3otg_host_num_u3_port bit control xHCI usb3 port number, default as 1.
11:8	RW	0x1	grf_con_usb3otg_host_num_u2_port usb3otg_host_num_u2_port bit control xHCI host USB2 Port number, default as 1.
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	grf_con_usb3otg_host_legacy_smi_bar usb3otg_host_legacy_smi_bar bit control 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	grf_con_usb3otg_host_legacy_smi_pci_cmd usb3otg_host_legacy_smi_pci_cmd bit control 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.
3:2	RO	0x0	reserved
1	RW	0x0	grf_con_usb3otg_pme_en usb3otg_pme_en bit control Enable signal for the pme_generation. Enable the core to assert pme_generation.
0	RW	0x0	grf_con_usb3otg_host_u3_port_disable USB 3.0 SS Port Disable control. 0: Port Enabled 1: Port Disabled

GRF_CPU_CON0
Address: Operational Base + offset (0x0500)
CPU control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	grf_con_cfgte cfgte bit control
11:8	RW	0x0	grf_con_cfgend cfgend bit control
7:5	RO	0x0	reserved
4	RW	0x0	grf_con_l2rstdisable l2rstdisable bit control
3:0	RW	0x0	grf_con_l1rstdisable l1rstdisable bit control

GRF_CPU_CON1

Address: Operational Base + offset (0x0504)

CPU control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:6	RO	0x0	reserved
5	RW	0x0	grf_con_evento_clear vento_clear bit control
4	RW	0x0	grf_con_eventi eventi bit control
3	RW	0x1	grf_con_dbgselfaddrv dbgselfaddrv bit control
2	RW	0x1	grf_con_dbgromaddrv dbgromaddrv bit control
1	RW	0x0	grf_con_cfgsdisable cfgsdisable bit control
0	RW	0x0	grf_con_clrexmonreq clrexmonreq bit control

GRF_CPU_STATUS0

Address: Operational Base + offset (0x0520)

CPU status register0

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	grf_st_l2flushdone l2flushdone bit status
11	RO	0x0	grf_st_clrexmonack clrexmonack bit status
10	RO	0x0	grf_st_jtagnsw jtagnsw bit status
9	RO	0x0	grf_st_jtagtop jtagtop bit status
8	RO	0x0	evento_rising_edge evento_rising_edge bit status
7:4	RO	0x0	power_state power_state bit status

Bit	Attr	Reset Value	Description
3:0	RO	0x0	grf_st_smpnamp smpnamp bit status

GRF_CPU_STATUS1
Address: Operational Base + offset (0x0524)
CPU status register1

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	grf_st_standbywfil2 standbywfil2 bit status
11:8	RO	0x0	cpu_state cpu state status
7:4	RO	0x0	grf_st_standbywfi standbywfi bit status
3:0	RO	0x0	grf_st_standbywfe standbywfe bit status

GRF_OS_REG0
Address: Operational Base + offset (0x05c8)
os register0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg0 Reserved reserved

GRF_OS_REG1
Address: Operational Base + offset (0x05cc)
os register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg1 Reserved reserved

GRF_OS_REG2
Address: Operational Base + offset (0x05d0)
os register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg2 Reserved reserved

GRF_OS_REG3
Address: Operational Base + offset (0x05d4)
os register3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg3 Reserved reserved

GRF_OS_REG4
Address: Operational Base + offset (0x05d8)
os register4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg4 Reserved reserved

GRF_OS_REG5
Address: Operational Base + offset (0x05dc)
os register5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg5 Reserved reserved

GRF_OS_REG6
Address: Operational Base + offset (0x05e0)
os register6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg6 Reserved reserved

GRF_OS_REG7
Address: Operational Base + offset (0x05e4)
os register7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg7 Reserved reserved

GRF_SIG_DETECT_CON

Address: Operational Base + offset (0x0680)

External signal detect configue register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	sdmmc_ext_detectn_neg_irq_en sdmmc_ext_detectn_neg_irq enable 1'b1: enable irq; 1'b0: disable irq.
2	RW	0x0	sdmmc_ext_detectn_pos_irq_en sdmmc_ext_detectn_pos_irq enable 1'b1: enable irq; 1'b0: disable irq.
1	RW	0x0	sdmmc_detectn_neg_irq_en sdmmc_detectn_neg_irq enable 1'b1: enable irq; 1'b0: disable irq.
0	RW	0x0	sdmmc_detectn_pos_irq_en sdmmc_detectn_pos_irq enable 1'b1: enable irq; 1'b0: disable irq.

GRF_SIG_DETECT_STATUS

Address: Operational Base + offset (0x0690)

External signal detect status register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	sdmmc_ext_detectn_neg_irq sdmmc_detectn_ext_neg irq status bit
2	RW	0x0	sdmmc_ext_detectn_pos_irq sdmmc_detectn_ext_pos irq status bit
1	RW	0x0	sdmmc_detectn_neg_irq sdmmc_detectn_neg irq status bit
0	RW	0x0	sdmmc_detectn_pos_irq sdmmc_detectn_pos irq status bit

GRF_SIG_DETECT_STATUS_CLEAR

Address: Operational Base + offset (0x06a0)

External signal detect status clear register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	WO	0x0	sdmmc_ext_detectn_neg_irq_clr sdmmc_ext_detectn_neg_irq clear bit

Bit	Attr	Reset Value	Description
2	RW	0x0	sdmmc_ext_detectn_pos_irq_clr sdmmc_ext_detectn_pos_irq clear bit
1	RW	0x0	sdmmc_detectn_neg_irq_clr sdmmc_detectn_neg_irq clear bit
0	RW	0x0	sdmmc_detectn_pos_irq_clr sdmmc_detectn_pos_irq clear bit

GRF_SDMMC_DET_COUNTER

Address: Operational Base + offset (0x06b0)

SDMMC detect counter register

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

GRF_HOST0_CON0

Address: Operational Base + offset (0x0700)

host0 control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:6	RW	0x20	grf_con_host0_fladj_val_common host0_fladj_val_common bit control
5:0	RW	0x20	grf_con_host0_fladj_val host0_fladj_val bit control

GRF_HOST0_CON1

Address: Operational Base + offset (0x0704)

host0 control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13	RW	0x0	grf_con_host0_arb_pause host0_arb_pause bit control
12	RW	0x0	grf_con_host0_ohci_susp_lgcy host0_ohci_susp_lgcy bit control
11	RW	0x0	grf_con_host0_ohci_cntsel host0_ohci_cntsel bit control
10	RW	0x1	grf_con_host0_ohci_clkcktrst host0_ohci_clkcktrst bit control
9	RW	0x0	grf_con_host0_app_prt_ovrcur host0_app_prt_ovrcur bit control
8	RW	0x0	grf_con_host0_autoppd_on_overcur_en host0_autoppd_on_overcur_en bit control
7	RW	0x1	grf_con_host0_word_if host0_word_if bit control
6	RW	0x0	grf_con_host0_sim_mode host0_sim_mode bit control
5	RW	0x1	grf_con_host0_incrx_en host0_incrx_en bit control
4	RW	0x1	grf_con_host0_incr8_en host0_incr8_en bit control
3	RW	0x1	grf_con_host0_incr4_en host0_incr4_en bit control
2	RW	0x1	grf_con_host0_incr16_en host0_incr16_en bit control
1	RW	0x0	grf_con_host0_hubsetup_min host0_hubsetup_min bit control
0	RW	0x0	grf_con_host0_app_start_clk host0_app_start_clk bit control

GRF_HOST0_CON2

Address: Operational Base + offset (0x0708)

host0 control register2

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

GRF_OTG_CON0

Address: Operational Base + offset (0x0880)

OTG control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:3	RO	0x0	reserved
2	RW	0x0	otg_dbnce_fltr_bypass otg_dbnce_fltr_bypass bit control
1:0	RW	0x0	otg_scaledown_mode otg_scaledown_mode bit control

GRF_HOST0_STATUS

Address: Operational Base + offset (0x0890)

HOST0 status register

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

Bit	Attr	Reset Value	Description
21	RO	0x0	grf_stat_host0_ohci_rmtwkp host0_ohci_rmtwkp bit status
20:17	RO	0x0	grf_stat_host0_ehci_lpsmc_state host0_ehci_lpsmc_state bit status
16:11	RO	0x00	grf_stat_host0_ehci_usbsts host0_ehci_usbsts bit status
10:0	RO	0x000	grf_stat_host0_ehci_xfer_cnt host0_ehci_xfer_cnt bit status

GRF_MAC_CON0

Address: Operational Base + offset (0x0900)

MAC control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13:7	RW	0x00	gmac2io_clk_rx_dl_cfg gmac2io_clk_rx_dl_cfg bit control
6:0	RO	0x00	gmac2io_clk_tx_dl_cfg gmac2io_clk_tx_dl_cfg bit control

GRF_MAC_CON1

Address: Operational Base + offset (0x0904)

MAC control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12:11	RW	0x0	gmac2io_gmii_clk_sel gmac2io_gmii_clk_sel bit control GMII clock selection 2'b00:125MHz 2'b11:25MHz 2'b10:2.5MHz
10	RW	0x0	gmac2io_rmii_extclk_sel gmac2io_rmii_extclk_sel bit control
9	RW	0x0	gmac2io_rmii_mode gmac2io_rmii_mode bit control RMII mode selection 2'b11:RMII mode 2'b00:MII mode 2'b01:reserved 2'b10:reserved
8	RO	0x0	reserved
7	RW	0x0	gmac2io_rmii_clk_sel gmac2io_rmii_clk_sel bit control RMII clock selection 1'b1:25MHz 1'b0:2.5MHz
6:4	RW	0x0	gmac2io_phy_intf_sel gmac2io_phy_intf_sel bit control PHY interface select 3'b001:RGMII 3'b100:RMII All others:Reserved
3	RW	0x0	gmac2io_flowctrl gmac2io_flowctrl bit control GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frame in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
2	RW	0x0	gmac2io_mac_speed gmac2io_mac_speed bit control MAC speed 1'b1:100-Mbps 1'b0:10-Mbps

Bit	Attr	Reset Value	Description
1	RW	0x0	gmac2io_rxclk_dly_ena gmac2io_rxclk_dly_ena bit control RGMII RX clock delayline enable 1'b1:enable 1'b0:disable
0	RW	0x0	gmac2io_txclk_dly_ena gmac2io_txclk_dly_ena bit control RGMII TX clock delayline enable 1'b1:enable 1'b0:disable

GRF_MAC_CON2

Address: Operational Base + offset (0x0908)

MAC control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11	RW	0x0	gmac2phy_use_inter_phy_txrx gmac2phy_use_inter_phy_txrx bit control gmac2phy_use_inter_phy_txrx bit control
10	RW	0x0	gmac2phy_rmii_extclk_sel gmac2phy_rmii_extclk_sel bit control gmac2phy_rmii_extclk_sel bit control
9	RW	0x0	gmac2phy_rmii_mode gmac2phy_rmii_mode bit control RMII mode selection 2'b11:RMII mode 2'b00:MII mode 2'b01:reserved 2'b10:reserved
8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	gmac2phy_rmii_clk_sel gmac2phy_rmii_clk_sel bit control PHY interface select 3'b001:RGMII 3'b100:RMII All others:Reserved
6:4	RW	0x0	gmac2phy_phy_intf_sel gmac2phy_phy_intf_sel bit control PHY interface select 3'b001:RGMII 3'b100:RMII All others:Reserved
3	RW	0x0	gmac2phy_flowctrl gmac2phy_flowctrl bit control GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frame in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
2	RW	0x0	gmac2phy_mac_speed gmac2phy_mac_speed bit control MAC speed 1'b1:100-Mbps 1'b0:10-Mbps
1:0	RO	0x0	reserved

GRF_MACPHY_CON0

Address: Operational Base + offset (0x0b00)

MACPHY control register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Reserved
15	RW	0x0	macphy_ref_clk_sel Tie to same level as macphy_clk_freq
14	RW	0x0	macphy_clk_freq 0: for 25 MHz clock input; 1: for 50 MHz clock input.
13	RW	0x1	macphy_automodix_en Enables auto-detection of MDI/MDIX mode. Refer to "cfg_mode"

Bit	Attr	Reset Value	Description
12	RW	0x0	macphy_en_high Defines polarity of output enable signals. ''0'' for active low output enable signal. "mdio_dir, rxdz,miiz,rxerz" signal polarity control.
11	RW	0x0	macphy_fx_mode Enables FX mode
10	RW	0x0	macphy_adc_bp Puts the ADC by default in bypass mode
9	RW	0x0	macphy_pll_bp Puts the PLL by default in bypass mode
8	RW	0x0	macphy_smii_soure_sync smii source sync register field. Only relevant for SMII mode
7:6	RW	0x0	macphy_mii_mode MII mode register field. ''00'' for MII mode, "01" for RMII mode, "10" for SMII, "11" reserved
5:3	RW	0x7	macphy_mode MODE register file. "000" - 10BaseT, Half Duplex, Auto negotiation disabled "001" - 10Base-T, Full Duplex, Auto negotiation disabled "010" - 100Base-TX, Half Duplex, Auto-negotiation disabled "011" - 100Base-TX, Full Duplex, Auto-negotiation disabled "100" - 100Base-Tx, Half Duplex, Auto-negotaition Enabled "101" - Repeater mode, 100Base-Tx, Half Duplex, Auto-negotiation Enabled "110" - Power down mode, In this mode phy wake up in power fown mode "111" - All capable, Full Duplex, 10 & 100 BT, Auto negotiation enabled, AutoMDIX enable
2	RW	0x0	macphy_powerup_reset Power Up Reset bit. Default value of powerup_reset bit 0 - Power up reset disabled by default 1- Power up reset enabled by default
1	RW	0x0	macphy_power_down Power Down bit. Default value of True power down bit 1 - True power down is active by default 0 - True power down is not active by default
0	RW	0x1	macphy_enable PHY enable signal (active high). 1 = Enable MACHY IP 0 = Disable MACHY IP

GRF_MACPHY_CON1

Address: Operational Base + offset (0x0b04)

MACPHY control register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Reserved reserved
15	RW	0x0	polarity_stat_tx polarity control of tx status
14	RW	0x0	polarity_stat_rx polarity control of rx status
13	RW	0x0	polarity_stat_duplex polarity control of duplex status
12	RW	0x0	polarity_stat_link polarity control of link status
11	RW	0x0	polarity_stat_speed10 polarity control of speed10 status
10	RW	0x0	polarity_stat_speed100 polarity control of speed100 status
9	RW	0x0	grf_con_rmii_mode rmii_mode bit control
8	RW	0x0	macphy_speed_sel 0: speed 100 1: speed 10
7:3	RW	0x00	macphy_phy_addr PHY ADD register field. Must be unique in multi-PHY environment (like repeater).
2:0	RW	0x0	macphy_np_msg_code Next Page Message Code. Automatic generation of Next page with fault code

GRF_MACPHY_CON2

Address: Operational Base + offset (0x0b08)

MACPHY control register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Reserved reserved
15:0	RW	0x0000	macphy_id PHY ID Number,macphy_cfg_phy_id[15:0]

GRF_MACPHY_CON3

Address: Operational Base + offset (0x0b0c)

MACPHY control register3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Reserved reserved
15:12	RW	0x0	macphy_cfg_rev_nr Manufacturer’s Revision Number
11:6	RW	0x00	macphy_model_nr Manufacturer’s Model Number
5:0	RW	0x00	macphy_id PHY ID Number,macphy_cfg_phy_id[21:16]

GRF_MACPHY_STATUS
Address: Operational Base + offset (0x0b10)
MACPHY status register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RO	0x0	macphy_stat_speed100 macphy_stat_speed100 bit status Speed100 indication. Output driven low
5	RO	0x0	macphy_stat_speed10 macphy_stat_speed10 bit status Speed10 indication. Output is driven low
4	RO	0x0	macphy_stat_duplex macphy_stat_duplex bit status Duplex indication (low = full-duplex mode).Output is driven low
3	RO	0x0	macphy_stat_rx macphy_stat_rx bit status RX activity indication.Output is driven low
2	RO	0x0	macphy_stat_link macphy_stat_link bit status Link ON indication. Output is driven low
1	RO	0x0	macphy_stat_tx macphy_stat_tx bit status TX activity indication.Output is driven low
0	RO	0x0	macphy_stat_powerup_reset macphy_stat_powerup_reset bit status Power up reset state signal. To signal to the system that PHY is out of power down mode

3.4 DDR_GRF Register Description

3.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDR_GRF_DDR_CON0	0x0000	W	0x00000000	DDR Control Register0
DDR_GRF_DDR_CON1	0x0004	W	0x00000000	DDR Control Register1
DDR_GRF_DDR_CON2	0x0008	W	0x00000000	DDR Control Register2
DDR_GRF_DDR_CON3	0x000c	W	0x00000000	DDR Control Register3
DDR_GRF_DDR_STATUS0	0x0100	W	0x00000000	DDR Status Register0
DDR_GRF_DDR_STATUS1	0x0104	W	0x00000000	DDR Status Register1
DDR_GRF_DDR_STATUS2	0x0108	W	0x00000000	DDR Status Register2
DDR_GRF_DDR_STATUS3	0x010c	W	0x00000000	DDR Status Register3
DDR_GRF_DDR_STATUS4	0x0110	W	0x00000000	DDR Status Register4
DDR_GRF_DDR_STATUS5	0x0114	W	0x00000000	DDR Status Register5
DDR_GRF_DDR_STATUS6	0x0118	W	0x00000000	DDR Status Register6
DDR_GRF_DDR_STATUS7	0x011c	W	0x00000000	DDR Status Register7
DDR_GRF_DDR_STATUS8	0x0120	W	0x00000000	DDR Status Register8
DDR_GRF_DDR_STATUS9	0x0124	W	0x00000000	DDR Status Register9
DDR_GRF_DDR_STATUS10	0x0128	W	0x00000000	DDR Status Register10

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

3.4.2 Detail Register Description

DDR_GRF_DDR_CON0

Address: Operational Base + offset (0x0000)

DDR Control Register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_con_csysreq_upctl_ddrstdby csysreq_upctl_ddrstdby bit control 1'b0: Let ddrstdby to control csysreq of upctl. 1'b1: Disable ddrstdby to control scysreq of upctl

Bit	Attr	Reset Value	Description
14	RW	0x0	grf_con_csysreq_upctl_pmu csysreq_upctl_pmu bit control 1'b0: Let pmu to control csysreq of upctl. 1'b1: Disable pmu to control scysreq of upctl
13:12	RW	0x0	grf_con_dfi_phymstr_type dfi_phymstr_type bit control Indicates which of the 4 types of PHY master interface times the dfi_phymstr_req signal is requesting: 00 - tphymstr_type0 01 - tphymstr_type1 10 - tphymstr_type2 11 - tphymstr_type3 For debug only.
11	RW	0x0	grf_con_dfi_phymstr_state_sel dfi_phymstr_state_sel bit control DFI PHY Master State Select: Indicates the state requested by the PHY: 0 - IDLE 1 - Self-Refresh For debug only.
10:9	RW	0x0	grf_con_dfi_phymstr_cs_state dfi_phymstr_cs_state bit control Indicates the state of the DRAM when the PHY becomes the master: 0 - the PHY specifies the required state, using the dfi_phymstr_state_sel signal 1 - the PHY does not specify the state This signal is valid only when dfi_phymstr_req is asserted. Each memory rank uses one bit. For debug only.
8	RW	0x0	grf_con_dfi_phymstr_req dfi_phymstr_req bit control Indicates if set that the PHY requests control on the DFI bus. For debug only.
7	RW	0x0	grf_con_upctl_axi upctl_axi bit control AXI Low-Power Request. Active low, it requests upctl to enter a low-power state.
6	RW	0x0	grf_con_upctl_arurgent_0 upctl_arurgent_0 bit control AXI Read Urgent. Sideband signal to indicate a read urgent transaction. When asserted, if rd_port_urgent_en register is set, causes the port arbiter to switch immediately to read. It can be asserted anytime, it's not associated to any particular command

Bit	Attr	Reset Value	Description
5	RW	0x0	grf_con_upctl_arposion upctl_arposion bit control AXI Read poison. Sideband signal to indicate an invalid read transaction. When asserted, all zeros are returned at the output. If not needed, signal must be tied to zero.
4	RW	0x0	grf_con_upctl_awposion upctl_awposion bit control AXI Write poison. Sideband signal to indicate an invalid write transaction. When asserted, no data is written to the memory. If not needed, signal must be tied to zero.
3	RW	0x0	grf_con_upctl_awurgent upctl_awurgent bit control AXI Write Urgent. Sideband signal to indicate a write urgent transaction. When asserted, if wr_port_urgent_en register is set, causes the port arbiter to switch immediately to write. It can be asserted anytime, it's not associated to any particular command
2	RW	0x0	grf_con_pa_wmask pa_wmask bit control When asserted (active high), it will mask (prevent) the corresponding application port write address channel from requesting to the PA. For debug only.
1:0	RW	0x0	grf_con_pa_rmask pa_rmask bit control When asserted (active high), it will mask (prevent) the corresponding application port read address channel from requesting to the PA. There are 2 bits for each port, first one for the blue queue, second for the red queue. For debug only.

DDR_GRF_DDR_CON1

Address: Operational Base + offset (0x0004)

DDR Control Register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:12	RW	0x0	grf_con_upctl_awregion upctl_awregion bit control AXI 4 Write Address REGION signal. This signals is not used by the Controller.
11:8	RW	0x0	grf_con_upctl_arregion upctl_arregion bit control AXI 4 Read Address REGION signal. This signals is not used by the Controller.
7:4	RW	0x0	grf_con_upctl_arqos upctl_arqos bit control AXI Read Quality of Service. Sideband signal to indicate the quality of service attributes of the write transaction. For singleport configurations, this signal has no effect.
3:0	RW	0x0	grf_con_upctl_awqos upctl_awqos bit control AXI Write Quality of Service. Sideband signal to indicate the quality of service attributes of the write transaction. For singleport configurations, this signal has no effect.

DDR_GRF_DDR_CON2

Address: Operational Base + offset (0x0008)

DDR Control Register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_dfi_init_start dfi_init_start bit control
14	RW	0x0	grf_dfi_init_start_sel dfi_init_start_sel control 1: set ddrphy dfi init start controlled by grf_dfi_init_start 0: set ddrphy dif init start controlled by by upctl
13	RW	0x0	grf_upctl_apb_gate_en upctl_apb_gate_en bit control When set to 1 and axi_cg_en=1 and axi_cactive_0=0, axi clock of upctl will be auto gated when there is no axi traffic and apb traffic.

Bit	Attr	Reset Value	Description
12	RW	0x0	grf_ddrc_idle_sel ddrc_idle_sel control 1: select the ~ddrc_cactive as ddrcstdby ctl_idle 0: select ctl_idel of upctl as ddrcstdby ctl_idle. It should set to 0x1.
11:9	RO	0x0	reserved
8	RW	0x0	grf_con_dfi_lp_ack dfi_lp_ack bit control The control signal of auto gated ddrc_core_clk. It should be 0x0.
7	RW	0x0	grf_con_dfi_lp_req dfi_lp_req bit control The control signal of auto gated ddrc_core_clk. It should be 0x0.
6	RW	0x0	grf_con_dfi_phyupd_req dfi_phyupd_req bit control The control signal of auto gated ddrc_core_clk. It should be 0x0.
5:2	RW	0x0	grf_con_ddrc_auto_sr_dly ddrc_auto_sr_dly bit control The delay of auto gated ddrc_core_clk. It should be to be 0x6.
1	RW	0x0	grf_con_ddrc_cg_en ddrc_cg_en bit control when ddrc_cg_en=1, ddrc_cactive=0 and in auto self-refresh state, ddrc_core_clock of upctl will be auto gated.
0	RW	0x0	grf_con_axi_cg_en axi_cg_en bit control when axi_cg_en=1 and axi_cactive_0=0, axi clock of upctl will be auto gated when there is no axi traffic.

DDR_GRF_DDR_CON3

Address: Operational Base + offset (0x000c)

DDR Control Register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	dfi_ctrlupd_ack2 dfi_ctrlupd_ack2 bit control Second acknowledgement signal for the Controller initiated update request. This is to be used for legacy PHYs.
11	RW	0x0	dfi_ctrlupd_ack dfi_ctrlupd_ack bit control This signal is asserted to acknowledge a Controller initiated update request. The PHY is not required to acknowledge this request.
10	RW	0x0	dfi_phyupd_req dfi_phyupd_req bit control DFI PHY-initiated Update Request: Indicates if set that the PHY requires the DFI to be idle, i.e. DFI command, read data and write data channels to be inactive, for a specified period of time.
9:8	RW	0x0	dfi_phyupd_type dfi_phyupd_type bit control DFI PHY-initiated Update Select: Indicates which one of the 4 types of PHY update times is being requested by the dfi_phyupd_req signal. Valid values are: 00 - Tphyupd_type0 01 - Tphyupd_type1 10 - Tphyupd_type2 11 - Tphyupd_type3
7:6	RW	0x0	dfi_wrlvl_mode dfi_wrlvl_mode bit control Defines responsibility over the write leveling operation. The following modes are supported: 00 - Write leveling is not supported by the PHY 10 - PHY WrLvl evaluation mode. The Controller enables and disables the write leveling logic in the PHY. The PHY contains logic to evaluate the results and set new delay values; 11 - PHY WrLvl independent mode. The PHY performs all write leveling operations; Controller WrLvl evaluation mode is not supported. 01 - Not supported (MC WrLvl evaluation mode).

Bit	Attr	Reset Value	Description
5:4	RW	0x0	dfi_rdlvl_gate_mode dfi_rdlvl_gate_mode bit control Defines responsibility over the read gate training operation. Read gate training is available for all modes: DDR2/DDR3/DDR4/mDDR/LPDDR2/LPDDR3. The following modes are supported: 00 - Gate training is not supported by the PHY 10 - PHY RdLvl evaluation mode. The Controller enables and disables the gate training logic in the PHY. The PHY contains logic to evaluate the results and to set new delay values 11 - PHY RdLvl independent mode. The PHY performs all read DQS eye training operations 01 - Not supported (MC RdLvl evaluation mode). It should be 0x3.
3:2	RW	0x0	dfi_rdlvl_mode dfi_rdlvl_mode bit control Defines responsibility over the read DQS eye training leveling operation. Read DQS eye training is available for DDR3/DDR4 or LPDDR2/LPDDR3 designs. The following modes are supported: 00 - Read leveling is not supported by the PHY; 10 - PHY RdLvl evaluation mode. The Controller enables and disables the read leveling logic in the PHY. The PHY contains logic to evaluate the results and set new delay values. 11 - PHY RdLvl independent mode. The PHY performs all read leveling operations. 01 - Not supported (MC RdLvl evaluation mode). It should be set to 0x3.
1:0	RW	0x0	dfi_alert_n dfi_alert_n bit control CRC or Parity error signal. It should be set to 0x3.

DDR_GRF_DDR_STATUS0

Address: Operational Base + offset (0x0100)

DDR Status Register0

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28	RO	0x0	ctl_idle_upctl ctl_idle_upctl bit status Signal to be used in conjunction with certain PHYs only - to trigger the PHY's Anti-Aging feature. This signal is not part of the DFI interface. ctl_idle is asserted at same time as dfi_lp_req - therefore is asserted only if DFI Low Power Interface is enabled via DFILPCFG0.dfi_lp_en_pd or DFILPCFG0.dfi_lp_en_sr or DFILPCFG0.dfi_lp_en_dpd or DFILPCFG1.dfi_lp_en_mpsm. It is enabled via DFIMISC.ctl_idle_en.
27	RO	0x0	grf_st_dfi_phymstr_ack upctl_dfi_phymstr_ack bit status When asserted, the PHY is the master of DRAM bus.
26	RO	0x0	grf_st_upctl_raq_pop_0 upctl_raq_pop_0 bit status Transaction read from the Read address FIFO (synchronous to core_ddrc_core_clk).
25	RO	0x0	grf_st_upctl_raq_push_0 upctl_raq_push_0 bit status Transaction written to the Read address FIFO (synchronous to aclk_0).
24	RO	0x0	grf_st_upctl_raq_split_0 upctl_raq_split_0 bit status First portion of a wrap burst going to the Read address FIFO (synchronous to aclk_0).
23	RO	0x0	grf_st_upctl_waq_pop_0 upctl_waq_split_0 bit status Transaction read from the Write address FIFO (synchronous to core_ddrc_core_clk).
22	RO	0x0	grf_st_upctl_waq_push_0 upctl_waq_split_0 bit status Transaction written to the Write address FIFO (synchronous to aclk_0).
21	RO	0x0	grf_st_upctl_waq_split_0 upctl_waq_split_0 bit status First portion of a wrap burst going to the Write address FIFO (synchronous to aclk_0).
20:14	RO	0x00	grf_st_lpr_credit_cnt lpr_credit_cnt bit status Number of available Low priority read CAM slots (free positions).Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the read CAM (LPR store)

Bit	Attr	Reset Value	Description
13:7	RO	0x00	grf_st_hpr_credit_cnt hpr_credit_cnt bit status Number of available High priority read CAM slots (free positions). Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the read CAM (HPR store).
6:0	RO	0x00	grf_st_wr_credit_cnt wr_credit_cnt bit status Number of available write CAM slots (free positions). Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the write CAM.

DDR_GRF_DDR_STATUS1

Address: Operational Base + offset (0x0104)

DDR Status Register1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	RO	0x0	hif_refresh_req_bank hif_refresh_req_bank bit status Indicates the next bank which will be refreshed; for multi-rank configurations, the bank number is reported independently for each rank, and the information for all ranks is concatenated to form this signal.
13:12	RO	0x0	stat_upctl_reg_selfref_type stat_upctl_reg_selfref_type bit status DDRC Self Refresh status and type. Equivalent to STAT.selfref_type register.
11	RO	0x0	csysack_upctl_axi csysack_upctl_axi bit status AXI Low-Power Request Acknowledge. Acknowledgement from the peripheral (Port 0) of a grf request.
10	RO	0x0	cactive_upctl_axi cactive_upctl_axi bit status AXI Clock Active. Indicates that the peripheral (Port 0) requires its clock signal
9:8	RO	0x0	reserved
7:4	RO	0x0	grf_st_upctl_raq_wcount_0 upctl_raq_wcount_0 bit status Number of used positions in the Read address FIFO (synchronous to core_ddrc_core_clk).

Bit	Attr	Reset Value	Description
3:0	RO	0x0	grf_st_upctl_waq_wcount_0 upctl_waq_wcount_0 bit status Number of used positions in the Write address FIFO (synchronous to core_ddrc_core_clk)

DDR_GRF_DDR_STATUS2

Address: Operational Base + offset (0x0108)

DDR Status Register2

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:14	RO	0x00	grf_st_wr_credit_cnt wr_credit_cnt bit status Number of available write CAM slots (free positions). Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the write CAM.
13:7	RO	0x00	grf_st_hpr_credit_cnt hpr_credit_cnt bit status Number of available High priority read CAM slots (free positions). Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the read CAM (HPR store).
6:0	RO	0x00	grf_st_lpr_credit_cnt lpr_credit_cnt bit status Number of available Low priority read CAM slots (free positions).Each slots holds a DRAM burst Synchronous to core clock (core_ddrc_core_clk). Value is decremented/incremented as the commands flow in out of the read CAM (LPR store)

DDR_GRF_DDR_STATUS3

Address: Operational Base + offset (0x010c)

DDR Status Register3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[31:0] DDR_STATUS3~DDR_STATUS10 are Mode Register Read Data. mrr_data0[31:0] data status. (LPDDR2/3/4): Mode register read data. (DDR4): Multi-purpose register (MPR) read data. Valid when hif_mrr_data_valid is high. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 or DDR4 For DDR4, the width of this signal is equal to the width of the dfi_rddata signal. DDR4 MPR read data received on the DFI interface can be read on hif_mrr_data when hif_mrr_data_valid is asserted.

DDR_GRF_DDR_STATUS4
Address: Operational Base + offset (0x0110)
DDR Status Register4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[63:32] mrr_data0[63:32] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS5
Address: Operational Base + offset (0x0114)
DDR Status Register5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[95:64] mrr_data0[95:64] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS6
Address: Operational Base + offset (0x0118)
DDR Status Register6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data0[127:96] mrr_data0[127:96] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS7
Address: Operational Base + offset (0x011c)
DDR Status Register7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[31:0] mrr_data1[31:0] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS8
Address: Operational Base + offset (0x0120)
DDR Status Register8

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[63:32] mrr_data1[63:32] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS9
Address: Operational Base + offset (0x0124)
DDR Status Register9

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[95:64] mrr_data1[95:64] data status. See DDR_STATUS3.

DDR_GRF_DDR_STATUS10
Address: Operational Base + offset (0x0128)
DDR Status Register10

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mrr_data1[127:96] mrr_data1[127:96] data status. See DDR_STATUS3.

3.5 USB2PHY_GRF Register Description

3.5.1 Internal Address Mapping

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

3.5.2 Registers Summary

Name	Offset	Size	Reset Value	Description
USBPHY_REG0	0x0000	W	0x00002146	USB PHY Register0
USBPHY_REG1	0x0004	W	0x00000000	USB PHY Register1
USBPHY_REG2	0x0008	W	0x00000002	USB PHY Register2
USBPHY_REG3	0x000c	W	0x000000c8	USB PHY Register3
USBPHY_REG4	0x0010	W	0x000015b4	USB PHY Register4
USBPHY_REG5	0x0014	W	0x000011cb	USB PHY Register5
USBPHY_REG6	0x0018	W	0x0000022b	USB PHY Register6
USBPHY_REG7	0x001c	W	0x00000044	USB PHY Register7
USBPHY_REG8	0x0020	W	0x00000000	USB PHY Register8
USBPHY_REG9	0x0024	W	0x00000000	USB PHY Register9
USBPHY_REG10	0x0028	W	0x00000000	USB PHY Register10
USBPHY_REG11	0x002c	W	0x00000000	USB PHY Register11
USBPHY_REG12	0x0030	W	0x00002146	USB PHY Register12
USBPHY_REG13	0x0034	W	0x00000000	USB PHY Register13
USBPHY_REG14	0x0038	W	0x00000002	USB PHY Register14
USBPHY_REG15	0x003c	W	0x000000c8	USB PHY Register15
USBPHY_REG16	0x0040	W	0x000015b4	USB PHY Register16
USBPHY_REG17	0x0044	W	0x000011cb	USB PHY Register17
USBPHY_REG18	0x0048	W	0x00000005	USB PHY Register18
USBPHY_REG19	0x004c	W	0x00000044	USB PHY Register19
USBPHY_REG20	0x0050	W	0x00000000	USB PHY Register20
USBPHY_REG21	0x0054	W	0x00000000	USB PHY Register21
USBPHY_REG22	0x0058	W	0x00000000	USB PHY Register22
USBPHY_REG23	0x005c	W	0x00000000	USB PHY Register23

Name	Offset	Size	Reset Value	Description
USBPHY_CON0	0x0100	W	0x00000052	USB PHY control register0
USBPHY_CON1	0x0104	W	0x000001d2	USB PHY control register1
USBPHY_CON2	0x0108	W	0x00000000	USB PHY control register2
USBPHY_CON3	0x010c	W	0x00000019	USB PHY control register3
SIG_DETECT_USB2PHY_CON0	0x0110	W	0x00000000	SIG DETECT USB2PHY control register0

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

3.5.3 Detail Register Description

USBPHY_REG0

Address: Operational Base + offset (0x0000)

USB PHY Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x2146	usbphy_reg0 usbcomb phy control reg. BIT15 to 0 usbcomb phy control reg. BIT15 to 0

USBPHY_REG1

Address: Operational Base + offset (0x0004)

USB PHY Register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg1 usbcomb phy control reg. BIT31 to 16 usbcomb phy control reg. BIT31 to 16

USBPHY_REG2
Address: Operational Base + offset (0x0008)
USB PHY Register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0002	usbphy_reg2 usbcomb phy control reg. BIT47 to 32 usbcomb phy control reg. BIT47 to 32

USBPHY_REG3
Address: Operational Base + offset (0x000c)
USB PHY Register3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x00c8	usbphy_reg3 usbcomb phy control reg. BIT63 to 48 usbcomb phy control reg. BIT63 to 48

USBPHY_REG4
Address: Operational Base + offset (0x0010)
USB PHY Register4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x15b4	usbphy_reg4 usbcomb phy control reg. BIT79 to 64 usbcomb phy control reg. BIT79 to 64

USBPHY_REG5
Address: Operational Base + offset (0x0014)
USB PHY Register5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x11cb	usbphy_reg5 usbcomb phy control reg. BIT95 to 80 usbcomb phy control reg. BIT95 to 80

USBPHY_REG6
Address: Operational Base + offset (0x0018)
USB PHY Register6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x022b	usbphy_reg6 usbcomb phy control reg. BIT111 to 96 usbcomb phy control reg. BIT111 to 96

USBPHY_REG7
Address: Operational Base + offset (0x001c)
USB PHY Register7

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0044	usbphy_reg7 usbcomb phy control reg. BIT127 to 112 usbcomb phy control reg. BIT127 to 112

USBPHY_REG8
Address: Operational Base + offset (0x0020)
USB PHY Register8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg8 usbcomb phy control reg. BIT143 to 128 usbcomb phy control reg. BIT143 to 128

USBPHY_REG9
Address: Operational Base + offset (0x0024)
USB PHY Register9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg9 usbcomb phy control reg. BIT159 to 144 usbcomb phy control reg. BIT159 to 144

USBPHY_REG10

Address: Operational Base + offset (0x0028)

USB PHY Register10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg10 usbcomb phy control reg. BIT175 to 160 usbcomb phy control reg. BIT175 to 160

USBPHY_REG11

Address: Operational Base + offset (0x002c)

USB PHY Register11

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	usbphy_reg11 usbcomb phy control reg. BIT191 to 176 usbcomb phy control reg. BIT191 to 176

USBPHY_REG12

Address: Operational Base + offset (0x0030)

USB PHY Register12

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x2146	usbphy_reg12 usbcomb phy control reg. BIT207 to 192 usbcomb phy control reg. BIT207 to 192

USBPHY_REG13

Address: Operational Base + offset (0x0034)

USB PHY Register13

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg13 usbcomb phy control reg. BIT223 to 208 usbcomb phy control reg. BIT223 to 208

USBPHY_REG14

Address: Operational Base + offset (0x0038)

USB PHY Register14

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0002	usbphy_reg14 usbcomb phy control reg. BIT239 to 224 usbcomb phy control reg. BIT239 to 224

USBPHY_REG15

Address: Operational Base + offset (0x003c)

USB PHY Register15

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x00c8	usbphy_reg15 usbcomb phy control reg. BIT255 to 240 usbcomb phy control reg. BIT255 to 240

USBPHY_REG16

Address: Operational Base + offset (0x0040)

USB PHY Register16

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x15b4	usbphy_reg16 usbcomb phy control reg. BIT271 to 256 usbcomb phy control reg. BIT271 to 256

USBPHY_REG17

Address: Operational Base + offset (0x0044)

USB PHY Register17

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x11cb	usbphy_reg17 usbcomb phy control reg. BIT287 to 272 usbcomb phy control reg. BIT287 to 272

USBPHY_REG18

Address: Operational Base + offset (0x0048)

USB PHY Register18

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0005	usbphy_reg18 usbcomb phy control reg. BIT303 to 288 usbcomb phy control reg. BIT303 to 288

USBPHY_REG19

Address: Operational Base + offset (0x004c)

USB PHY Register19

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0044	usbphy_reg19 usbcomb phy control reg. BIT319 to 304 usbcomb phy control reg. BIT319 to 304

USBPHY_REG20

Address: Operational Base + offset (0x0050)

USB PHY Register20

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg20 usbcomb phy control reg. BIT335 to 320 usbcomb phy control reg. BIT335 to 320

USBPHY_REG21

Address: Operational Base + offset (0x0054)

USB PHY Register21

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	usbphy_reg21 usbcomb phy control reg. BIT351 to 336 usbcomb phy control reg. BIT351 to 336

USBPHY_REG22

Address: Operational Base + offset (0x0058)

USB PHY Register22

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg22 usbcomb phy control reg. BIT367 to 352 usbcomb phy control reg. BIT367 to 352

USBPHY_REG23

Address: Operational Base + offset (0x005c)

USB PHY Register23

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	usbphy_reg23 usbcomb phy control reg. BIT383 to 368 usbcomb phy control reg. BIT383 to 368

USBPHY_CON0

Address: Operational Base + offset (0x0100)

USB PHY control register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x0	usbotg_utmi_iddig usbotg_utmi_iddig bit control USB Plug Indicator Ooutput
8	RW	0x0	usbotg_utmi_dmpulldown usbotg_utmi_dmpulldown bit control Enable DMINUS Pull Down resistor
7	RW	0x0	usbotg_utmi_dppulldown usbotg_utmi_dppulldown bit control Enable DPLUS Pull Down resistor
6	RW	0x1	usbotg_utmi_termselect usbotg_utmi_termselect bit control Termination select between FS/LS and HS Terminations
5:4	RW	0x1	usbotg_utmi_xcvrselect usbotg_utmi_xcvrselect bit control Transceiver Select between FS/LS and HS Transceivers
3:2	RW	0x0	usbotg_utmi_opmode usbotg_utmi_opmode bit control Operational mode selector between various modes
1	RW	0x1	usbotg_utmi_suspend_n usbotg_utmi_suspend_n bit control Suspend Mode enable 1'b0:suspend 1'b1:normal
0	RO	0x0	reserved

USBPHY_CON1

Address: Operational Base + offset (0x0104)

USB PHY control register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:9	RO	0x0	reserved
8	RW	0x1	usbhost_utmi_dmpulldown usbhost_utmi_dmpulldown bit control Enable DMINUS Pull Down resistor
7	RW	0x1	usbhost_utmi_dppulldown usbhost_utmi_dppulldown bit control Enable DPLUS Pull Down resistor
6	RW	0x1	usbhost_utmi_termselect usbhost_utmi_termselect bit control Termination select between FS/LS and HS Terminations
5:4	RW	0x1	usbhost_utmi_xcvrselect usbhost_utmi_xcvrselect bit control Transceiver Select between FS/LS and HS Transceivers
3:2	RW	0x0	usbhost_utmi_opmode usbhost_utmi_opmode bit control Operational mode selector between various modes
1	RW	0x1	usbhost_utmi_suspend_n usbhost_utmi_suspend_n bit control Suspend Mode enable 1'b0: suspend 1'b1: normal
0	RO	0x0	reserved

USBPHY_CON2

Address: Operational Base + offset (0x0108)

USB PHY control register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	vdm_src_en_usbotg vdm_src_en_usbotg bit control open dm voltage source
11	RW	0x0	vdp_src_en_usbotg vdp_src_en_usbotg bit control open dp voltage source
10	RW	0x0	rdm_pdwn_en_usbotg rdm_pdwn_en_usbotg bit control open dm pull down resistor
9	RW	0x0	idp_src_en_usbotg idp_src_en_usbotg bit control open dm source current
8	RW	0x0	idm_sink_en_usbotg idm_sink_en_usbotg bit control open dm sink current
7	RW	0x0	idp_sink_en_usbotg idp_sink_en_usbotg bit control open dp sink current
6:5	RO	0x0	reserved
4	RW	0x0	usbphy_commononn usbphy_commononn bit control configure PLL clock output in suspend mode
3	RW	0x0	bypasssel_usbotg bypasssel_usbotg bit control bypass select
2	RW	0x0	bypassdmen_usbotg bypassdmen_usbotg bit control bypass dm enable
1	RW	0x0	usbotg_disable_1 usbotg_disable_1 bit control bypass OTG function
0	RW	0x0	usbotg_disable_0 usbotg_disable_0 bit control bypass OTG function

USBPHY_CON3

Address: Operational Base + offset (0x010c)

USB PHY control register3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11	RW	0x0	usbhost_utmi_drvvbus usbhost_utmi_drvvbus bit control USB HOST utmi_fs_drvvbus bit control
10	RW	0x0	usbhost_utmi_drvvbus_sel usbhost_utmi_drvvbus_sel bit control USB HOST utmi_drvvbus_sel bit control
9	RW	0x0	usbhost_utmi_fs_se0 usbhost_utmi_fs_se0 bit control USB HOST utmi_fs_se0 bit control
8	RW	0x0	usbhost_utmi_fs_data usbhost_utmi_fs_data bit control USB HOST utmi_fs_data bit control
7	RW	0x0	usbhost_utmi_fs_oe usbhost_utmi_fs_oe bit control USB HOST utmi_fs_oe bit control
6	RW	0x0	usbhost_utmi_fs_xver_own usbhost_utmi_fs_xver_own bit control USB HOST utmi_fs_xver_own bit control
5	RW	0x0	usbhost_utmi_idpullup usbhost_utmi_idpullup bit control USB HOST utmi_idpullup bit control
4	RW	0x1	usbhost_utmi_dmpulldown usbhost_utmi_dmpulldown bit control Enable DMINUS Pull Down resistor
3	RW	0x1	usbhost_utmi_dppulldown usbhost_utmi_dppulldown bit control Enable DPLUS Pull Down resistor
2	RW	0x0	usbhost_utmi_dischrgvbus usbhost_utmi_dischrgvbus bit control USB HOST utmi_dischrgvbus bit control

Bit	Attr	Reset Value	Description
1	RW	0x0	usbhost_utmi_chrgvbus usbhost_utmi_chrgvbus bit control USB HOST utmi_chrgvbus bit control
0	RW	0x1	usbhost_utmi_drvvbus usbhost_utmi_drvvbus bit control USB HOST utmi_drvvbus bit control

SIG_DETECT_USB2PHY_CON0

Address: Operational Base + offset (0x0110)

SIG DETECT USB2PHY control register0

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RO	0x0	grf_stat_usbphy_dp_detected grf_stat_usbphy_dp_detected bit status grf_stat_usbphy_dp_detected bit status
24	RO	0x0	grf_stat_usbphy_cp_detected grf_stat_usbphy_cp_detected bit status grf_stat_usbphy_cp_detected bit status
23	RO	0x0	grf_stat_usbphy_dcp_detected grf_stat_usbphy_dcp_detected bit status grf_stat_usbphy_dcp_detected bit status
22	RO	0x0	usbhost_phy_ls_fs_rcv usbhost_phy_ls_fs_rcv bit status host_phy_ls_fs_rcv status
21	RO	0x0	usbhost_utmi_avalid usbhost_utmi_avalid bit status host_utmi_avalid status
20	RO	0x0	usbhost_utmi_bvalid usbhost_utmi_bvalid bit status host_utmi_bvalid status
19	RO	0x0	usbhost_utmi_hostdisconnect usbhost_utmi_hostdisconnect bit status host_utmi_hostdisconnect status
18	RO	0x0	usbhost_utmi_iddig_o usbhost_utmi_iddig_o bit status host_utmi_iddig_o status
17:16	RO	0x0	usbhost_utmi_linestate usbhost_utmi_linestate bit status host_utmi_linestate status

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	usbhost_utmi_sessend usbhost_utmi_sessend bit status host_utmi_sessend status
14	RO	0x0	usbhost_utmi_vbusvalid usbhost_utmi_vbusvalid bit status host_utmi_vbusvalid status
13	RO	0x0	usbhost_utmi_vmi usbhost_utmi_vmi bit status host_utmi_vmi status
12	RO	0x0	usbhost_utmi_vpi usbhost_utmi_vpi bit status host_utmi_vpi status
13:12	RW	0x0	host0_ls_filter_time_sel host0_ls_filter_time_sel bit control host0_ls_lfiter time select 00:100us 01:500us 10:1ms 11:10ms
11	RO	0x0	usbotg_phy_ls_fs_rcv usbotg_phy_ls_fs_rcv bit status utmi_phy_ls_fs_rcv_out status
10	RO	0x0	usbotg_utmi_avalid usbotg_utmi_avalid bit status otg_utmi avalid bit status
11:10	RW	0x0	otg0_ls_filter_time_sel otg0_ls_filter_time_sel bit control otg0_ls_lfiter time select 00:100us 01:500us 10:1ms 11:10ms
9	RO	0x0	usbotg_utmi_bvalid usbotg_utmi_bvalid bit status otg_utmi bvalid bit status

Bit	Attr	Reset Value	Description
8	RO	0x0	usbotg_utmi_fs_xver_own usbotg_utmi_fs_xver_own bit status OTG utmi_fs_xver_own bit control
9:8	RW	0x0	otg0_id_filter_time_sel otg0_id_filter_time_sel bit control otg0_id_filter_time select 00:5ms 01:15ms 10:35ms
7	RO	0x0	usbotg_utmi_hostdisconnect usbotg_utmi_hostdisconnect bit status otg_utmi_hostdisconnect status
6	RO	0x0	usbotg_utmi_iddig usbotg_utmi_iddig bit status usbotg_utmi_iddig select between grf and phy 1:from grf 0:from phy
5:4	RO	0x0	usbotg_utmi_linestate usbotg_utmi_linestate bit status otg_utmi_linestate bit status
5:4	RO	0x0	otg0_id_irq otg0_id_irq bit status otg0_id bit status
5:4	RW	0x0	otg0_id_irq otg0_id_irq bit control otg0_id bit status
3	RO	0x0	usbotg_utmi_sessend usbotg_utmi_sessend bit status otg_utmi_sessend bit status
2	RO	0x0	usbotg_utmi_vbusvalid usbotg_utmi_vbusvalid bit status otg_utmi_vbusvalid bit status
3:2	RO	0x0	otg0_bvalid_irq otg0_bvalid_irq bit status otg0_bvalid bit status
3:2	RW	0x0	otg0_bvalid_irq otg0_bvalid_irq bit control otg0_bvalid bit status
1	RO	0x0	usbotg_utmi_vmi usbotg_utmi_vmi bit status otg_utmi_vmi bit status
1	RW	0x0	host0_linestate_irq host0_linestate_irq bit control host0_linestate bit status

Bit	Attr	Reset Value	Description
1	RO	0x0	host0_linestate_irq host0_linestate_irq bit status host0_linestate bit status
0	RO	0x0	usbotg_utmi_vpi usbotg_utmi_vpi bit status otg_utmi_vpi bit status
0	RO	0x0	otg0_linestate_irq otg0_linestate_irq bit status otg0_linestate bit status
0	RW	0x0	otg0_linestate_irq otg0_linestate_irq bit control otg0_linestate bit status

3.6 USB3PHY_GRF Register Description

3.6.1 Internal Address Mapping

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

3.6.2 Registers Summary

Name	Offset	Size	Reset Value	Description
USB3PHY_CON0	0x0000	W	0x00000000	USB3 PHY Control Register0
USB3PHY_CON1	0x0004	W	0x00000000	USB3 PHY Control Register1
USB3PHY_CON2	0x0008	W	0x00000000	USB3 PHY Control Register2
USB3PHY_CON3	0x000c	W	0x00000001	USB3 PHY Control Register3
USB3PHY_CON4	0x0010	W	0x00000000	USB3 PHY Control Register4
USB3PHY_CON5	0x0014	W	0x00000000	USB3 PHY Control Register5
USB3PHY_CON6	0x0018	W	0x00000000	USB3 PHY Control Register6
USB3PHY_CON7	0x001c	W	0x00000000	USB3 PHY Control Register7
USB3PHY_CON8	0x0020	W	0x00000014	USB3 PHY Control Register8
USB3PHY_CON9	0x0024	W	0x00000000	USB3 PHY Control Register9
USB3PHY_SIG_DETECT_CON0	0x0028	W	0x00000000	USB3 PHY SIG DETECT Control Register0
USB3PHY_STATUS1	0x0034	W	0x00000000	USB3 PHY STATUS1 Register1
USB3_WAKEUP_CON0	0x0040	W	0x00000000	USB3 WAKEUP Control Register0

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

3.6.3 Detail Register Description

USB3PHY_CON0

Address: Operational Base + offset (0x0000)

USB3 PHY Control Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12	RW	0x0	vdm_src_en_usb3otg vdm_src_en_usb3otg bit control open dm voltage source
11	RW	0x0	vdp_src_en_usb3otg vdp_src_en_usb3otg bit control open dp voltage source
10	RW	0x0	rdm_pdwn_en_usb3otg rdm_pdwn_en_usb3otg bit control open dm pull down resistor
9	RW	0x0	idp_src_en_usb3otg idp_src_en_usb3otg bit control open dm source current
8	RW	0x0	idm_sink_en_usb3otg idm_sink_en_usb3otg bit control open dm sink current
7	RW	0x0	idp_sink_en_usb3otg idp_sink_en_usb3otg bit control open dp sink current
6:0	RO	0x0	reserved

USB3PHY_CON1

Address: Operational Base + offset (0x0004)

USB3 PHY Control Register1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; …… When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	usb3otg_utmi_iddig usb3otg_utmi_iddig bit control usb3otg_utmi_iddig bit control
0	RW	0x0	usb3otg_utmi_dmpulldown usb3otg_utmi_dmpulldown bit control usb3otg_utmi_dmpulldown bit control
0	RW	0x0	usb3otg_utmi_dppulldown usb3otg_utmi_dppulldown bit control usb3otg_utmi_dppulldown bit control
0	RW	0x0	usb3otg_utmi_suspend_n usb3otg_utmi_suspend_n bit control usb3otg_utmi_suspend_n bit control
0	RW	0x0	usb3otg_utmi_opmode usb3otg_utmi_opmode bit control usb3otg_utmi_opmode bit control
0	RW	0x0	usb3otg_utmi_xcvrselect usb3otg_utmi_xcvrselect bit control usb3otg_utmi_xcvrselect bit control
0	RW	0x0	usb3otg_utmi_termselect usb3otg_utmi_termselect bit control usb3otg_utmi_termselect bit control

USB3PHY_CON2

Address: Operational Base + offset (0x0008)

USB3 PHY Control Register2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con2 Reserved reserved

USB3PHY_CON3

Address: Operational Base + offset (0x000c)

USB3 PHY Control Register3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x0	usb3otg_utmi_fs_se0 usb3otg_utmi_fs_se0 bit control OTG utimi_fs_xver_own bit control
8	RW	0x0	usb3otg_utmi_fs_data usb3otg_utmi_fs_data bit control OTG utimi_fs_xver_own bit control
7	RW	0x0	usb3otg_utmi_fs_oe usb3otg_utmi_fs_oe bit control OTG utmi_fs_xver_own bit control
6	RW	0x0	usb3otg_utmi_fs_xver_own usb3otg_utmi_fs_xver_own bit control OTG utmi_fs_xver_own bit control
5	RO	0x0	reserved
4	RW	0x0	usb3otg_utmi_dischrgvbus usb3otg_utmi_dischrgvbus bit control USB3 OTG utmi_dischrgvbus bit control
3	RW	0x0	usb3otg_utmi_chrgvbus usb3otg_utmi_chrgvbus bit control USB3 OTG utmi_chrgvbus bit control
2	RW	0x0	usb3otg_utmi_drvvbus usb3otg_utmi_drvvbus bit control USB3 OTG utmi_drvvbus bit control
1	RW	0x0	usb3otg_utmi_drvvbus_sel usb3otg_utmi_drvvbus_sel bit control USB3 OTG utmi_drvvbus_sel bit control
0	RW	0x1	usb3otg_utmi_idpullup usb3otg_utmi_idpullup bit control USB3 OTG utmi_idpullup bit control

USB3PHY_CON4

Address: Operational Base + offset (0x0010)

USB3 PHY Control Register4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con4 usb3phy_con4 bit control reserved

USB3PHY_CON5

Address: Operational Base + offset (0x0014)

USB3 PHY Control Register5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con5 usb3phy_con5 bit control reserved

USB3PHY_CON6

Address: Operational Base + offset (0x0018)

USB3 PHY Control Register6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con6 usb3phy_con6 bit control reserved

USB3PHY_CON7

Address: Operational Base + offset (0x001c)

USB3 PHY Control Register7

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con7 usb3phy_con7 bit control reserved

USB3PHY_CON8

Address: Operational Base + offset (0x0020)

USB3 PHY Control Register8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usb3phy_usb2only usb3phy_usb2only bit control usb3phy_usb2only bit control
14	RW	0x0	usb3otg_pipe3_powerpresent usb3otg_pipe3_powerpresent bit control usb3otg_pipe3_powerpresent bit control
13:6	RO	0x0	reserved
5	RW	0x0	usb3otg_pipe3_txdetectrxloopbk usb3otg_pipe3_txdetectrxloopbk bit control usb3otg_pipe3_txdetectrxloopbk bit control
4:3	RW	0x2	usb3otg_pipe3_powerdown usb3otg_pipe3_powerdown bit control usb3otg_pipe3_powerdown bit control
2	RW	0x1	usb3otg_pipe3_txelecidle usb3otg_pipe3_txelecidle bit control usb3otg_pipe3_txelecidle bit control
1	RW	0x0	usb3otg_pipe3_rxtermination usb3otg_pipe3_rxtermination bit control usb3otg_pipe3_rxtermination bit control
0	RW	0x0	grf_con_usb3_sftsel grf_con_usb3_sftsel bit control grf_con_usb3_sftsel bit control

USB3PHY_CON9

Address: Operational Base + offset (0x0024)

USB3 PHY Control Register9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:1	RO	0x0	reserved
0	RW	0x0	usb3phy_con9 Reserved reserved

USB3PHY_SIG_DETECT_CON0

Address: Operational Base + offset (0x0028)

USB3 PHY SIG DETECT Control Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RO	0x0	grf_stat_usb3phy_dp_detected grf_stat_usb3phy_dp_detected bit status grf_stat_usb3phy_dp_detected bit status
13	RO	0x0	grf_stat_usb3phy_cp_detected grf_stat_usb3phy_cp_detected bit status grf_stat_usb3phy_cp_detected bit status
12	RO	0x0	grf_stat_usb3phy_dcp_detected grf_stat_usb3phy_dcp_detected bit status grf_stat_usb3phy_dcp_detected bit status
11	RO	0x0	usb3otg_utmireset usb3otg_utmireset bit status usb3otg_utmireset bit status
10	RO	0x0	usb3otg_phy_ls_fs_rcv usb3otg_phy_ls_fs_rcv bit status usb3otg_phy_ls_fs_rcv bit status

Bit	Attr	Reset Value	Description
9	RO	0x0	usb3otg_utmi_avalid usb3otg_utmi_avalid bit status usb3otg_utmi_avalid bit status
8	RO	0x0	usb3otg_utmi_bvalid usb3otg_utmi_bvalid bit status usb3otg_utmi_bvalid bit status
7	RO	0x0	usb3otg_utmi_hostdisconnect usb3otg_utmi_hostdisconnect bit status usb3otg_utmi_hostdisconnect bit status
6	RO	0x0	usb3otg_utmi_iddig usb3otg_utmi_iddig bit status usb3otg_utmi_iddig bit status
5:4	RO	0x0	usb3otg_utmi_linestate usb3otg_utmi_linestate bit status usb3otg_utmi_linestate bit status
3	RO	0x0	usb3otg_utmi_sessend usb3otg_utmi_sessend bit status usb3otg_utmi_sessend bit status
2	RO	0x0	usb3otg_utmi_vbusvalid usb3otg_utmi_vbusvalid bit status usb3otg_utmi_vbusvalid bit status
3:2	RW	0x0	otg0_ls_filter_time_sel otg0_ls_filter_time_sel bit control otg_ls filter time select 00:100us 01:500us 10:1ms 11:10ms
1	RO	0x0	usb3otg_utmi_vmi usb3otg_utmi_vmi bit status usb3otg_utmi_vmi bit status
0	RO	0x0	usb3otg_utmi_vpi usb3otg_utmi_vpi bit status usb3otg_utmi_vpi bit status
1:0	RW	0x0	otg0_id_filter_time_sel otg0_id_filter_time_sel bit control otg_id_filter time select 00:5ms 01:15ms 10:35ms

USB3PHY_STATUS1

Address: Operational Base + offset (0x0034)

USB3 PHY STATUS1 Register1

Bit	Attr	Reset Value	Description
31	RO	0x0	usb3phy_tx_pll_lock usb3phy_tx_pll_lock bit status usb3phy_tx_pll_lock bit status
30	RO	0x0	usb3otg_pipe3_reset_n usb3otg_pipe3_reset_n bit status usb3otg_pipe3_reset_n bit status
29:24	RO	0x0	reserved
23:16	RO	0x00	usb3_phy_obs usb3_phy_obs bit status usb3_phy_obs bit status
15	RO	0x0	usb3otg_pipe3_elasbuffermode usb3otg_pipe3_elasbuffermode bit status usb3otg_pipe3_elasbuffermode bit status
14:13	RO	0x0	usb3otg_pipe3_powerdown usb3otg_pipe3_powerdown bit status usb3otg_pipe3_powerdown bit status
12	RO	0x0	usb3otg_pipe3_rxeqtrain usb3otg_pipe3_rxeqtrain bit status usb3otg_pipe3_rxeqtrain bit status
11	RO	0x0	usb3otg_pipe3_rxpolarity usb3otg_pipe3_rxpolarity bit status usb3otg_pipe3_rxpolarity bit status
10	RO	0x0	usb3otg_pipe3_rxtermination usb3otg_pipe3_rxtermination bit status usb3otg_pipe3_rxtermination bit status
9	RO	0x0	usb3otg_pipe3_txdetectrxloopbk usb3otg_pipe3_txdetectrxloopbk bit status usb3otg_pipe3_txdetectrxloopbk bit status
8	RO	0x0	usb3otg_pipe3_compliance usb3otg_pipe3_compliance bit status usb3otg_pipe3_compliance bit status
7	RO	0x0	usb3otg_pipe3_txoneszeros usb3otg_pipe3_txoneszeros bit status usb3otg_pipe3_txoneszeros bit status
6	RO	0x0	usb3otg_pipe3_phystatus usb3otg_pipe3_phystatus bit status usb3otg_pipe3_phystatus bit status
5	RO	0x0	usb3otg_pipe3_rxelecidle usb3otg_pipe3_rxelecidle bit status usb3otg_pipe3_rxelecidle bit status
4:2	RO	0x0	usb3otg_pipe3_rxstatus usb3otg_pipe3_rxstatus bit status usb3otg_pipe3_rxstatus bit status

Bit	Attr	Reset Value	Description
1	RO	0x0	usb3otg_pipe3_rxvalid usb3otg_pipe3_rxvalid bit status usb3otg_pipe3_rxvalid bit status
0	RO	0x0	usb3otg_pipe3_powerpresent usb3otg_pipe3_powerpresent bit status usb3otg_pipe3_powerpresent bit status

USB3_WAKEUP_CON0

Address: Operational Base + offset (0x0040)

USB3 WAKEUP Control Register0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Bit0~15 write enable "When bit16=1, bit0 can be written by software. When bit16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 cannot be written by software;
15:6	RO	0x0	reserved
5:4	RO	0x0	usb3_id_irq usb3_id_irq bit status usb3_id_irq bit status
4	RW	0x0	usb3_rxdet_en usb3_rxdet_en bit control usb3_rxdet_en bit control
5:4	RW	0x0	usb3_id_irq usb3_id_irq bit control usb3_id_irq bit control
3:2	RO	0x0	usb3_bvalid_irq usb3_bvalid_irq bit status usb3_bvalid_irq bit status
3:2	RW	0x0	usb3_bvalid_irq usb3_bvalid_irq bit control usb3_bvalid_irq bit control
1	RO	0x0	usb3_rxdet_irq usb3_rxdet_irq bit status usb3_rxdet_irq bit status
1	RW	0x0	usb3_rxdet_irq usb3_rxdet_irq bit control usb3_rxdet_irq bit control

Bit	Attr	Reset Value	Description
0	RO	0x0	usb3_linestate_irq usb3_linestate_irq bit status usb3_linestate_irq bit status
0	RW	0x0	usb3_linestate_irq usb3_linestate_irq bit control usb3_linestate_irq bit control

Chapter 4 Cortex-A53

4.1 Overview

The RK3328 has a quad-core Cortex-A53 cluster with 256K L2 memory. Cortex-A53 processor, which is a mid-range, low-power processor that implements the ARMv8-A architecture.

The Cortex-A53 processor includes following features:

- Full implementation of the ARMv8-A architecture instruction set
- Support for both AArch32 and AArch64 Execution status.
- Support for all exception levels, EL0, EL1, EL2, and EL3, in each execution states.
- Support A32 instruction set, previously called the ARM instruction set.
- Support T32 instruction set, previously called the Thumb instruction set.
- Support A64 instruction set.
- In-order pipeline with symmetric dual-issue of most instructions.
- Harvard Level 1(L1) memory system with a Memory Management Unit (MMU).
- Level 2(L2) memory system providing cluster memory coherency, with L2 cache.
- Support advanced SIMD and Floating-point Extension for integer and floating-point vector operations.
- Support ARMv8 Cryptography Extensions.
- Support AMBA 4 ACE bus architecture.

The configuration details of little cluster and big cluster are shown in following tables

Table 1-1 CPU Configuration

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

4.2 Block Diagram

The Cortex-A53 sub system is shown in Figure 1-1. As illustrated, dual-core Cortex-A53 connects to system bus through asynchronous bridges which can handle with CDC(clock domain crossing) issue.

The Cortex-A53 is connected with system counter, which can run under a constant frequency clock, for PPI interrupt generation.

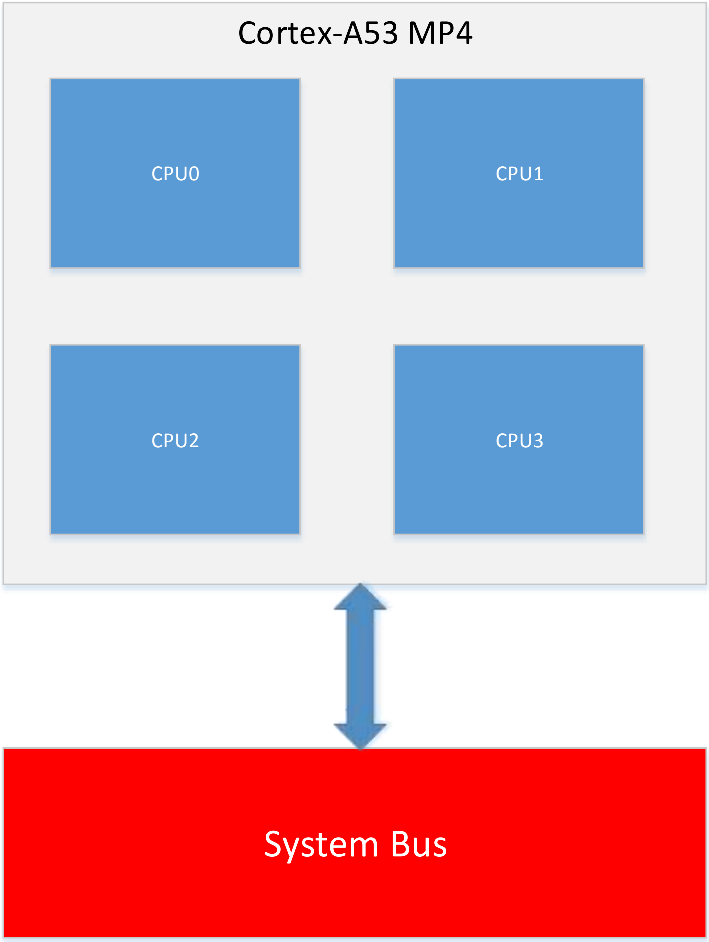


Fig. 4-1 Block Diagram

4.3 Function Description

Please refer to the document cortex_a53_r0p4_trm.pdf for the detail function description.

Chapter 5 Embedded SRAM

5.1 Overview

The Embedded SRAM is the AXI slave device, which supports read and write access to provide system fast access data storage

5.1.1 Features supported

- Provide 36KB access space
- Support security and non-security access
- Security or non-security space is software programmable
- Security space is nx4KB(up to whole memory space)
- Support 64bit AXI bus

5.1.2 Features not supported

- Don't support AXI lock transaction
- Don't support AXI exclusive transaction
- Don't support AXI cache function
- Don't support AXI protection function

5.2 Block Diagram

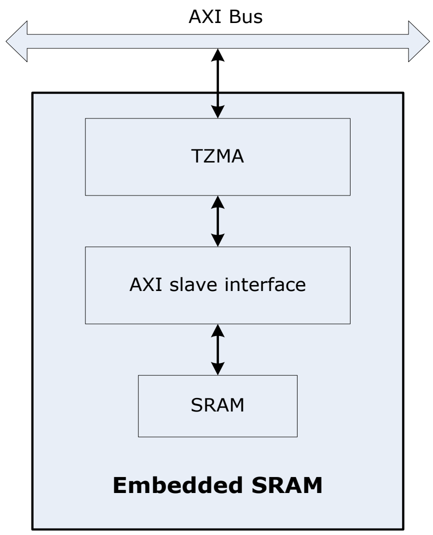


Fig. 5-1 Embedded SRAM block diagram

5.3 Function Description

5.3.1 TZMA

Please refer to 7.3.3 for TZMA functional description

5.3.2 AXI slave interface

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

5.3.3 Embedded SRAM access path

The Embedded SRAM can only be accessed by Cortex-A53, DMAC_BUS and CRYPTO

5.3.4 Remap

The Embedded SRAM support remap.

Before remap, the Embedded SRAM address range is 0xff09_0000~0xff09_8fff,
After set remap, (ref Security GRF register SGRF_SCON0, bit[10]), the system can still access the Embedded SRAM by the old address. at same time, the system also can access the Embedded SRAM by the new address 0xffff_0000 ~ 0xffff_8fff (include the bootaddr)

Chapter 6 Power Management Unit (PMU)

6.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RK3328. The RK3328 PMU is dedicated for managing the power of the whole chip.

6.1.1 Features

- Support DDR self-refresh
- Support DDR retention
- Support CPU2/CPU3 power down/up by software
- Support CPU2/CPU3 auto-power management
- Support L2 flush interface

6.2 Block Diagram

6.2.1 Voltage partition

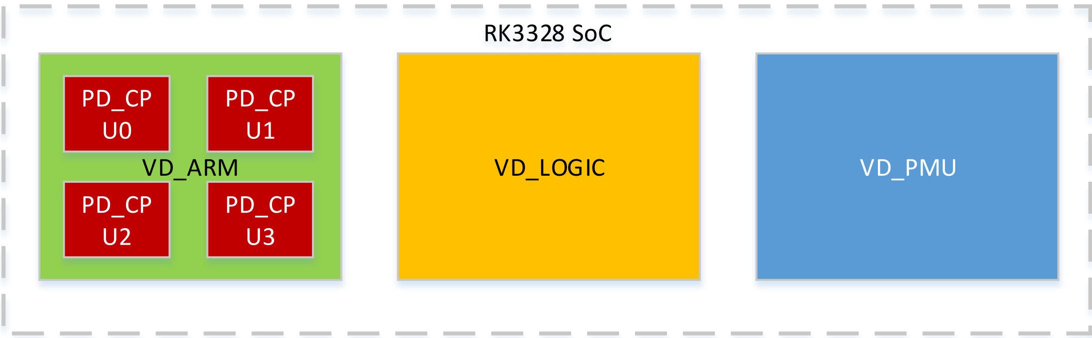


Fig. 6-1 RK3328 Power Domain Partition

The above diagram describes voltage domain partition, notice that there are no power domains inside RK3328 except PD_CPU2 and PD_CPU3. PD_CPU2 and PD_CPU3 have MTCMOS inside, and for the blocks with name pd_xxx are not real power domains.

Table 6-1 RK3328 Power Domain and Voltage Domain Summary

Voltage Domain	**Blocks (not real power domain)**	**Description**
VD_ARM	PD_CPU0	CPU Core 0 with NEON and FPU, DAP-lite
	PD_CPU1	CPU Core 1 with NEON and FPU, DAP-lite
	PD_CPU2	CPU Core 2 with NEON and FPU, DAP-lite
	PD_CPU3	CPU Core 3 with NEON and FPU, DAP-lite
	PD_SCU	DAP Lite, SCU and 256KB L2
VD_LOGIC	PD_GPU	Mali-450
	PD_RKVENC	Video encoder
	PD_RKVDEC	Video decoder, NANDC, EMMC, SDIO, SDMMC, GMAC2PHY, GMAC2IO
	PD_VIO	ISP, IEP, VOP, RGA, CIF0/1/2/3, TV decoder, HDMI host, DSI host
	PD_PERI	Peri NIU
	PD_DDR	UPCTL, MEM scheduler, DDR mon, DDR GRF

	PD_BUS & TOP	CRYPTO, SPDIF, I2S0/1/2, PDM, TSP, SGRF, SEFUSE, SOTP, SRAM(36KB), ROM(20KB), DDRPHY, ACODEC, VDAC, HDMI PHY, PLLx4, GRF, I2Cx4, WDT, CRU, TIMERx6, EFUSE1024, SCR, TSADC, PMU, SARADC, SPI, PWMx4, GPIOx4, UARTx3, DFI monitor, TSADC CTL, Stimerx2, DCF, NSEFUSE, NSOTP
	PD_VPU	VPU
VD_PMU	PD_PMU	OSC, Pmux, and PAD ring

6.2.2 PMU block diagram

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

- APB interface and register, which can accept the system configuration
- Low Power State Control, which generate low power control signals.
- Power Switch Control, which control all power domain switch

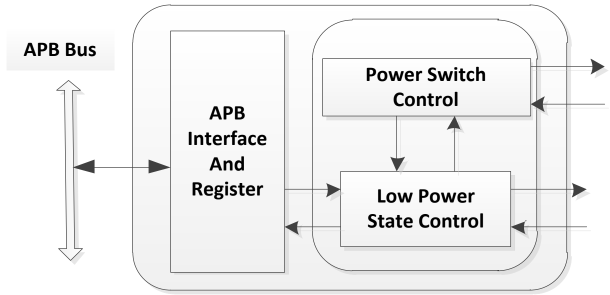


Fig. 4-2 PMU Bock Diagram

6.3 Function Description

First of all, we define two operation modes of PMU, normal mode and low power mode.
When operating at normal mode, that means software can manage power sources directly by accessing PMU register.
For example, Cortex-A53 CPU can write PMU_PWRDN_CON register to determine that power off/on which power domain independently.
When operating at low power mode, software manages power sources indirectly through FSM (Finite States Machine) in PMU and those settings always not take effect immediately.
That means software also can configure PMU registers to power down/up some power resources, but these setting will not be executed immediately after configuration. They will delay to execute after FSM running in particular phase.
To entering low power mode, after setting some power configurations, the PMU_POWER_MODE[0] bit must be set 1 to enable PMU FSM. Then Cortex-A53 CPU needs to execute a WFI command to perform ready signal. After PMU detects all Cortex-A53 CPUs in WFI status, then the FSM will be fetched. And the specific power sources will be controlled during specific status in FSM. So the low power mode is a “delay affect” way to handle power sources inside the RK3328 chip.

6.4 Register Description

6.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_PMU_WAKEUP_CFG0	0x0000	W	0x00000000	
PMU_PMU_PWRDN_CON	0x000c	W	0x00000000	
PMU_PMU_PWRDN_ST	0x0010	W	0x00000000	
PMU_PMU_PWRMODE_COMMON_CON	0x0018	W	0x00000000	
PMU_PMU_SFT_CON	0x001c	W	0x00000000	
PMU_PMU_INT_CON	0x0020	W	0x00000000	
PMU_PMU_INT_ST	0x0024	W	0x00000000	
PMU_PMU_POWER_ST	0x0044	W	0x00000000	
PMU_PMU_CPU0APM_CON	0x0080	W	0x00000000	
PMU_PMU_CPU1APM_CON	0x0084	W	0x00000000	
PMU_PMU_CPU2APM_CON	0x0088	W	0x00000000	
PMU_PMU_CPU3APM_CON	0x008c	W	0x00000000	
PMU_PMU_SYS_REG0	0x00a0	W	0x00000000	
PMU_PMU_SYS_REG1	0x00a4	W	0x00000000	
PMU_PMU_SYS_REG2	0x00a8	W	0x00000000	
PMU_PMU_SYS_REG3	0x00ac	W	0x00000000	

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

6.4.2 Detail Register Description

PMU_PMU_WAKEUP_CFG0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	wakeup_int_cluster_en interrupt wakeup enable 0: disable 1: enable

PMU_PMU_PWRDN_CON

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	pd_a53_3_pwrdwn_en a53 cpu3 power down enable 0: disable 1: enable
2	RW	0x0	pd_a53_2_pwrdwn_en a53 cpu2 power down enable 0: disable 1: enable
1	RW	0x0	pd_a53_1_pwrdwn_en a53 cpu1 power down enable 0: disable 1: enable
0	RW	0x0	pd_a53_0_pwrdwn_en a53 cpu0 power down enable 0: disable 1: enable

PMU_PMU_PWRDN_ST

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	pd_a53_3_pwr_stat CPU3 power status 0: power up 1: power down
2	RW	0x0	pd_a53_2_pwr_stat CPU2 power status 0: power up 1: power down
1	RW	0x0	pd_a53_1_pwr_stat CPU1 power status 0: power up 1: power down
0	RW	0x0	pd_a53_0_pwr_stat CPU0 power status 0: power up 1: power down

PMU_PMU_PWRMODE_COMMON_CON

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	ddrio_ret_en ddrio retention enable 0: disable 1: enable
7	RW	0x0	ddrio_ret_de_req ddrio retention de request 0: disable 1: enable
6	RW	0x0	l2_idle_en wait for L2 idle enable 0: disable 1: enable
5	RW	0x0	l2_flush_en flush L2 during power mode 0: disable 1: enable
4	RW	0x0	wait_wakeup_begin_cfg pmu start to observe for wakeup signals 0: disable 1: enable
3	RW	0x0	cpu0_pd_en power down cpu0 enable 0: disable 1: enable
2	RW	0x0	global_int_disable_cfg global interrupt disable configure 0: enable interrupt 1: disable interrupt
1	RW	0x0	sref_enter_en DDR enter self-refresh enable when in power mode 0: disable 1: enable
0	RW	0x0	power_mode_en enable FSM 0: disable 1: enable

PMU_PMU_SFT_CON
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	ddr_io_ret_cfg software request ddr retention 0: disable 1: enable
1	RW	0x0	l2flushreq_req software request l2 flush 0: disable 1: enable
0	RW	0x0	upctl_c_sysreq_cfg software request ddr self-refresh 0: disable 1: enalbe

PMU_PMU_INT_CON

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RW	0x0	a53_l3_pwr_switch_int_en a53 CPU3 power switch interrupt enable 0: disable 1: enable
17	RW	0x0	a53_l2_pwr_switch_int_en a53 CPU2 power switch interrupt enable 0: disable 1: enable
16	RW	0x0	a53_l1_pwr_switch_int_en a53 CPU1 power switch interrupt enable 0: disable 1: enable
15	RW	0x0	a53_l0_pwr_switch_int_en a53 CPU0 power switch interrupt enable 0: disable 1: enable
14:5	RO	0x0	reserved
4	RW	0x0	wakeup_int_en interrupt wakeup interrupt enable 0: disable 1: enable
3:2	RO	0x0	reserved
1	RW	0x0	pwrmode_wakeup_int_en power mode wakeup interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
0	RW	0x0	pmu_int_en pmu interrupt global enable 0: disable 1: enable

PMU_PMU_INT_ST

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	a53_l3_pwr_switch_status a53 cpu3 power switch interrupt status
4	RW	0x0	a53_l2_pwr_switch_status a53 cpu2 power switch interrupt status
3	RW	0x0	a53_l1_pwr_switch_status a53 cpu1 power switch interrupt status
2	RW	0x0	a53_l0_pwr_switch_status a53 cpu0 power switch interrupt status
1	RW	0x0	wakeup_int_status interrupt wakeup status
0	RW	0x0	pwrmode_wakeup_status power mode wakeup status

PMU_PMU_POWER_ST

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	pwr_status pmu power FSM value

PMU_PMU_CPU0APM_CON

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu0_sft_wakeup cpu0 software wakeup enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
2	RW	0x0	global_int_disable_0_cfg disable interrupt to cpu0 0: enable interrupt 1: disable interrupt
1	RW	0x0	cpu0_int_wakeup_en cpu0 interrupt wakeup enable 0: disable 1: enable
0	RW	0x0	cpu0_wfi_pwrdn_en cpu0 WFI power down enable 0: disable 1: enable

PMU_PMU_CPU1APM_CON
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu1_sft_wakeup cpu1 software wakeup enable 0: disable 1: enable
2	RW	0x0	global_int_disable_1_cfg disable interrupt to cpu1 0: enable interrupt 1: disable interrupt
1	RW	0x0	cpu1_int_wakeup_en cpu1 interrupt wakeup enable 0: disable 1: enable
0	RW	0x0	cpu1_wfi_pwrdn_en cpu1 WFI power down enable 0: disable 1: enable

PMU_PMU_CPU2APM_CON
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	cpu2_sft_wakeup cpu2 software wakeup enable 0: disable 1: enable
2	RW	0x0	global_int_disable_2_cfg disable interrupt to cpu2 0: enable interrupt 1: disable interrupt
1	RW	0x0	cpu2_int_wakeup_en cpu2 interrupt wakeup enable 0: disable 1: enable
0	RW	0x0	cpu2_wfi_pwrdn_en cpu2 WFI power down enable 0: disable 1: enable

PMU_PMU_CPU3APM_CON
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu3_sft_wakeup cpu3 software wakeup enable 0: disable 1: enable
2	RW	0x0	global_int_disable_3_cfg disable interrupt to cpu3 0: enable interrupt 1: disable interrupt
1	RW	0x0	cpu3_int_wakeup_en cpu3 interrupt wakeup enable 0: disable 1: enable
0	RW	0x0	cpu3_wfi_pwrdn_en cpu3 WFI power down enable 0: disable 1: enable

PMU_PMU_SYS_REG0
Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg0 system register 0

PMU_PMU_SYS_REG1
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg1 system register 1

PMU_PMU_SYS_REG2
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg2 system register 2

PMU_PMU_SYS_REG3
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg3 system register 3

6.5 Timing Diagram

6.5.1 Each domain power switch timing

The following figure is the each domain power down and power up timing.

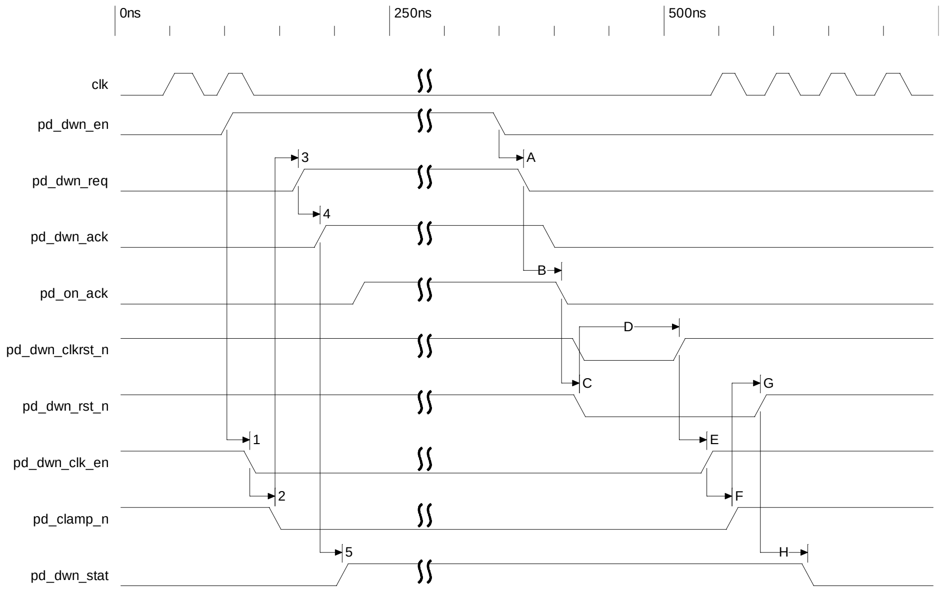


Fig. 4-5 Each Domain Power Switch Timing

6.5.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV, SIM detect wakeup, GPIO0 wakeup source and so on. All these external wakeup sources must meet the timing requirement (at least 200us) when the wakeup event is asserted. The following figure gives the timing information.

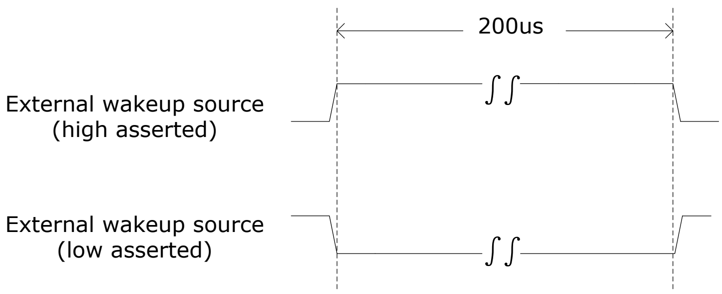


Fig. 4-6 External Wakeup Source PAD Timing

6.6 Application Note

6.6.1 Low power mode

PMU can work in the Low power mode by setting bit[0] of PMU_PWRMODE_CON register. After setting this bit and all CPU cores enters WFI states, PMU low power FSM will start to run. In the low power mode, PMU will manage power resources by hardware, such as power on/off the specified power domain, send idle request to specified power domain, shut down/up PLL and so on. All of above are configurable by setting corresponding registers. ALL FSM power states could be monitored through IO. The following table describes all power states of PMU FSM.

Table 4-4 Low Power State

Num	STATES	Description
0	ST_NORMAL	Still in normal state
1	ST_CPU0_PWRDN	Hold CPU0 in reset status, not really power down
2	ST_L2_FLUSH	Flush L2 by hardware
3	ST_L2_IDLE	Wait for L2 idle
4	ST_SREF_ENTER	Enter DDR self-refresh
5	ST_DDR_IO_RET	DDR IO retention
6	ST_WAIT_WAKEUP	Wait for wake up
7	ST_SREF_EXIT	Exit DDR self-refresh
8	ST_CPU0_PWRUP	De-assert reset for CPU0

Chapter 7 Generic Interrupt Controller (GIC)

7.1 Overview

There is a generic interrupt controller(GIC400) in RK3328 which generates physical interrupts to Cortex-A53. It has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A53. The details of CPU interface connectivity are shown in the following table.

Table 1-1 CPU interface connectivity

CPU Interface Number	Connectivity
CPU interface 0	CPU0
CPU interface 1	CPU1
CPU interface 2	CPU2
CPU interface 3	CPU3

It supports the following features:

- Supports 128 hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A53 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

7.2 Block Diagram

The generic interrupt controller comprises with:

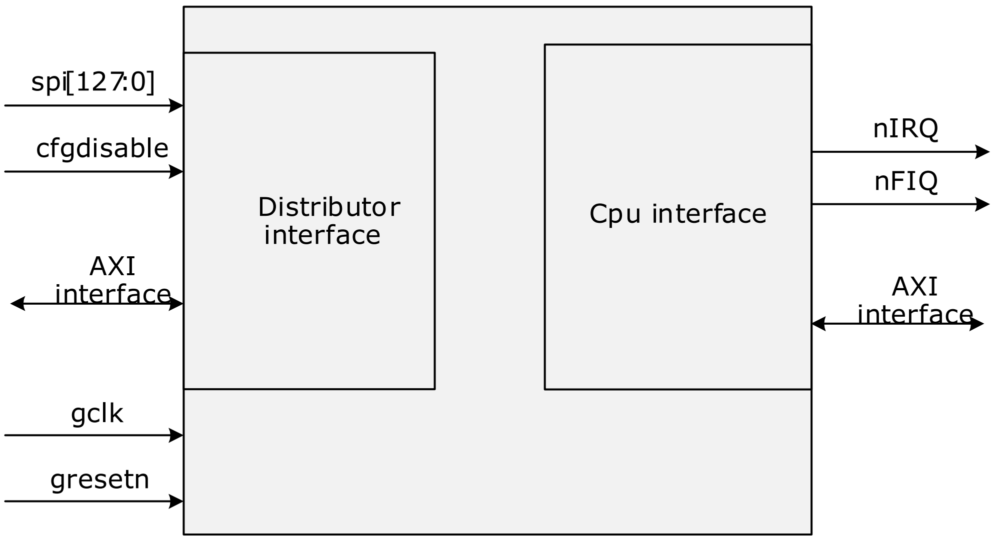


Fig. 7-1 Block Diagram

7.3 Function Description

Please refer to the document IHI0048B_gic_architecture_specification.pdf for the detail function description.

Chapter 8 DMA Controller (DMAC)

8.1 Overview

This device supports 1 Direct Memory Access (DMA) Controllers. It (DMAC) supports transfers between memory and memory, peripheral and memory. DMAC is under Non-secure state after reset, and the secure state can be changed by configuring SGRF module.
DMAC supports the following features:

- Supports Trustzone technology
- Supports 17 peripheral request
- Up to 64bits data size
- 8 channel at the same time
- Up to burst 16
- 16 interrupts output and 1 abort output
- Supports 128 MFIFO depth

Following table shows the DMAC request mapping scheme.

Table 8-1 DMAC Request Mapping Table

Req number	Source	Polarity
0	I2S2_2CH_TX	High level
1	I2S2_2CH_RX	High level
2	UART0_TX	High level
3	UART0_RX	High level
4	UART1_TX	High level
5	UART1_RX	High level
6	UART2_TX	High level
7	UART2_RX	High level
8	SPI0_TX	High level
9	SPI0_RX	High level
10	SPDIF_8CH_TX	High level
11	I2S0_8CH_TX	High level
12	I2S0_8CH_RX	High level
13	PWM_TX	High level
14	I2S1_8CH_TX	High level
15	I2S1_8CH_RX	High level
16	PDM_TX	High level

DMAC support incrementing-address burst and fixed-address burst. But in the case of access SPI and UART at byte or halfword size, DMAC only support fixed-address burst and the address must be aligned to word.

8.2 Block Diagram

Following figure shows the block diagram of DMAC.

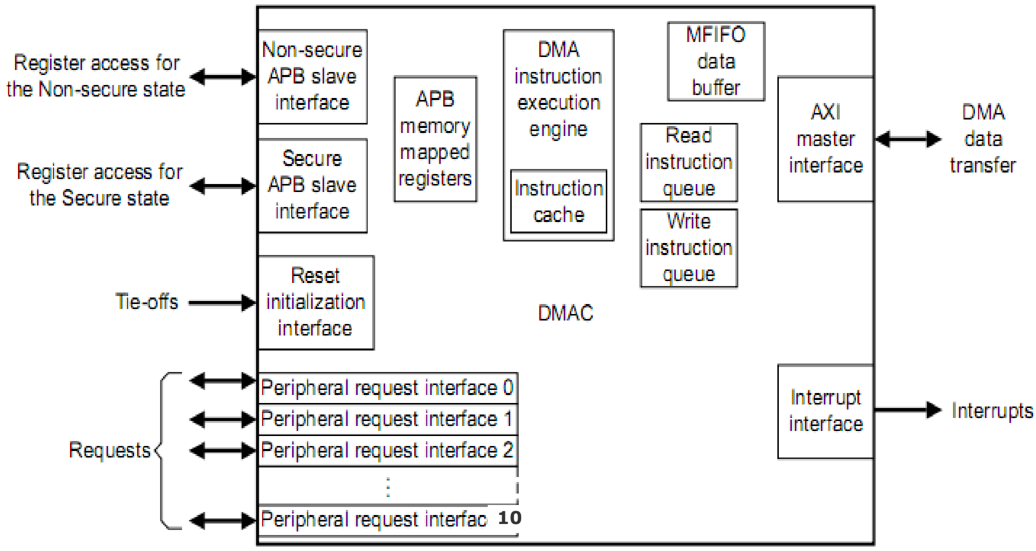


Fig. 8-1 Block diagram of DMAC

As the DMAC supports Trustzone technology, so dual APB interfaces enable the operation of the DMAC to be partitioned into the secure state and Non-secure state. You can use the APB interfaces to access status registers and also directly execute instructions in the DMAC. The default interface after reset is Non-secure apb interface.

8.3 Function Description

8.3.1 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.

8.3.2 Operating states

Following figure shows the operating states for the DMA manager thread and DMA channel threads.

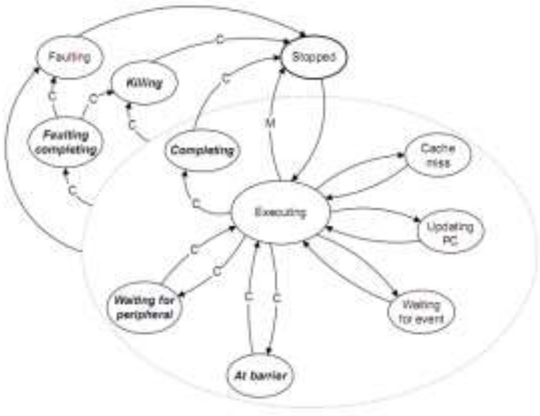


Fig. 8-2 DMAC operation states

Notes:arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:

C DMA channel threads only.

M DMA manager thread only.

After the DMAC exits from reset, it sets all DMA channel threads to the stopped state, and DMA manager thread moves to the Stopped state.

8.4 Register Description

8.4.1 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 Register
DMAC_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel Register

Name	Offset	Size	Reset Value	Description
DMAC_FTR6	0x0058	W	0x00000000	Fault Type DMA Channel Register
DMAC_FTR7	0x005c	W	0x00000000	Fault Type DMA Channel Register
DMAC_CSR0	0x0100	W	0x00000000	Channel Status Registers
DMAC_CPC0	0x0104	W	0x00000000	Channel Program Counter Registers
DMAC_CSR1	0x0108	W	0x00000000	Channel Status Registers
DMAC_CPC1	0x010c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR2	0x0110	W	0x00000000	Channel Status Registers
DMAC_CPC2	0x0114	W	0x00000000	Channel Program Counter Registers
DMAC_CSR3	0x0118	W	0x00000000	Channel Status Registers
DMAC_CPC3	0x011c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR4	0x0120	W	0x00000000	Channel Status Registers
DMAC_CPC4	0x0124	W	0x00000000	Channel Program Counter Registers
DMAC_CSR5	0x0128	W	0x00000000	Channel Status Registers
DMAC_CPC5	0x012c	W	0x00000000	Channel Program Counter Registers
DMAC_CSR6	0x0130	W	0x00000000	Channel Status Registers
DMAC_CPC6	0x0134	W	0x00000000	Channel Program Counter Registers
DMAC_CSR7	0x0138	W	0x00000000	Channel Status Registers
DMAC_CPC7	0x013c	W	0x00000000	Channel Program Counter Registers
DMAC_SAR0	0x0400	W	0x00000000	Source Address Registers
DMAC_DAR0	0x0404	W	0x00000000	Destination Address Registers
DMAC_CCR0	0x0408	W	0x00000000	Channel Control Registers
DMAC_LC0_0	0x040c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_0	0x0410	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR1	0x0420	W	0x00000000	Source Address Registers
DMAC_DAR1	0x0424	W	0x00000000	Destination Address Registers
DMAC_CCR1	0x0428	W	0x00000000	Channel Control Registers
DMAC_LC0_1	0x042c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_1	0x0430	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR2	0x0440	W	0x00000000	Source Address Registers
DMAC_DAR2	0x0444	W	0x00000000	Destination Address Registers
DMAC_CCR2	0x0448	W	0x00000000	Channel Control Registers
DMAC_LC0_2	0x044c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_2	0x0450	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR3	0x0460	W	0x00000000	Source Address Registers
DMAC_DAR3	0x0464	W	0x00000000	Destination Address Registers

Name	Offset	Size	Reset Value	Description
DMAC_CCR3	0x0468	W	0x00000000	Channel Control Registers
DMAC_LC0_3	0x046c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_3	0x0470	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR4	0x0480	W	0x00000000	Source Address Registers
DMAC_DAR4	0x0484	W	0x00000000	Destination Address Registers
DMAC_CCR4	0x0488	W	0x00000000	Channel Control Registers
DMAC_LC0_4	0x048c	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_4	0x0490	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR5	0x04a0	W	0x00000000	Source Address Registers
DMAC_DAR5	0x04a4	W	0x00000000	Destination Address Registers
DMAC_CCR5	0x04a8	W	0x00000000	Channel Control Registers
DMAC_LC0_5	0x04ac	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_5	0x04b0	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR6	0x04c0	W	0x00000000	Source Address Registers
DMAC_DAR6	0x04c4	W	0x00000000	Destination Address Registers
DMAC_CCR6	0x04c8	W	0x00000000	Channel Control Registers
DMAC_LC0_6	0x04cc	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_6	0x04d0	W	0x00000000	Loop Counter 1 Registers
DMAC_SAR7	0x04e0	W	0x00000000	Source Address Registers
DMAC_DAR7	0x04e4	W	0x00000000	Destination Address Registers
DMAC_CCR7	0x04e8	W	0x00000000	Channel Control Registers
DMAC_LC0_7	0x04ec	W	0x00000000	Loop Counter 0 Registers
DMAC_LC1_7	0x04f0	W	0x00000000	Loop Counter 1 Registers
DMAC_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMAC_DBGCMD	0x0d04	W	0x00000000	Debug Command Register
DMAC_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMAC_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMAC_CR0	0x0e00	W	0x00047051	Configuration Register 0
DMAC_CR1	0x0e04	W	0x00000057	Configuration Register 1
DMAC_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMAC_CR3	0x0e0c	W	0x00000000	Configuration Register 3
DMAC_CR4	0x0e10	W	0x00000006	Configuration Register 4
DMAC_CRDn	0x0e14	W	0x02094733	DMA 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.*
For DMAC0 channel register, only the channel 0~5 is valid.

8.4.2 Detail Register Description

DMAC_DSR

Address: Operational Base + offset (0x0000)

DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RO	0x0	Provides the security status of the DMA manager thread: 0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state.
8:4	RO	0x00	When the DMA manager thread executes a DMAWFE instruction, it waits for the following event to occur: b00000 = event[0] b00001 = event[1] b00010 = event[2] … b11111 = event[31].
3:0	RO	0x0	The operating state of the DMA manager: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101-b1110 = reserved b1111 = Faulting.

DMAC_DPC

Address: Operational Base + offset (0x0004)

DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA manager thread

DMAC_INTEN

Address: Operational Base + offset (0x0020)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Program the appropriate bit to control how the DMAC responds when it executes DMASEV: Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request. Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interruptrequest.

DMAC_EVENT_RIS

Address: Operational Base + offset (0x0024)

Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Returns the status of the event-interrupt resources: Bit [N] = 0 Event N is inactive or irq[N] is LOW. Bit [N] = 1 Event N is active or irq[N] is HIGH.

DMAC_INTMIS

Address: Operational Base + offset (0x0028)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the status of the interrupts that are active in the DMAC: Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW. Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH

DMAC_INTCLR

Address: Operational Base + offset (0x002c)

Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	Controls the clearing of the irq outputs: Bit [N] = 0 The status of irq[N] does not change. Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change.

DMAC_FSRD

Address: Operational Base + offset (0x0030)

Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the fault status of the DMA manager. Read as: 0 = the DMA manager thread is not in the Faulting state 1 = the DMA manager thread is in the Faulting state.

DMAC_FSRC

Address: Operational Base + offset (0x0034)

Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Each bit provides the fault status of the corresponding channel. Read as: Bit [N] = 0 No fault is present on DMA channel N. Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state.

DMAC_FTRD

Address: Operational Base + offset (0x0038)

Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	If the DMA manager aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	RO	0x0	reserved
16	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	Indicates if the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAWFE or DMASEV 1 = a DMA manager thread in the Non-secure state attempted to execute either: DMAWFE to wait for a secure event DMASEV to create a secure event or secure interrupt
4	RO	0x0	Indicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = 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	Indicates if the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand.
0	RW	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC_FTR0~DMAC_FTR7

Address: Operational Base + offset (0x0040)
Operational Base+0x44
Operational Base+0x48
Operational Base+0x4C
Operational Base+0x50
Operational Base+0x54
Operational Base+0x58
Operational Base+0x5C

Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	RO	0x0	Indicates if the DMA channel has locked-up because of resource starvation: 0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	RO	0x0	If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	RO	0x0	Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort.
16	RO	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RO	0x0	Indicates if the MFIFO did not contain the data to enable the DMAC to perform the DMAST: 0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.
12	RO	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMALD or DMAST. Depending on the instruction: DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	RO	0x0	reserved
7	RO	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to program the CCRn Register to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	RO	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	RO	0x0	Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: DMAWFE to wait for a secure event 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	Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort.
0	RO	0x0	Indicates if the DMA channel thread was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMAC_CSR0~DMAC_CSR7

Address: Operational Base+0x100
Operational Base+0x108
Operational Base+0x110
Operational Base+0x118
Operational Base+0x120
Operational Base+0x128
Operational Base+0x130
Operational Base+0x138

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	The channel non-secure bit provides the security of the DMA channel: 0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the periph operand was set: 0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	RO	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the burst or single operand were set: 0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set.
13:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:4	RO	0x00	If the DMA channel is in the Waiting for event state or the Waiting for peripheral state then these bits indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 … b11111 = DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	The channel status encoding is: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101 = At barrier b0110 = reserved b0111 = Waiting for peripheral b1000 = Killing b1001 = Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMAC_CPC0~DMAC_CPC7

Address: Operational Base+0x104
Operational Base+0x10C
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12C
Operational Base+0x134
Operational Base+0x13C

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Program counter for the DMA channel 0 thread

DMAC_SAR0~DMAC_SAR7

Address: Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4A0
Operational Base+0x4C0
Operational Base+0x4E0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the source data for DMA channel 0

DMAC_DAR0~DMAC_DAR7

Address: Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4A4
Operational Base+0x4C4
Operational Base+0x4E4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Address of the Destination data for DMA channel 0

DMAC_CCR0~DMAC_CCR7

Address: Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4A8
Operational Base+0x4C8
Operational Base+0x4E8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data. Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	RO	0x0	Programs the state of AWPROT[2:0]a when the DMAC writes the destination data. Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH

Bit	Attr	Reset Value	Description
21:18	RO	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers … b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size
17:15	RO	0x0	For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination: b000 = writes 1 byte per beat b001 = writes 2 bytes per beat b010 = writes 4 bytes per beat b011 = writes 8 bytes per beat b100 = writes 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	Programs the burst type that the DMAC performs when it writes the destination data: 0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data. Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.
10:8	RO	0x0	Programs the state of ARPROT[2:0]a when the DMAC reads the source data. Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.

Bit	Attr	Reset Value	Description
7:4	RO	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers … b1111 = 16 data transfers. The total number of bytes that the DMAC 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	For each beat within a burst, it programs the number of bytes that the DMAC reads from the source: b000 = reads 1 byte per beat b001 = reads 2 bytes per beat b010 = reads 4 bytes per beat b011 = reads 8 bytes per beat b100 = reads 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
0	RO	0x0	Programs the burst type that the DMAC performs when it reads the source data: 0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMAC_LC0_0~DMAC_LC0_7

Address:Operational Base+0x40c
Operational Base+0x42C
Operational Base+0x44C
Operational Base+0x46C
Operational Base+0x48C
Operational Base+0x4AC
Operational Base+0x4CC
Operational Base+0x4EC

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 0 iterations

DMAC_LC1_0~DMAC_LC1_7

Address:Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470

Operational Base+0x490
Operational Base+0x4B0
Operational Base+0x4D0
Operational Base+0x4F0

Loop Counter 1 Registers

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	Loop counter 1 iterations

DMAC_DBGSTATUS

Address: Operational Base + offset (0x0d00)

Debug Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RO	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved.

DMAC_DBGCMD

Address: Operational Base + offset (0x0d04)

Debug Command Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	WO	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved

DMAC_DBGINST0

Address: Operational Base + offset (0x0d08)

Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 1
23:16	WO	0x00	Instruction byte 0
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	WO	0x0	DMA channel number: b000 = DMA channel 0 b001 = DMA channel 1 b010 = DMA channel 2 … b111 = DMA channel 7
7:1	RO	0x0	reserved
0	WO	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC_DBGINST1

Address: Operational Base + offset (0x0d0c)

Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	WO	0x00	Instruction byte 5
23:16	WO	0x00	Instruction byte 4
15:8	WO	0x00	Instruction byte 3
7:0	WO	0x00	Instruction byte 2

DMAC_CR0

Address: Operational Base + offset (0x0e00)

Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:17	RO	0x02	Number of interrupt outputs that the DMAC provides: b00000 = 1 interrupt output, irq[0] b00001 = 2 interrupt outputs, irq[1:0] b00010 = 3 interrupt outputs, irq[2:0] … b11111 = 32 interrupt outputs, irq[31:0].
16:12	RO	0x07	Number of peripheral request interfaces that the DMAC provides: b00000 = 1 peripheral request interface b00001 = 2 peripheral request interfaces b00010 = 3 peripheral request interfaces … b11111 = 32 peripheral request interfaces.
11:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RO	0x5	Number of DMA channels that the DMAC supports: b000 = 1 DMA channel b001 = 2 DMA channels b010 = 3 DMA channels … b111 = 8 DMA channels.
3	RO	0x0	reserved
2	RO	0x0	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset: 0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH.
1	RO	0x0	Indicates the status of the boot_from_pc signal when the DMAC exited from reset: 0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH
0	RO	0x1	Supports peripheral requests: 0 = the DMAC does not provide a peripheral request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.

DMAC_CR1
Address: Operational Base + offset (0x0e04)
Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RO	0x5	[7:4] num_i-cache_lines Number of i-cache lines: b0000 = 1 i-cache line b0001 = 2 i-cache lines b0010 = 3 i-cache lines … b1111 = 16 i-cache lines.
3	RO	0x0	reserved
2:0	RO	0x7	The length of an i-cache line: b000-b001 = reserved b010 = 4 bytes b011 = 8 bytes b100 = 16 bytes b101 = 32 bytes b110-b111 = reserved

DMAC_CR2
Address: Operational Base + offset (0x0e08)
Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_CR3

Address: Operational Base + offset (0x0e0c)

Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	Provides the security state of an event-interrupt resource: Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state.

DMAC_CR4

Address: Operational Base + offset (0x0e10)

Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000006	Provides the security state of the peripheral request interfaces: Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMAC_CRDn

Address: Operational Base + offset (0x0e14)

DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RO	0x020	The number of lines that the data buffer contains: b000000000 = 1 line b000000001 = 2 lines … b111111111 = 1024 lines
19:16	RO	0x9	The depth of the read queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14:12	RO	0x4	Read issuing capability that programs the number of outstanding read transactions: b000 = 1 b001 = 2 … b111 = 8
11:8	RO	0x7	The depth of the write queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
7	RO	0x0	reserved
6:4	RO	0x3	Write issuing capability that programs the number of outstanding write transactions: b000 = 1 b001 = 2 … b111 = 8
3	RO	0x0	reserved
2:0	RO	0x3	The data bus width of the AXI interface: b000 = reserved b001 = reserved b010 = 32-bit b011 = 64-bit b100 = 128-bit b101-b111 = reserved.

DMAC_WD
Address: Operational Base + offset (0x0e80)
DMA Watchdog Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	Controls how the DMAC responds when it detects a lock-up condition: 0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH.

8.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

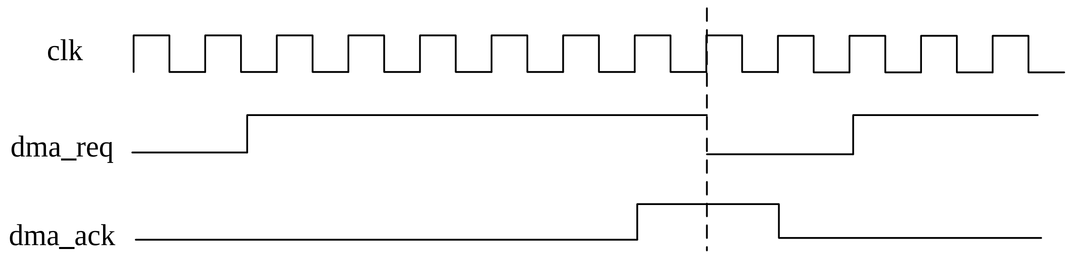


Fig. 8-3 DMAC request and acknowledge timing

8.6 Interface Description

DMAC has the following tie-off signals. It can be configured by SGRF register. (Please refer to the chapter to find how to configure)

Table 8-2 DMAC boot interface

Interface	Reset value	Control source
boot_manager_ns	0x0	sgrf_dmac1_con5[15]
boot_irq_ns	0x0	sgrf_dmac1_con3[15:0]
boot_periph_ns	0x0	{sgrf_dmac1_con5[3:0],sgrf_dmac1_con4[15:0]}
grf_drtype_uart0_tx	0x1	sgrf_dmac1_con0[1:0]
grf_drtype_uart0_rx	0x1	sgrf_dmac1_con0[3:2]
grf_drtype_uart1_tx	0x1	sgrf_dmac1_con0[5:4]
grf_drtype_uart1_rx	0x1	sgrf_dmac1_con0[7:6]
grf_drtype_uart2_tx	0x1	sgrf_dmac1_con0[9:8]
grf_drtype_uart2_rx	0x1	sgrf_dmac1_con0[11:10]
grf_drtype_spi0_tx	0x1	sgrf_dmac1_con0[13:12]
grf_drtype_spi0_rx	0x1	sgrf_dmac1_con0[15:14]
grf_drtype_i2s0_8ch_tx	0x1	sgrf_dmac1_con1[1:0]
grf_drtype_i2s0_8ch_rx	0x1	sgrf_dmac1_con1[3:2]
grf_drtype_i2s1_8ch_tx	0x1	sgrf_dmac1_con1[5:4]
grf_drtype_i2s1_8ch_rx	0x1	sgrf_dmac1_con1[7:6]
grf_drtype_i2s2_2ch_tx	0x1	sgrf_dmac1_con1[9:8]
grf_drtype_i2s2_2ch_rx	0x1	sgrf_dmac1_con1[11:10]
grf_drtype_spdif	0x1	sgrf_dmac1_con1[13:12]
grf_drtype_pwm	0x1	sgrf_dmac1_con1[15:14]
grf_drtype_pdm	0x1	sgrf_dmac1_con2[1:0]

boot_manager_ns

When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:

0 = assigns DMA manager to the Secure state

1 = assigns DMA manager to the Non-secure state.

boot_irq_ns

Controls the security state of an event-interrupt resource, when the DMAC exits from reset:

boot_irq_ns[x] is LOW

The DMAC assigns event<x> or irq[x] to the Secure state.

boot_irq_ns[x] is HIGH

The DMAC assigns event<x> or irq[x] to the Non-secure state.

boot_periph_ns

Controls the security state of a peripheral request interface, when the DMAC exits from reset:

boot_periph_ns[x] is LOW

The DMAC assigns peripheral request interface x to the Secure state.

boot_periph_ns[x] is HIGH

The DMAC assigns peripheral request interface x to the Non-secure state.

grf_drtype_<x>

The DMAC sets the state of the request_type flag:

grf_drtype_<x>[1:0]=b00: request_type<x> = Single.

grf_drtype_<x>[1:0]=b01: request_type<x> = Burst.

8.7 Application Notes

8.7.1 Using the APB slave interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.

The necessary steps to start a DMA channel thread using the debug instruction registers as following:

1. Create a program for the DMA channel.
2. Store the program in a region of system memory.
3. Poll the DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.
4. Write to the DBGINST0 Register and enter the:
   - Instruction byte 0 encoding for DMAGO.
   - Instruction byte 1 encoding for DMAGO.
   - Debug thread bit to 0. This selects the DMA manager thread.
5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program, that was written to system memory in step 2.
6. Writing zero to the DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

8.7.2 Security usage

DMA manager thread is in the secure state

If the DNS bit is 0, the DMA manager thread operates in the secure state and it only performs secure instruction fetches. When a DMA manager thread in the secure state processes:

DMAGO

It uses the status of the ns bit, to set the security state of the DMA channel thread by writing to the CNS bit for that channel.

DMAWFE

It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.

DMASEV

It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.

DMA manager thread is in the Non-secure state

If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:

DMAGO

The DMAC uses the status of the ns bit, to control if it starts a DMA channel thread. If:

ns = 0

The DMAC does not start a DMA channel thread and instead it:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

ns = 1

The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.

DMAWFE

The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0

The event is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1

The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV

The DMAC uses the status of the corresponding INS bit, in the CR3Register, to control if it creates the event-interrupt. If:

INS = 0

The event-interrupt resource is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1

The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMA channel thread is in the secure state

When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.
When a DMA channel thread in the secure state processes the following instructions:
DMAWFE
The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.
DMASEV
The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.
DMAWFP
The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
DMALDP, DMASTP
The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
DMAFLUSHP
The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.
When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses
DMA channel thread is in the Non-secure state
When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.
When a DMA channel thread in the Non-secure state processes the following instructions:
DMAWFE
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:
INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.
DMASEV
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event. If:
INS = 0
The event-interrupt resource is in the Secure state. The DMAC:
1. Executes a NOP.

2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers .
4. Moves the DMA channel to the Faulting completing state.

INS = 1

The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMAWFP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it waits for the peripheral to signal a request. If:

PNS = 0

The peripheral is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.

DMALDP, DMASTP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends an acknowledgement to the peripheral. If:

PNS = 0

The peripheral is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC sends a message to the peripheral to communicate when the data transfer is complete.

DMAFLUSHP

The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends a flush request to the peripheral. If:

PNS = 0

The peripheral is in the secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1

The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.

When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:
1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

8.7.3 Programming restrictions

Fixed unaligned bursts

The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:

Unaligned read

- src_inc field is 0 in the CCRn Register
- the SARn Register contains an address that is not aligned to the size of data that the src_burst_size field contain

Unaligned write

- dst_inc field is 0 in the CCRn Register
- the DARn Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions

If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.

Updating DMA channel control registers during a DMA cycle restrictions

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.

Resource sharing between DMA channels

DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

8.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.

Data corruption might occur if all of the following are true:
1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is splited across two lines in the data buffer (see Splitting data, below).
3. There is one idle cycle between the two read data beats.
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface need to be splited across two lines in the internal data buffer. This occurs when the read data beat contains data bytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.

Most applications of DMA-330 do not split data in this way, so are NOT vulnerable to data corruption from this defect.

The following cases are NOT vulnerable to data corruption because they do not split data:

- Byte lane offset between source and destination addresses is 0 when source and destination addresses have the same byte lane alignment, the offset is 0 and a wrap operation that splits data cannot occur.
- Byte lane offset between source and destination addresses is a multiple of source size

Table 8-3 Source size in CCRn

Source size in CCRn	Allowed offset between SARn and DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

8.7.5 Interrupt shares between channel

As the DMAC does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help identify the interrupt source.

There are 4 likely information sources that can be used to determine the progress made by a program:

- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.

The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:

1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

8.7.6 Instruction sets

Table 8-4 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

8.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled. For the other instructions, please refer to pl330_trm.pdf.

DMAADNH

Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMAC to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Registers and Destination Address Registers. The immediate unsigned 16-bit value is one-extended to 32 bits, to create a value that is the two’s complement representation of a negative number between -65536 and -1, before the DMAC adds it to the address using 32-bit addition. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the current value in the Source or Destination Address Register. Following table shows the instruction encoding.

Table 8-5 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

SARn Register and sets ra to 0.

DAR

DARn Register and sets ra to 1.
<16-bit immediate>
The immediate value to be added to the <address_register>.
You should specify the 16-bit immediate as the number that is to be represented in the instruction encoding. For example, DMAADNH DAR, 0xFFF0 causes the value 0xFFFFFFF0 to be added to the current value of the Destination Address Register, effectively subtracting 16 from the DAR.
You can only use this instruction in a DMA channel thread.

Chapter 9 Temperature Sensor ADC (TSADC)

9.1 Overview

TS-ADC Controller module supports user-defined mode and automatic mode. User-defined mode refers, TSADC all the control signals entirely by software writing to register for direct control. Automatic mode refers to the module automatically poll TSADC output, and the results were checked. If you find that the temperature 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.

TS-ADC Controller supports the following features:

- Support User-Defined Mode and Automatic Mode
- In User-Defined Mode, start_of_conversion can be controlled completely by software, and also can be generated by hardware.
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 1 channel TS-ADC, the temperature criteria can be configurable
- In Automatic Mode, the time interval of temperature detection can be configurable
- In Automatic Mode, when detecting a high temperature, the time interval of temperature detection can be configurable
- High temperature denounce can be configurable
- -40~125°C temperature range and 5°C temperature resolution
- 10-bit SARADC up to 50KS/s sampling rate

9.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

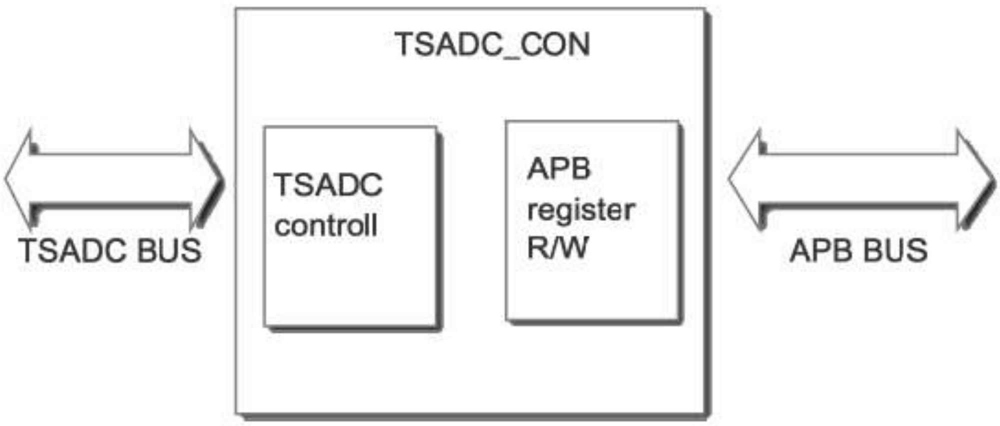


Fig. 9-1 TS-ADC Controller Block Diagram

9.3 Function Description

9.3.1 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.

9.3.2 TS-ADC Controller

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block. This block compares the analog input with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

9.4 Register description

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000200	The control register of A/D Converter.
TSADC_AUTO_CON	0x0004	W	0x00000000	TSADC auto mode control register
TSADC_INT_EN	0x0008	W	0x00000000	
TSADC_INT_PD	0x000c	W	0x00000000	
TSADC_DATA0	0x0020	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_DATA1	0x0024	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_COMP0_INT	0x0030	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMP1_INT	0x0034	W	0x00000000	TSADC high temperature level for source 1
TSADC_COMP0_SHUT	0x0040	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMP1_SHUT	0x0044	W	0x00000000	TSADC high temperature level for source 1
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	high temperature debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	high temperature debounce
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	TSADC auto access period
TSADC_AUTO_PERIOD_HT	0x006c	W	0x00010000	TSADC auto access period when temperature is high
TSADC_COMP0_LOW_INT	0x0080	W	0x00000000	TSADC low temperature level for source 0
TSADC_COMP1_LOW_INT	0x0084	W	0x00000000	TSADC low temperature level for source 1

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

9.4.2 Detail Register Description

TSADC_USER_CON

Address: Operational Base + offset (0x0000)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress
11:6	RW	0x08	inter_pd_soc interleave between power down and start of conversion
5	RW	0x0	start When software write 1 to this bit , start_of_conversion will be assert. This bit will be cleared after TSADC access finishing. When TSADC_USER_CON[4] = 1'b1 take effect.
4	RW	0x0	start_mode start mode. 0: tsadc controller will asert start_of_conversion after "inter_pd_soc" cycles. 1: the start_of_conversion will be controlled by TSADC_USER_CON[5].
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down 1: ADC power up and reset
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) Others : Reserved

TSADC_AUTO_CON

Address: Operational Base + offset (0x0004)

TSADC auto mode control register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	last_tshut_2cru last_tshut_2cru for cru first/second reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.

Bit	Attr	Reset Value	Description
24	RW	0x0	last_tshut_2gpio last_tshut_2gpio for hardware reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
23:18	RO	0x0	reserved
17	RO	0x0	sample_dly_sel 0: AUTO_PERIOD is used 1: AUTO_PERIOD_HT is used
16	RO	0x0	auto_status 0: auto mode stop; 1: auto mode in progress.
15:14	RO	0x0	reserved
13	RW	0x0	src1_lt_en 0: do not care low temperature of source 0 1: enable the low temperature monitor of source 0
12	RW	0x0	src0_lt_en 0: do not care low temperature of source 0 1: enable the low temperature monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_prolarity 0: low active 1: high active
7:6	RO	0x0	reserved
5	RW	0x0	src1_en 0: do not care the temperature of source 1 1: if the temperature of source 0 is too high , TSHUT will be valid
4	RW	0x0	src0_en 0: do not care the temperature of source 0 1: if the temperature of source 0 is too high , TSHUT will be valid
3:2	RO	0x0	reserved
1	RW	0x0	tsadc_q_sel temperature coefficient 1'b0:use tsadc_q as output(positive temperature coefficient) 1'b1:use(1024 - tsadc_q) as output (negative temperature coefficient) RK3328 is negative temprature coefficient, so please set this bit as 1'b1
0	RW	0x0	auto_en 0: TSADC controller works at user-define mode 1: TSADC controller works at auto mode

TSADC_INT_EN

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_en eoc_Interrupt enable. eoc_interrupt enable in user defined mode 0: disable 1: enable
15:14	RO	0x0	reserved
13	RW	0x0	lt_inten_src1 low temperature interrupt enable for src1 0: disable 1: enable
12	RW	0x0	lt_inten_src0 low temperature interrupt enable for src0 0: disable 1: enable
11:10	RO	0x0	reserved
9	RW	0x0	tshut_2cru_en_src1 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.
8	RW	0x0	tshut_2cru_en_src0 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.
7:6	RO	0x0	reserved
5	RW	0x0	tshut_2gpio_en_src1 0: TSHUT output to gpio disabled. TSHUT output will always keep low . 1: TSHUT output works.
4	RW	0x0	tshut_2gpio_en_src0 0: TSHUT output to gpio disabled. TSHUT output will always keep low . 1: TSHUT output works.
3:2	RO	0x0	reserved
1	RW	0x0	ht_inten_src1 high temperature interrupt enable for src1 0: disable 1: enable
0	RW	0x0	ht_inten_src0 high temperature interrupt enable for src0 0: disable 1: enable

TSADC_INT_PD

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_pd Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
15:14	RO	0x0	reserved
13	RW	0x0	lt_irq_src1 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared.
12	RW	0x0	lt_irq_src0 When TSADC output is lower than COMP_INT_LOW, this bit will be valid, which means temperature is low, and the application should in charge of this. write 1 to it , this bit will be cleared.
11:6	RO	0x0	reserved
5	RW	0x0	tshut_o_src1 TSHUT output status When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared.
4	RW	0x0	tshut_o_src0 TSHUT output status When TSADC output is bigger than COMP_SHUT, this bit will be valid, which means temperature is VERY high, and the application should in charge of this. write 1 to it , this bit will be cleared.
3:2	RO	0x0	reserved
1	RW	0x0	ht_irq_src1 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared.
0	RW	0x0	ht_irq_src0 When TSADC output is bigger than COMP_INT, this bit will be valid, which means temperature is high, and the application should in charge of this. write 1 to it , this bit will be cleared.

TSADC_DATA0

Address: Operational Base + offset (0x0020)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[11:0]).

TSADC_DATA1
Address: Operational Base + offset (0x0024)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[11:0]).

TSADC_COMP0_INT
Address: Operational Base + offset (0x0030)
TSADC high temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP1_INT
Address: Operational Base + offset (0x0034)
TSADC high temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP0_SHUT
Address: Operational Base + offset (0x0040)
TSADC high temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_COMP1_SHUT

Address: Operational Base + offset (0x0044)

TSADC high temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is bigger than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_HIGHT_INT_DEBOUNCE

Address: Operational Base + offset (0x0060)

high temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_INT for "debounce" times.

TSADC_HIGHT_TSHUT_DEBOUNCE

Address: Operational Base + offset (0x0064)

high temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_SHUT for "debounce" times.

TSADC_AUTO_PERIOD

Address: Operational Base + offset (0x0068)

TSADC auto access period

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period when auto mode is enabled, this register controls the interleave between every two accessing of TSADC.

TSADC_AUTO_PERIOD_HT

Address: Operational Base + offset (0x006c)

TSADC auto access period when temperature is high

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT

TSADC_COMP0_LOW_INT

Address: Operational Base + offset (0x0080)

TSADC low temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

TSADC_COMP1_LOW_INT

Address: Operational Base + offset (0x0084)

TSADC low temperature level for source 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src1 TSADC low temperature level. TSADC output is lower than tsadc_comp, means the temperature is low. TSADC_LOW_INT will be valid.

9.5 Application Notes

9.5.1 Single-sample conversion

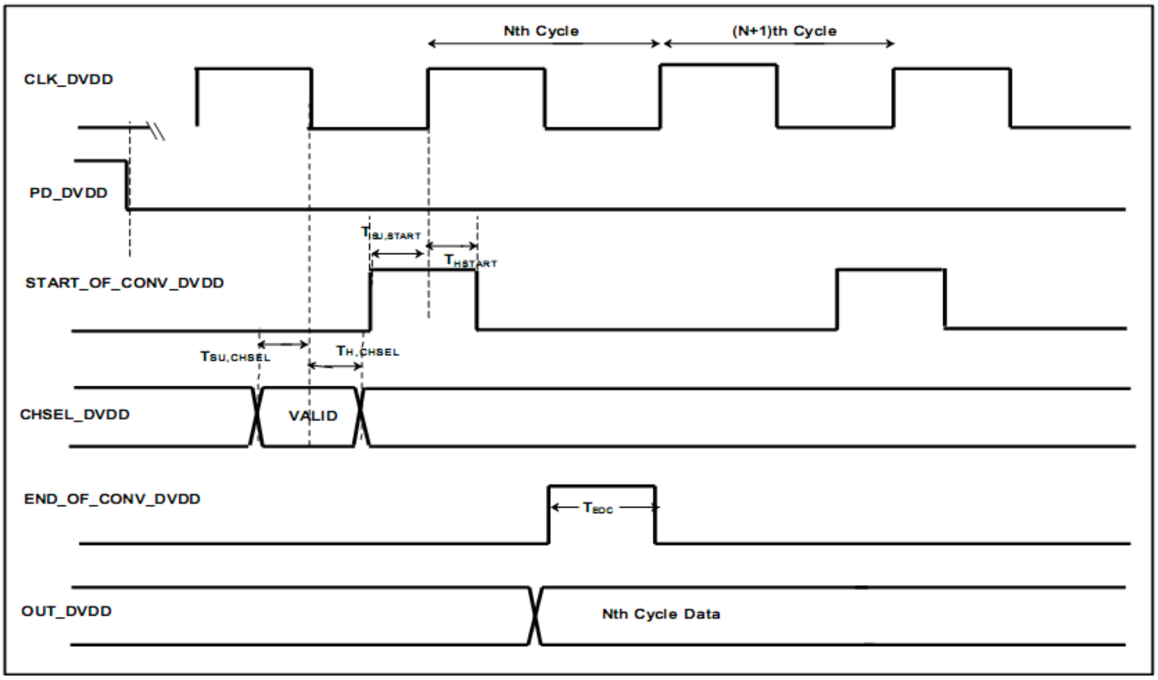


Fig. 9-2 the start flow to enable the sensor and adc

9.5.2 Temperature-to-code mapping

Table 9-1 Temperature Code Mapping

temp (C)	Code
-40	3800
-35	3792
-30	3783
-25	3774
-20	3765
-15	3756
-10	3747
-5	3737
0	3728
5	3718
10	3708
15	3698
20	3688
25	3678
30	3667
35	3656
40	3645
45	3634
50	3623
55	3611
60	3600
65	3588
70	3575
75	3563
80	3550
85	3537
90	3524
95	3510
100	3496
105	3482
110	3467
115	3452
120	3437
125	3421

Note:

Code to Temperature mapping of the Temperature sensor is a piece wise linear curve. Any temperature, code faling between to 2 give temperatures can be linearly interpolated.

Code to Temperature mapping should be updated based on sillcon results.

9.5.3 User-Define Mode

- In user-define mode, the PD_DVDD and CHSEL_DVDD are generate by setting register TSADC_USER_CON, bit[3] and bit[2:0]. In order to ensure timing between PD_DVDD and CHSEL_DVDD, the CHSEL_DVDD must be set before the PD_DVDD.
- In user-define mode, you can choose the method to control the START_OF_CONVERSION by setting bit[4] of TSADC_USER_CON. If set to 0, the start_of_conversion will be assert after "inter_pd_soc" cycles, which could be set by bit[11:6] of TSADC_USER_CON. And

if start_mode was set 1, the start_of_conversion will be controlled by bit[5] of TSADC_USER_CON.

- Software can get the four channel temperature from TSADC_DATAn (n=0,1,2,3).

9.5.4 Automatic Mode

You can use the automatic mode with the following step:

- Set TSADC_AUTO_PERIOD，configure the interleave between every two accessing of TSADC in normal operation.
- Set TSADC_AUTO_PERIOD_HT. configure the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.
- Set TSADC_COMPn_INT(n=0,1), configure the high temperature level, if tsadc output is smaller than the value, means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0,1), configure the super high temperature level, if tsadc output is smaller than the value, means the temperature is too high, TSHUT will be asserted.
- Set TSADC_INT_EN, you can enable the high temperature interrupt for all channel; and you can also set TSHUT output to gpio to reset the whole chip; and you can set TSHUT output to cru to reset the whole chip.
- Set TSADC_HIGHT_INT_DEBOUNCE and TSADC_HIGHT_TSHUT_DEBOUNCE, if the temperature is higher than COMP_INT or COMP_SHUT for “debounce” times, TSADC controller will generate interrupt or TSHUT.
- Set TSADC_AUTO_CON, enable the TSADC controller.

Chapter 10 SARADC

10.1 Overview

The ADC is a 6-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 13MHz A/D converter clock.

10.2 Block Diagram

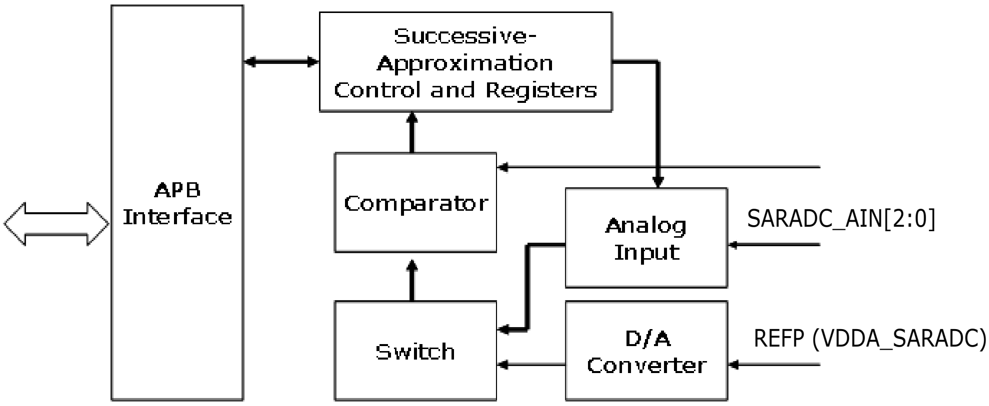


Fig. 10-1 SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

10.3 Function Description

10.3.1 APB Interface

In RK3328, SAR-ADC works at single-sample operation mode.

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

10.4 Register description

10.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.

Name	Offset	Size	Reset Value	Description
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

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

10.4.2 Detail Register Description

SARADC_DATA

Address: Operational Base + offset (0x0000)

This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS

Address: Operational Base + offset (0x0004)

The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

SARADC_CTRL

Address: Operational Base + offset (0x0008)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 0: Disable 1: Enable
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down; 1: ADC power up and reset. start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000 : Input source 0 (SARADC_AIN[0]) 001 : Input source 1 (SARADC_AIN[1]) 010 : Input source 2 (SARADC_AIN[2]) 011 : Input source 3 (SARADC_AIN[3]) 100 : Input source 4 (SARADC_AIN[4]) 101 : Input source 5 (SARADC_AIN[5]) Others : Reserved

SARADC_DLY_PU_SOC

Address: Operational Base + offset (0x000c)

delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

10.5 Timing Diagram

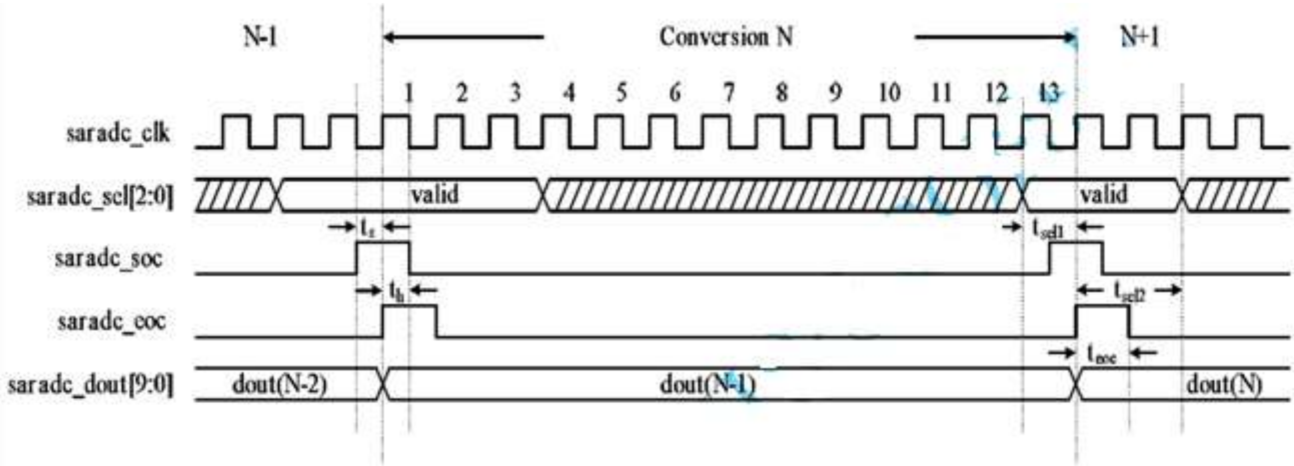


Fig. 10-2 SAR-ADC timing diagram in single-sample conversion mode

10.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]

Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 11 System Debug

11.1 Overview

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

11.1.1 Features

- Invasive debug with core halted
- SW-DP

11.1.2 Debug components address map

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

Module	Base Address
DAP_ROM	0xff800000

11.2 Block Diagram

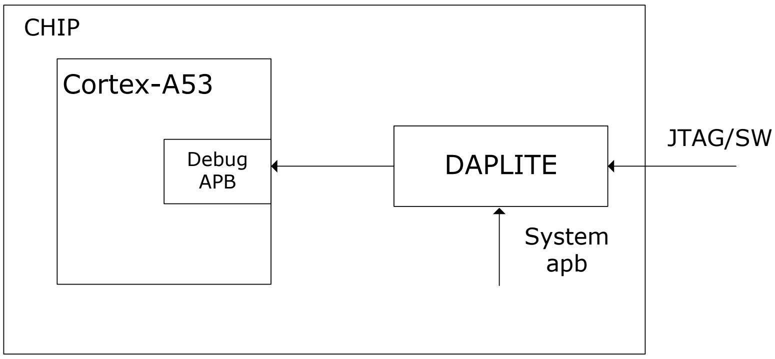


Fig. 11-1 Debug system structure

11.3 Function Description

11.3.1 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.

11.4 Register Description

Please refer to the documentCoreSight_DAPLite_TRM.pdf for the debug detail description.

11.5 Interface Description

11.5.1 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.

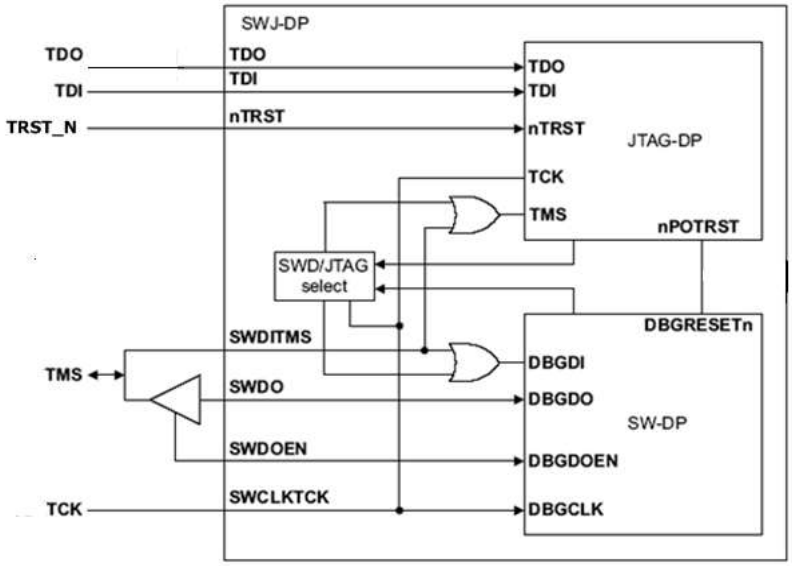


Fig. 11-2 DAP SWJ interface

11.5.2 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.

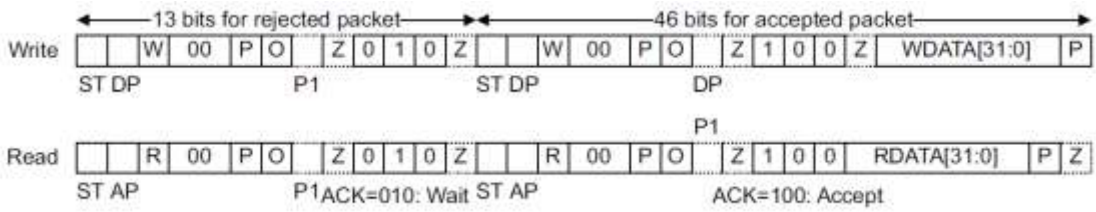


Fig. 11-3 SW-DP acknowledgement timing

Table 11-1 SW-DP Interface Description

Module pin	Direction	Pad name	IOMUX
jtag_tck	I	IO_SDMMC0d2_JTAGtck_GPIO1A2vccio3	GRF_GPIO1A_IOMUX[5:4]=2’b10 & mmc0_detn
jtag_tms	I/O	IO_SDMMC0d3_JTAGtms_GPIO1A3vccio3	GRF_GPIO1A_IOMUX[7:6]=2’b10 & mmc0_detn

Note : mmc0_detn, when high, no sd card is used.

Chapter 12 eFuse

12.1 Overview

In this device, there are two eFuse. Both of them are organized as 32 bits by 32 one-time programmable electrical fuses with random access interface.
It is a type of non-volatile memory fabricated in standard CMOS logic process. The main features are as follows:

- Programming condition : VQPS_EFUSE = 1.5V±10%
- Program time : 10us±0.2us .
- Read condition : VQPS_EFUSE = 0V
- Provide standby mode

12.2 Block Diagram

In the following diagram, all the signals except power supply VDD_EFUSE, VSS_EFUSE and VQPS_EFUSE are controlled by registers. For detailed description, please refer to detailed register descriptions.

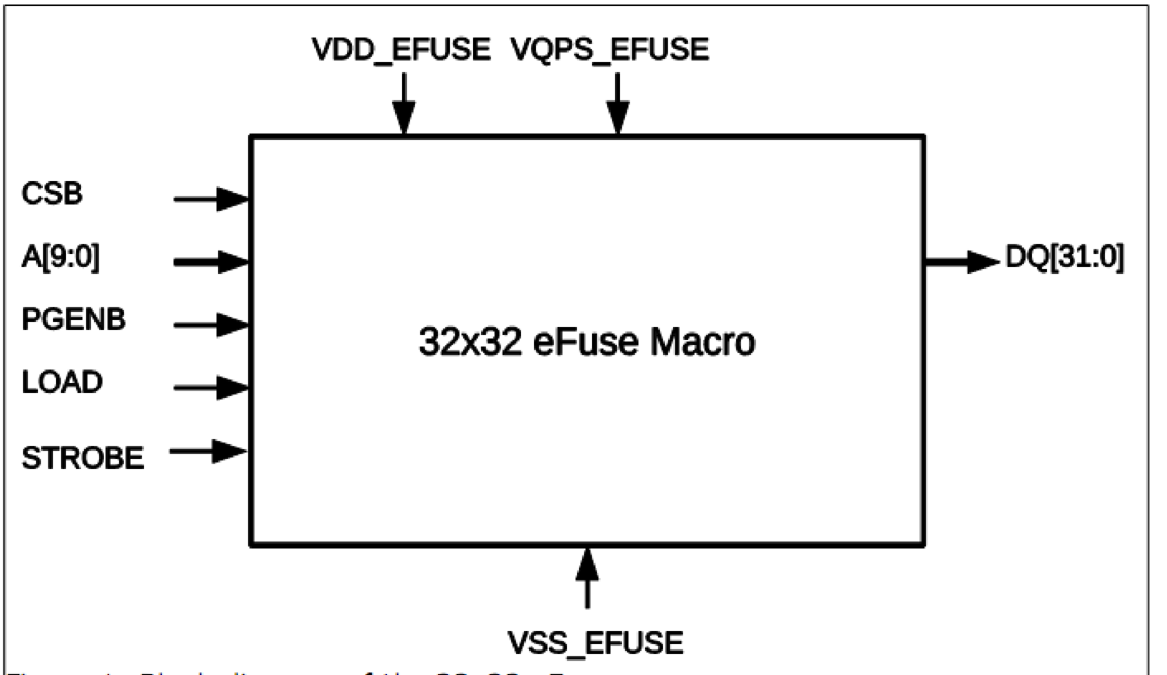


Figure 1: Block diagram of the 32x32 eFuse macro

Fig. 12-1 eFuse block diagram

12.3 Function Description

eFuse has three operation modes. They are defined as standby, read and programming.

Program (PGM) Mode

In order to enter programming mode, the following conditions need to be satisfied: VQPS_EFUSE is at high voltage, LOAD signal is low, PGENB signal is low, and CSB signal is low. All bits can be individually programmed (one at a time) with the proper address selected, the STROBE signal high and the address bits satisfying setup and hold time with respect to STROBE.

Read Mode

In order to enter read mode the following conditions need to be satisfied: VQPS_EFUSE is at ground, the LOAD signal is high, the PGENB signal is high, and the CSB is low. An entire 8-bit word of data can be read in one read operation with STROBE being high and a proper address selected (address signals A5~A7 are "don't cares").

Standby Mode

Standby is defined when the macro is not being programmed or read. The conditions for standby mode are: the LOAD signal is low, the STROBE signal is low, the CSB signal is high and PGENB is high.

Table 1-1 list of allowed modes

Signals/Supplies					Mode
VQPS_EFUSE	CSB	PGENB	LOAD	STROBE*	
High	Low	Low	Low	Low to High	Programming
Low	Low	High	High	Low to High	Read
Low	High	High	Low	Low	Standby

12.4 Register Description

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
EFUSE_EFUSE_MOD	0x0000	W	0x00000006	EFUSE Mode Control Register
EFUSE_EFUSE_RD_MASK_S	0x0004	W	0x00000000	EFUSE Read Mask control In Secure Mode
EFUSE_EFUSE_PG_MASK_S	0x0008	W	0x00000000	EFUSE Program Mask control In Secure Mode
EFUSE_EFUSE_RD_MASK_NS	0x000c	W	0x00000000	EFUSE Read Mask control In Non-Secure Mode
EFUSE_EFUSE_PG_MASK_NS	0x0010	W	0x00000000	EFUSE Program Mask control In Non-Secure Mode
EFUSE_EFUSE_INT_CON	0x0014	W	0x00000000	EFUSE Interrupt Control
EFUSE_EFUSE_INT_STATUS	0x0018	W	0x00000000	EFUSE Interrupt Status
EFUSE_EFUSE_USER_CTRL	0x001c	W	0x00000009	EFUSE User Mode Control
EFUSE_EFUSE_DOUT	0x0020	W	0x00000000	EFUSE Data Out
EFUSE_EFUSE_AUTO_CTRL	0x0024	W	0x00000000	EFUSE Auto Mode Control
EFUSE_T_CSB_P	0x0028	W	0x000f0000	EFUSE CSB timing control in Program mode
EFUSE_T_PGENB_P	0x002c	W	0x00000000	EFUSE PGENB timing control in Program mode
EFUSE_T_LOAD_P	0x0030	W	0x00000000	EFUSE LOAD timing control in Program mode
EFUSE_T_ADDR_P	0x0034	W	0x00000000	EFUSE Address timing control in Program mode

Name	Offset	Size	Reset Value	Description
EFUSE_T_STROBE_P	0x0038	W	0x00000000	EFUSE STROBE timing control in Program mode
EFUSE_T_CSB_R	0x003c	W	0x00000000	EFUSE CSB timing control in Read mode
EFUSE_T_PGENB_R	0x0040	W	0x00000000	EFUSE PGENB timing control in Read mode
EFUSE_T_LOAD_R	0x0044	W	0x00000000	EFUSE LOAD timing control in Read mode
EFUSE_T_ADDR_R	0x0048	W	0x00000000	EFUSE ADDR timing control in Read mode
EFUSE_T_STROBE_R	0x004c	W	0x00000000	EFUSE STROBE timing control in Read mode
EFUSE_REVISION	0x0050	W	0x00000010	EFUSE Design Revision

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

12.4.2 Detail Register Description

EFUSE_EFUSE_MOD

Address: Operational Base + offset (0x0000)

EFUSE Mode Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	efuse_rd_enb_user efuse read enable in user mode 0: disable 1: enable
5	RW	0x0	efuse_pg_enb_user efuse program enable in user mode 0: disable 1: enable
4	RW	0x0	strobe_pol STROBE polarity 0: Active HIGH 1: Active LOW
3	RW	0x0	load_pol LOAD polarity 0: Active HIGH 1: Active LOW
2	RW	0x1	pgenb_pol PGENB polarity 0: Active HIGH 1: Active LOW

Bit	Attr	Reset Value	Description
1	R/W SC	0x1	csb_pol CSB polarity 0: Active HIGH 1: Active LOW
0	R/W SC	0x0	work_mod EFUSE controller working mode 0: auto_mode 1: user_mode

EFUSE_EFUSE_RD_MASK_S
Address: Operational Base + offset (0x0004)
EFUSE Read Mask control In Secure Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	efuse_rd_mask_s efuse read mask enable in secure mode 0: disable 1: enable

EFUSE_EFUSE_PG_MASK_S
Address: Operational Base + offset (0x0008)
EFUSE Program Mask control In Secure Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	efuse_pg_mask_s efuse program mask enable in secure mode 0: disable 1: enable

EFUSE_EFUSE_RD_MASK_NS
Address: Operational Base + offset (0x000c)
EFUSE Read Mask control In Non-Secure Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1 C	0x0	efuse_rd_mask_ns efuse read mask enable in non-secure mode 0: disable 1: enable

EFUSE_EFUSE_PG_MASK_NS
Address: Operational Base + offset (0x0010)

EFUSE Program Mask control In Non-Secure Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	efuse_pg_mask_ns efuse program mask enable in non-secure mode 0: disable 1: enable

EFUSE_EFUSE_INT_CON

Address: Operational Base + offset (0x0014)

EFUSE Interrupt Control

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	user_s_pg_mask_int_en user_s_pg_mask interrupt enable 0: disable 1: enable
11	RW	0x0	user_s_rd_mask_int_en user_s_rd_mask interrupt enable 0: disable 1: enable
10	RW	0x0	user_ns_pg_mask_int_en user_ns_pg_mask interrupt enable 0: disable 1: enable
9	RW	0x0	user_ns_rd_mask_int_en user_ns_rd_mask interrupt enable 0: disable 1: enable
8	RW	0x0	auto_s_pg_mask_int_en auto_s_pg_mask interrupt enable 0: disable 1: enable
7	RW	0x0	auto_s_rd_mask_int_en auto_s_rd_mask interrupt enable 0: disable 1: enable
6	RW	0x0	auto_ns_pg_mask_int_en auto_ns_pg_mask interrupt enable 0: disable 1: enable
5	RW	0x0	auto_ns_rd_mask_int_en auto_ns_rd_mask interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
4	RW	0x0	user_s_access_ns_err_int_en user_s_access_ns_err interrupt enable 0: disable 1: enable
3	RW	0x0	user_ns_access_s_err_int_en user_ns_access_s_err interrupt enable 0: disable 1: enable
2	RW	0x0	auto_s_access_ns_err_int_en auto_s_access_ns_err interrupt enable 0: disable 1: enable
1	RW	0x0	auto_ns_access_s_err_int_en auto_ns_access_s_err interrupt enable 0: disable 1: enable
0	RO	0x0	finish_int_en finish interrupt enable 0: disable 1: enable

EFUSE_EFUSE_INT_STATUS

Address: Operational Base + offset (0x0018)

EFUSE Interrupt Status

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	W1C	0x0	user_s_pg_mask_int_status user_s_pg_mask_int status bit
11	W1C	0x0	user_s_rd_mask_int_status user_s_rd_mask_int status bit
10	W1C	0x0	user_ns_pg_mask_int_status user_ns_pg_mask_int status bit
9	W1C	0x0	user_ns_rd_mask_int_status user_ns_rd_mask_int status bit
8	W1C	0x0	auto_s_pg_mask_int_status auto_s_pg_mask_int status bit

Bit	Attr	Reset Value	Description
7	W1C	0x0	auto_s_rd_mask_int_status auto_s_rd_mask_int status bit
6	W1C	0x0	auto_ns_pg_mask_int_status auto_ns_pg_mask_int status bit
5	W1C	0x0	auto_ns_rd_mask_int_status auto_ns_rd_mask_int status bit
4	W1C	0x0	user_s_access_ns_err_int_status user_s_access_ns_err_int status bits
3	W1C	0x0	user_ns_access_s_err_int_status user_ns_access_s_err_int status bit
2	W1C	0x0	auto_s_access_ns_err_int_status auto_s_access_ns_err_int status bit
1	W1C	0x0	auto_ns_access_s_err_int_status auto_ns_access_s_err_int status bit
0	W1C	0x0	finish_int_status finish_int status bit

EFUSE_EFUSE_USER_CTRL

Address: Operational Base + offset (0x001c)

EFUSE User Mode Control

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	efuse_addr_user efuse_addr bit control in user mode
15:4	RO	0x0	reserved
3	RW	0x1	efuse_pgenb_user efuse_pgenb bit control in user mode
2	RW	0x0	efuse_load_user efuse_load bit control in user mode
1	RW	0x0	efuse_strobe_user efuse_strobe bit control in user mode
0	RO	0x1	efuse_csb_user efuse_csb bit control in user mode

EFUSE_EFUSE_DOUT

Address: Operational Base + offset (0x0020)

EFUSE Data Out

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	efuse_dout efuse data out

EFUSE_EFUSE_AUTO_CTRL

Address: Operational Base + offset (0x0024)

EFUSE Auto Mode Control

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	efuse_addr_auto efuse address in auto mode
15:2	RO	0x0	reserved
1	RW	0x0	p_r_mode program and read control 0: programming mode 1: read mode
0	R/W SC	0x0	enb enable of auto mode 0: disable 1: enable Note, this bit is clear auto

EFUSE_T_CSB_P

Address: Operational Base + offset (0x0028)

EFUSE CSB timing control in Program mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RO	0xf	t_csb_p_s csbstart delay time in programming mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_csb_p_l lasted time in programming mode

EFUSE_T_PGENB_P

Address: Operational Base + offset (0x002c)

EFUSE PGENB timing control in Program mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_pgenb_p_s pgenb start delay time in programming mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_pgenb_p_l pgenb lasted time in programming mode

EFUSE_T_LOAD_P

Address: Operational Base + offset (0x0030)

EFUSE LOAD timing control in Program mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_load_p_s load start delay time in programming mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_load_p_l load lasted time in programming mode

EFUSE_T_ADDR_P
Address: Operational Base + offset (0x0034)
EFUSE Address timing control in Program mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_addr_p_s address start delay time in programming mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_addr_p_l address lasted time in programming mode

EFUSE_T_STROBE_P
Address: Operational Base + offset (0x0038)
EFUSE STROBE timing control in Program mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_strobe_p_s strobe start delay time in programming mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_strobe_p_l strobe lasted time in programming mode

EFUSE_T_CSB_R
Address: Operational Base + offset (0x003c)
EFUSE CSB timing control in Read mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_csb_r_s csb start delay time in read mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_csb_r_l csb lasted time in read mode

EFUSE_T_PGENB_R
Address: Operational Base + offset (0x0040)
EFUSE PGENB timing control in Read mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_pgenb_r_s pgenb start delay time in read mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_pgenb_r_l pgenb lasted time in read mode

EFUSE_T_LOAD_R

Address: Operational Base + offset (0x0044)

EFUSE LOAD timing control in Read mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_load_r_s load start delay time in read mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_load_r_l load lasted time in read mode

EFUSE_T_ADDR_R

Address: Operational Base + offset (0x0048)

EFUSE ADDR timing control in Read mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_addr_r_s address start delay time in read mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_addr_r_l address lasted time in read mode

EFUSE_T_STROBE_R

Address: Operational Base + offset (0x004c)

EFUSE STROBE timing control in Read mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	t_strobe_r_s strobe start delay time in read mode
15:10	RO	0x0	reserved
9:0	RW	0x000	t_strobe_r_l strobe lasted time in read mode

EFUSE_REVISION

Address: Operational Base + offset (0x0050)

EFUSE Design Revision

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x10	revision efuse design revsion

12.5 Timing Diagram

- **When efuse32×32 is in program(PGM) mode.**

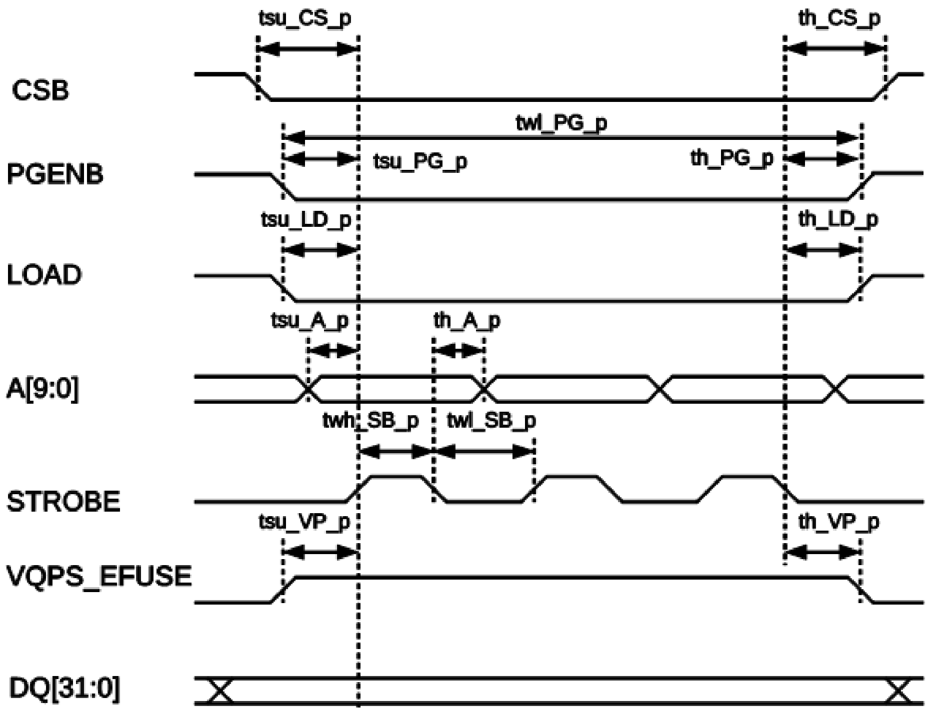


Fig. 12-2 efuse32×32 timing diagram in program mode

- **When efuse32×8 is in read mode.**

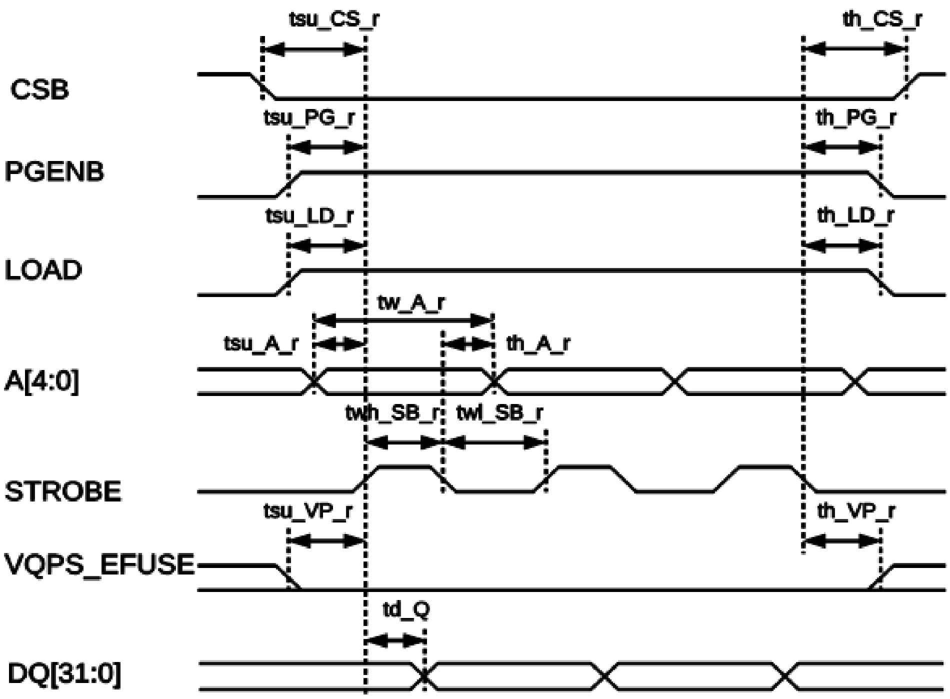


Fig. 12-3 efuse32×32 timing diagram in read mode

The following table has shows the detailed value for timing parameters in the above diagram.

Table 12-2 eFuse timing parameters list

Mode	Item	Description	Min	Typ	Max	Unit
Read Mode	twh_SB_r	Pulse width high of STROBE read strobe	20		-	ns
	twl_SB_r	Pulse width low of STROBE read strobe	15		-	ns
	tsu_A_r	A[9:0] to STROBE setup time in read mode	25		-	ns
	th_A_r	A[9:0] to STROBE hold time in read mode	3		-	ns
	tw_A_r	A[9:0] pulse width while LOAD high in read mode	48		100	ns
	tsu_CS_r	CSB to STROBE setup time in read mode	16		-	ns
	th_CS_r	CSB to STROBE hold time in read mode	6		-	ns
	tsu_PG_r	PGENB to STROBE setup time in read mode	14		-	ns
	th_PG_r	PGENB to STROBE hold time in read mode	10		-	ns
	tsu_LD_r	LOAD to STROBE setup time in read mode	10		-	ns
	th_LD_r	LOAD to STROBE hold time in read mode	7		-	ns
	tsu_VP_r	VQPS_EFUSE to STROBE setup time in read mode	20		-	ns
	th_VP_r	VQPS_EFUSE to STROBE hold time in read mode	20		-	ns
	td_Q	DQ[31:0] delay time after STROBE high	0		8	ns
PGM Mode	twh_SB_p	Pulse width high of STROBE PGM strobe	9.8	10	10.2	us
	twl_SB_p	Pulse width low of STROBE PGM strobe	15		-	ns
	tsu_A_p	A[9:0] to STROBE setup time in PGM mode	12		-	ns
	th_A_p	A[9:0] to STROBE hold time in PGM mode	3		-	ns
	tsu_CS_p	CSB to STROBE setup time in PGM mode	16		-	ns
	th_CS_p	CSB to STROBE hold time in PGM mode	6		-	ns
	tsu_PG_p	PGENB to STROBE setup time in PGM mode	14		-	ns
	th_PG_p	PGENB to STROBE hold time in PGM mode	10		-	ns
	twl_PG_p	PGENB pulse width low (cumulative) in PGM mode	-		100	ms
	tsu_LD_p	LOAD to STROBE setup time in PGM mode	10		-	ns
	th_LD_p	LOAD to STROBE hold time in PGM mode	7		-	ns
	tsu_VP_p	VQPS_EFUSE to STROBE setup time in PGM mode	20		-	ns
	th_VP_p	VQPS_EFUSE to STROBE hold time in PGM mode	20		-	ns

12.6 Application Notes

During usage of efuse, customers must pay more attention to the following items:
1. In condition of program(PGM) mode, VQPS_EFUSE= 1.5V±10%.
2. Q0~Q7 will be reset to “0” once CSB at high.
3. No data access allowed at the rising edge of CSB.
4. All the program timing for each signal must be more than the value defined in the timing table. Please refer to the timing diagram of READ_MODE and PGM_MODE as well as the function description.

Configuration steps:
1. set csb(EFUSE_CTRL[0]), pgenb(EFUSE_CTRL[3]), load(EFUSE_CTRL[2]) at proper value.
2. set addr(EFUSE_CTRL[15:6]).
3. set strobe(EFUSE_CTRL[1]).
4. stay for enough cycle. (Satisfy the timing parameter)
5. dis-assert strobe(EFUSE_CTRL[1]).
6. set csb(EFUSE_CTRL[0]), pgenb(EFUSE_CTRL[3]), load(EFUSE_CTRL[2]) at proper value.
7. read efuse_data(EFUSE_DOUT).

Chapter 13 WatchDog

13.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may becaused by conflicting parts or programs in a SoC.The WDT would generate interrupt or reset signal when it's 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 is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
  - Generate a system reset
  - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period

13.2 Block Diagram

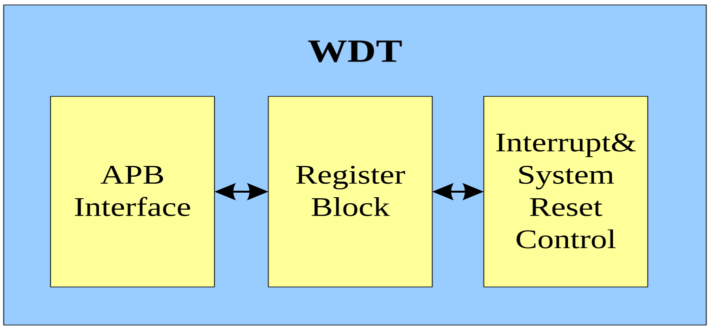


Fig. 13-1 WDT block diagram

Block Descriptions:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & system reset control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

13.3 Function Description

13.3.1 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

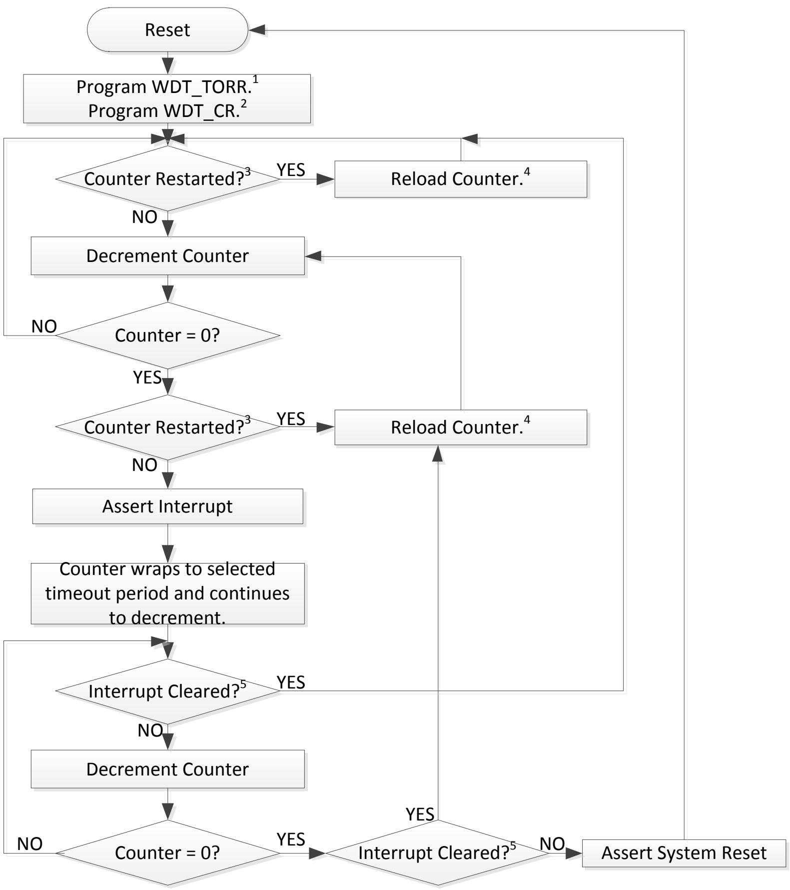
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

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.

13.3.2 Programming sequence

Operation Flow Chart (Response mode=1)



1. Select required timeout period.
2. Set reset pulse length, response mode, and enable WDT.
3. Write 0x76 to WDT_CRR.
4. Starts back to selected timeout period.
5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 13-2 WDT Operation Flow

13.4 Register Description

This section describes the control/status registers of the design.

13.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x00000000	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

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

13.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Control Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. 000: 2 pclk cycles 001: 4 pclk cycles 010: 8 pclk cycles 011: 16 pclk cycles 100: 32 pclk cycles 101: 64 pclk cycles 110: 128 pclk cycles 111: 256 pclk cycles
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 0: Generate a system reset. 1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.

Bit	Attr	Reset Value	Description
0	RW	0x0	wdt_en WDT enable Writable when the configuration parameter WDT_ALWAYS_EN=0, otherwise,it is readable. This bit is used to enable and disable the DW_apb_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. 0: WDT disabled; 1: WDT enabled.

WDT_TORR
Address: Operational Base + offset (0x0004)
Timeout range Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 0000: 0x0000ffff 0001: 0x0001ffff 0010: 0x0003ffff 0011: 0x0007ffff 0100: 0x000fffff 0101: 0x001fffff 0110: 0x003fffff 0111: 0x007fffff 1000: 0x00ffffff 1001: 0x01ffffff 1010: 0x03ffffff 1011: 0x07ffffff 1100: 0x0fffffff 1101: 0x1fffffff 1110: 0x3fffffff 1111: 0x7fffffff

WDT_CCVR
Address: Operational Base + offset (0x0008)
Current counter value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt Current counter value This register, when read, is the current value of the internal counter. This value is read coherently whenever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)
Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	cnt_restart Counter restart This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.

WDT_STAT
Address: Operational Base + offset (0x0010)
Interrupt status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1: Interrupt is active regardless of polarity; 0: Interrupt is inactive.

WDT_EOI
Address: Operational Base + offset (0x0014)
Interrupt clear Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RC	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

13.5 Application Notes

Please refer to the function description section

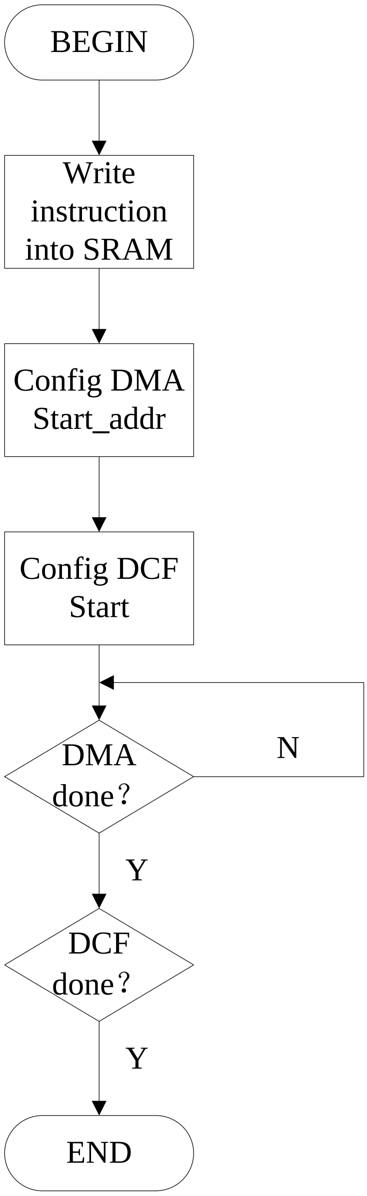


Fig. 13-3 DCF work flow

13.5.1 Instruction format

One piece of instruction, which is 64bit, should consist of the information of :

1、 Address
2、 Data
3、 Command

63:56	55:34	33:32	31:0
cmd[7:0]	addr[23:2]	(r1,r0)	data

The overall principle of instruction information is:

1、 addr[31:24] is reserved for 8bit command, which represents the corresponding operation

2、 addr[1:0] is used to indicate operation of r0 or r1

3、 addr[23:2] is the real bus address. If 0, it means no bus operation ; if not 0, it means a combination of 2 instructions with a bus operation ahead and an arithmetic c operation followed in order to improve efficiency.

For example, let us analyze the instruction: 01620002_00000003

1、 command is 1,which represents an bitwise AND operation

2、 address is 0xff620000, represents a bus-read operation

3、 r1 is indicated, represents that the middle result is stored into internal register r1.

4、 Data is 0x00000003, represents that the operation value is 0x3

So, this instruction will do following operations:

1、 LDR #0xff620000, r1 ; //read register 0xff620000, and store value into r1

2、 AND r1, 0x00000003 ; //r1 is bitwise AND with 0x3, and re-store the result.

The following table lists all the supported command

INSTR	cmd[7:0]	addr[23:2]	R1	R0	Data[31:0]	
IDLE	8’h00	NA	NA	NA	#data	IDL #data
AND	8’h01	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; AND r0 #data
			1	0	#data	ldr #addr r1 ; AND r1 #data
			1	1	NA	ldr #addr r1 ; AND r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	AND r0 #data
			1	0	#data	AND r1 #data
			1	1	NA	AND r1 r0
OR	8’h02	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; OR r0 #data
			1	0	#data	ldr #addr r1 ; OR r1 #data
			1	1	NA	ldr #addr r1 ; OR r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	OR r0 #data
			1	0	#data	OR r1 #data
			1	1	NA	OR r1 r0
INV	8’h03	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; XOR r0 ^#data
			1	0	#data	ldr #addr r1 ; XOR r1 ^#data
			1	1	#data	ldr #addr r1 ; XOR r1 ^r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	NA	INV r0
			1	0	NA	INV R1
			1	1	NA	SWP r0 r1
LSR	8’h04	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; LSR r0 #data
			1	0	#data	ldr #addr r1 ; LSR r1 #data
			1	1	NA	ldr #addr r1 ; LSR r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	LSR r0 #data
			1	0	#data	LSR r1 #data

			1	1	NA	LSR r1 r0
RSR	8’ h05	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; RSR r0 #data
			1	0	#data	ldr #addr r1 ; RSR r1 #data
			1	1	NA	ldr #addr r1 ; RSR r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	RSR r0 #data
			1	0	#data	RSR r1 #data
			1	1	NA	RSR r1 r0
CMPEQ	8’ h06	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; CMPEQ r0 #data,flag
			1	0	#data	ldr #addr r0 ; CMPEQ r1 #data,flag
			1	1	NA	ldr #addr r0 ; CMPEQ r1 r0,flag
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	CMPEQ r0 #data, flag
			1	0	#data	CMPEQ r1 #data, flag
			1	1	NA	CMPEQ r1 r0, flag
CMPNE	8’ h07	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; CMPNE r0 #data,flag
			1	0	#data	ldr #addr r1 ; CMPNE r1 #data,flag
			1	1	NA	ldr #addr r1 ; CMPNE r1 r0,flag
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	CMPNE r0 #data, flag
			1	0	#data	CMPNE r1 #data, flag
			1	1	NA	CMPNE r1 r0, flag
ADD	8’ h08	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; ADD r0 #data
			1	0	#data	ldr #addr r0 ; ADD r1 #data
			1	1	NA	ldr #addr r0 ; ADD r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	ADD r0 #data
			1	0	#data	ADD r1 #data
			1	1	NA	ADD r1 r0
SUB	8’ h09	#addr	0	0	NA	ldr #addr r0 ; ldr #addr r1
			0	1	#data	ldr #addr r0 ; SUB r0 #data
			1	0	#data	ldr #addr r0 ; SUB r1 #data
			1	1	NA	ldr #addr r0 ; SUB r1 r0
		All 0	0	0	NA	mov r0 r0 ; mov r1 r1
			0	1	#data	SUB r0 #data
			1	0	#data	SUB r1 #data
			1	1	NA	SUB r1 r0
STR	8’ h0a	#addr	0	0	#data	STR #addr #data
			0	1	NA	STR #ADDR r0
			1	0	NA	STR #ADDR r1
			1	1	#data	STR #addr #data

ISB	8’ h0b	#addr	0	0	#data	STR #addr #data
			0	1	NA	STR #ADDR r0
			1	0	NA	STR #ADDR r1
			1	1	#data	STR #addr #data
POLEQ	8’ h0c	NA	0	1	#data	poll r0=#data, repeat last command
			1	0	#data	poll r1=#data, repeat last command
			1	1	NA	poll r1=r0, repeat last command
POLNEQ	8’ h0d	NA	0	1	#data	poll r0!=#data, repeat last command
			1	0	#data	poll r1!=#data, repeat last command
			1	1	NA	poll r1!=r0, repeat last command
BL_U	8’ h0e	NA	NA	NA	#data	brr #data, ?flag (upwards)
		NA	0	1	NA	brr r0, ?flag
		NA	1	0	NA	brr r1, ?flag
BL_D	8’ h0f	NA	NA	NA	#data	brr #data, ?flag (downwards)
		NA	0	1	NA	brr r0, ?flag
		NA	1	0	NA	brr r1, ?flag
DMA_S	8’ h10	NA	NA	NA	#data	set dma_start_addr = #data
DMA_D	8’ h11	NA	NA	NA	#data	set dma_end_addr = #data
DMA_DO	8’ h12	NA	NA	NA	#data	set dma_length = #data (byte) dma_start
END	8’ hed	NA	NA	NA	NA	End of instruction

13.5.2 Hardware trigger flow

When DCF_CTRL.vop_hw_en is enabled, DCF can be triggered by any of the followed three sources ：dma_finish 、vop_standby、vop_clkgate_en。

DCF is edge sensitive for dma_finish signal, and level sensitive for vop_standby and vop_clkgate_en signal.

When DCF is working， a dcf_idle is driven to low to indicate vop not to exit vop_standby status. And when DCF is not working, dcf_idle is driven to high for SOC and VOP to inquire.

Chapter 14 Timer

14.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. In RK3328 there are 6 Timers and 2 Secure Timers(STimer).
Timer5 and STimer0~1 count up from zero to a programmed value and generate an interrupt when the counter reaches the programmed value.
Timer0~4 count down from a programmed value to zero and generate an interrupt when the counter reaches zero.
Timer supports the following features:

- Timer0~5 is used for no-secure, STimer0~1 is used for secure.
- Two operation modes: free-running and user-defined count.
- Maskable for each individual interrupt.

14.2 Block Diagram

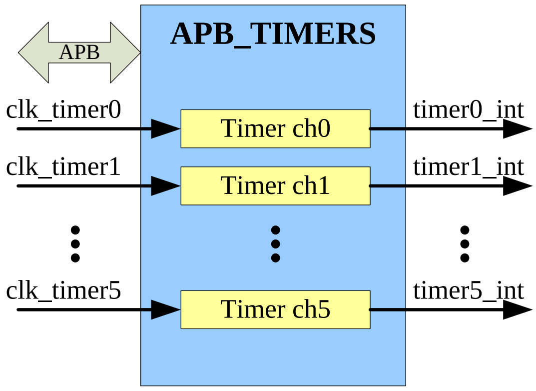


Fig. 14-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channels) that in the bus subsystem. The Stimers that in the bus subsystem only include two programmable timer channels.

14.3 Function Description

14.3.1 Timer clock

TIMER0~ TIMER5 and STIMER0~1 are in the pd_bus subsystem. The timer clock is 24MHz OSC.

14.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG (0≤n≤5) register:

- Disable the timer by writing a “0” to the timer enable bit (bit 0). Accordingly, the timer_en output signal is de-asserted.
- Program the timer mode—user-defined or free-running—by writing a “0” or “1” respectively, to the timer mode bit (bit 1).
- Set the interrupt mask as either masked or not masked by writing a “0” or “1” respectively, to the timer interrupt mask bit (bit 2).

2. Load the timer count value into the TIMERn_LOAD_COUNT1 (0≤n≤5) and TIMERn_ LOAD_COUNT0 (0≤n≤5) register.
3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG (0≤n≤5).

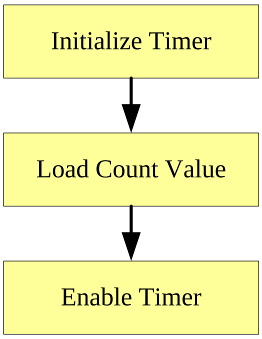


Fig. 14-2 Timer Usage Flow

14.3.3 Loading a timer count value

For the descending Timers(Timer0~4).The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the load count register (TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0). Two events can cause a timer to load the initial value from its load count register:

- Timer is enabled after reset or disabled.
- Timer counts down to 0, when timer is configured into free-running mode.

For the incremental Timers(Timer5 and STimer0~1).The initial value for each timer is zero. The count register will count up to the value loaded in the register TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0. Two events can cause a timer to load zero:

- Timer is enabled after reset or disabled.
- Timer counts up to the value stored in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0, when timer is configured into free-running mode.

14.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 registers (for descending timers) or zero (for incremental timers) as initial value. When the timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_ COUNT0 (for incremental timers),it will not automatically reload the count register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads the TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0(for descending timers) or zero (for incremental timers)register as initial value. Timer will automatically reload the count register, when timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for incremental timers).

14.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_n_LOAD_COUNT0	0x0000	W	0x00000000	Timer n Load Count Register
TIMER_n_LOAD_COUNT1	0x0004	W	0x00000000	Timer n Load Count Register
TIMER_n_CURRENT_VALUE0	0x0008	W	0x00000000	Timer n Current Value Register
TIMER_n_CURRENT_VALUE1	0x000c	W	0x00000000	Timer n Current Value Register
TIMER_n_CONTROLREG	0x0010	W	0x00000000	Timer n Control Register
TIMER_n_INTSTATUS	0x0018	W	0x00000000	Timer Interrupt Stauts Register

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

14.4.2 Detail Register Description

TIMER_n_LOAD_COUNT0

Address: Operational Base + offset (0x00)

Timer n Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_low bits Low 32 bits value to be loaded into Timer n. This is the value from which counting commences.

TIMER_n_LOAD_COUNT1

Address: Operational Base + offset (0x04)

Timer n Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_high bits High 32 bits value to be loaded into Timer n. This is the value from which counting commences.

TIMER_n_CURRENT_VALUE0

Address: Operational Base + offset (0x08)

Timer n Current Value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_cnt_lowbits Low 32 bits of current value of timer n.

TIMER_n_CURRENT_VALUE1

Address: Operational Base + offset (0x0c)

Timer n Current Value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_cnt_highbits High 32 bits of current value of timer n.

TIMER_n_CONTROLREG

Address: Operational Base + offset (0x10)

Timer n Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	int_en Timer interrupt mask 0: mask 1: not mask
1	RW	0x0	timer_mode Timer mode. 0: free-running mode 1: user-defined count mode
0	RW	0x0	timer_en Timer enable. 0: disable 1: enable

TIMER_n_INTSTATUS

Address: Operational Base + offset (0x18)

Timer Interrupt Stauts Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1 C	0x0	int_pd This register contains the interrupt status for timer n. Write 1 to this register will clear the interrupt.

14.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (0⩽n⩽5)), the 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.

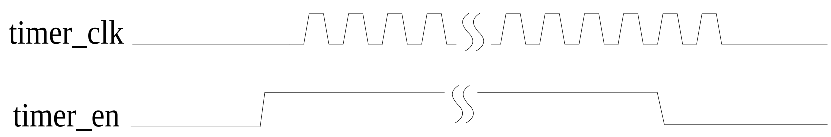


Fig. 14-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 15 Transport Stream Processing Module (TSP)

15.1 Overview

The Transport Stream Processing Module(TSP) is designed for processing Transport Stream Packets, including receiving TS packets, PID filtering, TS descrambling, De-multiplexing and TS outputting. Processed data are transferred to memory buffer which are continued to be processing by software.

TSP supports the following features:

- Supports 1 TS input channels
- Supports 4 TS Input Mode: sync/valid mode in the case of serial TS input; nosync/valid mode, sync/valid, sync/burst mode in the case of parallel TS input
- Supports 2 TS sources: demodulators and local memory
- Supports 1 Built-in PTIs(Programmable Transport Interface) to process TS simultaneously
- Supports 1 PVR(Personal Video Recording) output channel
- 1 built-in multi-channel DMA Controller
- Support DMA LLP transfer
- Each PTI supports
  - 64 PID filters
  - TS descrambling with 16 sets of Control Word under CSA v2.0 standard, up to 104Mbps
  - 16 PES/ES filters with PTS/DTS extraction and ES start code detection
  - 4/8 PCR extraction channels
  - 64 Section filters with CRC check, and three interrupt mode: stop per unit, full-stop, recycle mode with version number check
  - PID done and error interrupts for each channel
  - PCR/DTS/PTS extraction interrupt for each channel
- Support 32 bit AXI MMU.

15.2 Block Diagram

The TSP wrapper comprises of following components:

- TSP module (include: AHB slave, register block ,PTI ,DMAC, AHB master)
- AHB/AXI bridge
- 32bit AXI MMU

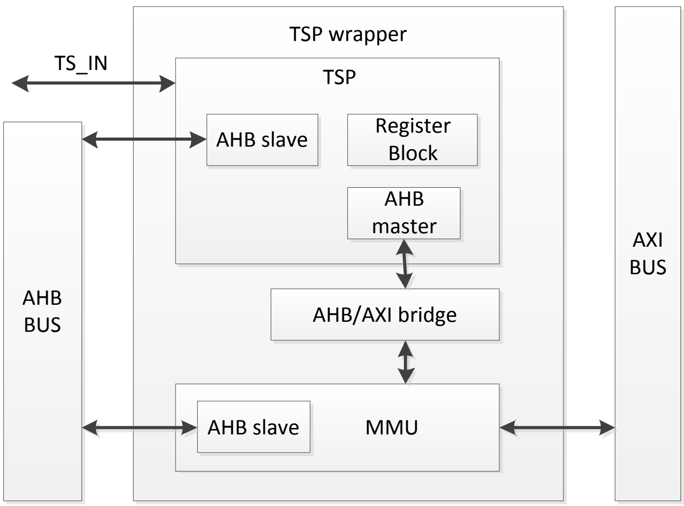


Fig. 15-1 TSP architecture

AHB Slave INTERFACE

The host processor can get access to the TSP and MMU register block through AHB slave interface. The slave interface supports 32bit access.

Register block

All registers in the TSP are addressed at 32-bit boundaries to remain consistent with the AHB 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.

PTI

Most of the TS processing are dealt with PTI. TS packets are re-synchronized, filtered, descrambled and demultiplexing, and the processed packets are transferred to memory buffer to be processed further by software. The embedded TS in interface can receive TS packets by connecting to a compliant TS demodulator. TS stream stored in the local memory is another source to fed into PTI through by using LLP DMA mode.

DMAC

The DMAC performs all DMA transfers which get access to memory.

AHB/AXI bridge

Convert AHB master to AXI master.

MMU

Support AXI interface,4K page size and TLB pre-fetch. Data bus width is 32 bit.

15.3 Function Description

15.3.1 TS Stream of TS_IN Interface

TS_IN interface supports 4 input TS stream mode: sync/valid serial mode, sync/valid parallel mode, sync/burst parallel mode, nosync/valid parallel mode.

A.Sync/Valid Serial Mode

In this mode, TS_IN interface takes use of TSI_SYNC and TSI_VALID clocked with TSI_CLK signal to sample input serial TS packet data.

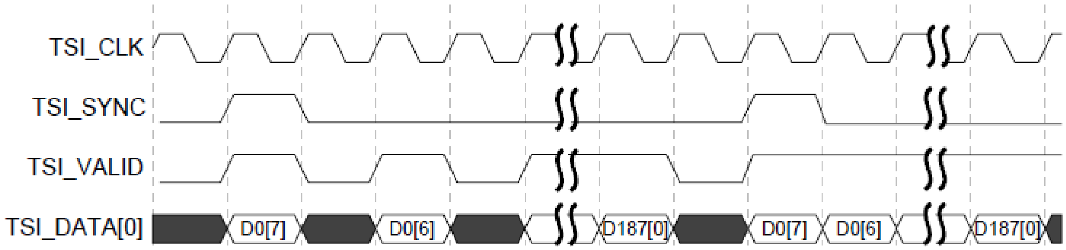


Fig. 15-2 Sync/Valid Serial Mode with Msb-Lsb Bit Ordering

TSI_SYNC must be active high together with TSI_VALID when indicating the first valid bit of a TS packet, and TSI_VALID indicates the 188*8 valid bits of a TS packet. TSI supports both msb-lsb and lsb-msb bit ordering.

B. Sync/Valid Parallel Mode

In this mode, TS_IN interface takes use of TSI_SYNC and TSI_VALID clocked with TSI_CLK signal to sample input parallel TS packet data.

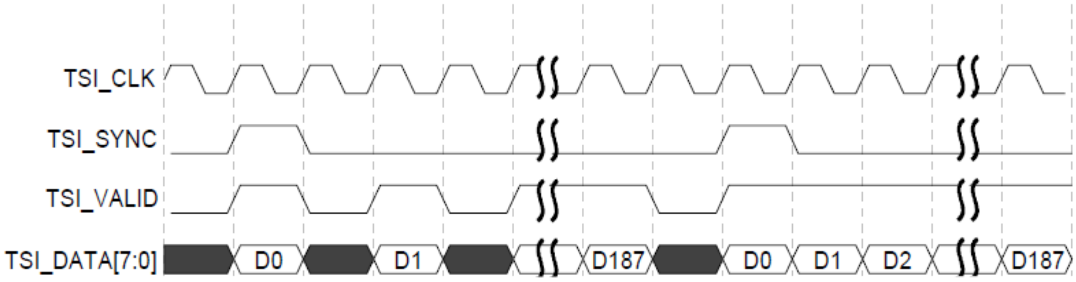


Fig. 15-3 Sync/valid Parallel Mode

TSI_SYNC must be active high together with TSI_VALID when indicating the first valid byte of a TS packet, and TSI_VALID indicates the 188 valid byte of a TS packet.

C. Sync/Burst Parallel Mode

In this mode, TSI only takes use of TSI_SYNC to sample input parallel TS packet data.

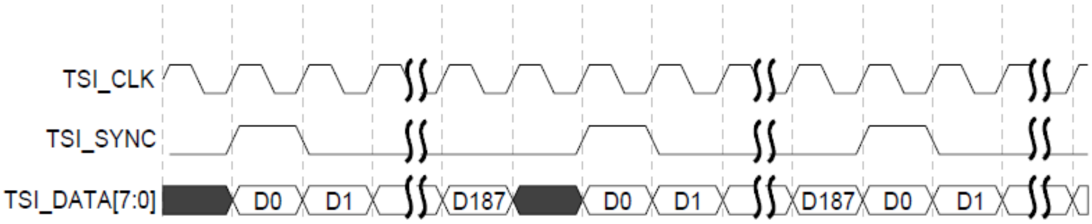


Fig. 15-4 Sync/Burst Parallel Mode

When active high, TSI_SYNC implies the first valid byte of a TS packet and remaining 187 valid bytes of a TS packet are upcoming within the following successive 187 clock cycles.

D. Nosync/Valid Parallel Mode

In this mode, TSI only takes uses of TSI_VALID to sample input parallel TS packet data.

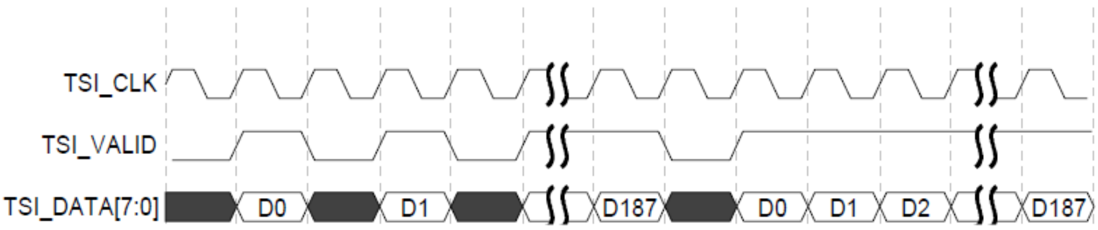


Fig. 15-5 Nosync/Valid Parallel Mode

When active high, TSI_VALID implies a valid byte of a TS packet.

15.3.2 TS output of TS Out Interface

TS out interface transmit the TS data in two mode: serial mode and parallel mode. In the serial mode, the bit order can be lsb-msb or msb-lsb.
The TS_SYNC will be active high when indicating the header of the TS packets, and it only lasts for one cycle. TS_VALID will be active high when the output TS data is valid. The output data is 188 byte TS packet data.
TS out interface also stamp the TS output stream with new PCR value, making PCR adjustment. PCR is used to measure the transport rate.

$$PCR(i) = PCR\_base(i) \times 300 + PCR\_ext(i)$$

where:

$$PCR\_base(i) = ((system\_clock\_frequency \times t(i))\ DIV\ 300)\ \%\ 2^{33}$$

$$PCR\_ext(i) = ((system\_clock\_frequency \times t(i))\ DIV\ 1)\ \%\ 300$$

$$transport\_rate(i) = \frac{((i' - i'') \times system\_clock\_frequency)}{PCR(i') - PCR(i'')}$$

Where
i' is the index of the byte containing the last bit of the immediately following program_clock_reference_base field applicable to the program being decoded.
i is the is the index of any byte in the Transport Stream for i''< i < i'.
i''is the index of the byte containing the last bit of the most recent program_clock_reference_base fieldapplicable to the program being decoded.
System clock is 27Mhz.

15.3.3 Demux and descrambling

Each PTI has 64 PID channels to deal with demultiplexing and descrambling operation. The PTI can descramble the TS Packets which are scrambled with CSA v2.0 standard. The TS packets can be scrambled either in TS level or PES level.
The demux module can do the section filtering, pes filtering and es filtering, or directly output TS packets.

15.4 Register Description

15.4.1 TSP Register Summary

Name	Offset	Size	Reset Value	Description
TSP_GCFG	0x0000	W	0x00000000	Global Configuration Register
TSP_PVR_CTRL	0x0004	W	0x00000000	PVR Control Register
TSP_PVR_LEN	0x0008	W	0x00000000	PVR DMA Transaction Length
TSP_PVR_BASE_ADDR	0x000c	W	0x00000000	PVR DMA transaction starting address
TSP_PVR_INT_STS	0x0010	W	0x00000000	PVR DMA Interrupt Status Register

Name	Offset	Size	Reset Value	Description
TSP_PVR_INT_ENA	0x0014	W	0x00000000	DMA Interrupt Enable Register
TSP_TSOUT_CTRL	0x0018	W	0x00000000	TS Out Control Register
TSP_PVR_TOP_ADDR	0x001c	W	0x00000000	PVR buffer top address
TSP_PVR_WRITE_ADDR	0x0020	W	0x00000000	PVR buffer write point
TSP_PTIx_CTRL	0x0100	W	0x00000000	PTI Channel Control Register
TSP_PTIx_LLP_CFG	0x0104	W	0x00000000	LLP DMA Control Register
TSP_PTIx_LLP_BASE	0x0108	W	0x00000000	LLP Descriptor BASE Address
TSP_PTIx_LLP_WRITE	0x010c	W	0x00000000	LLP DMA Writing Software Descriptor Counter
TSP_PTIx_LLP_READ	0x0110	W	0x00000000	LLP DMA Reading Hardware Descriptor Counter
TSP_PTIx_PID_STS0	0x0114	W	0x00000000	PTI PID Channel Status 0 Register
TSP_PTIx_PID_STS1	0x0118	W	0x00000000	PTI PID Channel Status 1 Register
TSP_PTIx_PID_STS2	0x011c	W	0x00000000	PTI PID Channel Status 2 Register
TSP_PTIx_PID_STS3	0x0120	W	0x00000000	PTI PID Channel Status 3 Register
TSP_PTIx_PID_INT_ENA0	0x0124	W	0x00000000	PID Interrupt Enable Register 0
TSP_PTIx_PID_INT_ENA1	0x0128	W	0x00000000	PID Interrupt Enable Register 1
TSP_PTIx_PID_INT_ENA2	0x012c	W	0x00000000	PID Interrupt Enable Register 2
TSP_PTIx_PID_INT_ENA3	0x0130	W	0x00000000	PID Interrupt Enable Register 3
TSP_PTIx_PCR_INT_STS	0x0134	W	0x00000000	PTI PCR Interrupt Status Register
TSP_PTIx_PCR_INT_ENA	0x0138	W	0x00000000	PTI PCR Interrupt Enable Register
TSP_PTIx_PCRn_CTRL	0x013c	W	0x00000000	PID PCR Control Register
TSP_PTIx_PCRn_H	0x015c	W	0x00000000	High Order PCR value
TSP_PTIx_PCRn_L	0x0160	W	0x00000000	Low Order PCR value
TSP_PTIx_DMA_STS	0x019c	W	0x00000000	LLP DMA Interrupt Status Register
TSP_PTIx_DMA_ENA	0x01a0	W	0x00000000	DMA Interrupt Enable Register
TSP_PTIx_DATA_FLAG0	0x01a4	W	0x00000000	PTI_PID_WRITE Flag 0
TSP_PTIx_DATA_FLAG1	0x01a8	W	0x00000000	PTI_PID_WRITE Flag 1
TSP_PTIx_LIST_FLAG	0x01ac	W	0x00000000	PTIx_LIST_WRITE Flag
TSP_PTIx_DST_STS0	0x01b0	W	0x00000000	PTI Destination Status Register
TSP_PTIx_DST_STS1	0x01b4	W	0x00000000	PTI Destination Status Register
TSP_PTIx_DST_ENA0	0x01b8	W	0x00000000	PTI Destination Interrupt Enable Register
TSP_PTIx_DST_ENA1	0x01bc	W	0x00000000	PTI Destination Interrupt Enable Register
TSP_PTIx_ECWn_H	0x0200	W	0x00000000	The Even Control Word High Order
TSP_PTIx_ECWn_L	0x0204	W	0x00000000	The Even Control Word Low Order
TSP_PTIx_OCWn_H	0x0208	W	0x00000000	The Odd Control Word High Order
TSP_PTIx_OCWn_L	0x020c	W	0x00000000	The Odd Control Word Low Order
TSP_PTIx_PIDn_CTRL	0x0300	W	0x00000000	PID Channel Control Register
TSP_PTIx_PIDn_BASE	0x0400	W	0x00000000	PTI Data Memory Buffer Base Address

Name	Offset	Size	Reset Value	Description
TSP_PTIx_PIDn_TOP	0x0404	W	0x00000000	PTI Data Memory Buffer Top Address
TSP_PTIx_PIDn_WRITE	0x0408	W	0x00000000	PTI Data Memory Buffer Hardware Writing Address
TSP_PTIx_PIDn_READ	0x040c	W	0x00000000	PTI Data Memory Buffer Software Reading Address
TSP_PTIx_LISTn_BASE	0x0800	W	0x00000000	PTI List Memory Buffer Base Address
TSP_PTIx_LISTn_TOP	0x0804	W	0x00000000	PTI List Memory Buffer Top Address
TSP_PTIx_LISTn_WRITE	0x0808	W	0x00000000	PTI List Memory Buffer Hardware Writing Address
TSP_PTIx_LISTn_READ	0x080c	W	0x00000000	PTI List Memory Buffer Software Reading Address
TSP_PTIx_PIDn_CFG	0x0900	W	0x00000008	PID Demux Configure Register
TSP_PTIx_PIDn_FILT_0	0x0904	W	0x00000000	Fliter Word 0
TSP_PTIx_PIDn_FILT_1	0x0908	W	0x00000000	Fliter Word 1
TSP_PTIx_PIDn_FILT_2	0x090c	W	0x00000000	Fliter Word 2
TSP_PTIx_PIDn_FILT_3	0x0910	W	0x00000000	Fliter Word 3

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

15.4.2 TSP Detail Register Description

TSP_GCFG

Address: Operational Base + offset (0x0000)

Global Configuration Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x0	arbit_cnt DMA channel arbiter counter This field is used to adjust the priority of DMA channels to prevent one channel holds the highest priority for a long time. The 3-bit field sets the largest times for a DMA channel to hold the highest priority to send the bus request. After requested times reach this limit, the highest priority is passed to next DMA channel in order.
3	RW	0x0	tsout_on TS Output Module Switch 1: TS output module switched on 0: TS output module switched off
2	RW	0x0	pvr_on PVR Module Switch 1: PVR function turned on ; 0: PVR function turned off ;

Bit	Attr	Reset Value	Description
1	RW	0x0	pti1_on PTI0 channel switch 1: PTI1 channel switched on 0: PTI1 channel switched off
0	RW	0x0	pti0_on PTI0 channel switch 1: PTI0 channel switched on 0: PTI1 channel switched off

TSP_PVR_CTRL
Address: Operational Base + offset (0x0004)
PVR Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	fixaddr_en Fix Address Mode Select 1: fixed address mode; 0: incrementing address mode;
5:4	RW	0x0	burst_mode PVR burst mode PVR DMA burst mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserverd
3:2	RW	0x0	source PVR Source Select TS source for PVR output. 00: non-PID-filtered TS packets in PTI0; 01: PID filtered TS packets in PTI0; 10: non-PID-filtered TS packets in PTI1; 11: PID-filtered TS packets in PTI1;
1	R/W SC	0x0	stop PVR stop Write 1 to stop DMA channel. DMA will complete current burst transfer and then stop. It may takes several cycles. 1: PVR Stop ; 0: no effect ;
0	R/W SC	0x0	start PVR start Write 1 to start PVR. This bit will be cleared if PVR is stopped or PVR transaction is completed. 1: start PVR 0: no effect.

TSP_PVR_LEN
Address: Operational Base + offset (0x0008)
PVR DMA Transaction Length

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	len Transaction Length Transaction Length

TSP_PVR_BASE_ADDR
Address: Operational Base + offset (0x000c)
PVR DMA transaction starting address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr PVR DMA transaction starting address PVR DMA transaction starting address

TSP_PVR_INT_STS
Address: Operational Base + offset (0x0010)
PVR DMA Interrupt Status Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	W1C	0x0	pvr_update_flag pvr address pageover flag When write_addr >= (base + top_addr/2), or write addr >= top_addr, the pvr_update_flag will assert HIGH. The application can write 1 to this bit to clear it.
1	W1C	0x0	pvr_error PVR DMA transaction error 1: error response during PVR DMA transaction; 0: no error response during PVR DMA transaction;
0	W1C	0x0	pvr_done PVR DMA transaction done 1: PVR DMA transaction completed; 0: PVR DMA transaction not completed;

TSP_PVR_INT_ENA
Address: Operational Base + offset (0x0014)
DMA Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	pvr_update_ena 1: pvr_update interrupt enable 0: pvr_update interrupt disable
1	RW	0x0	pvr_error_ena PVR DMA Transcation Error Interrupt Enable 1: Error Interrupt Enabled 0: Error Interrupt Disabled
0	RW	0x0	pvr_done_ena PVR DMA Transaction Done Interrupt Enable 1: Done Interrupt Enabled 0: Done Interrupt Disabled

TSP_TSOUT_CTRL

Address: Operational Base + offset (0x0018)

TS Out Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	tso_sdo_sel TS serial data output 1: bit[0] use as serial data output ; 0: bit[7] use as serial data output ;
5	RW	0x0	tso_clk_phase TS output clock phase 0: ts output clock; 1: inverse of ts output clock.
4	RW	0x0	mode TS Output mode Selection Output mode select: 0: Serial Mode 1: Parallel Mode
3	RW	0x0	bit_order ts output serial data byte order Indicates that the output serial data byte order, ignored in the parallel: 0: MSB to LSB 1: LSB to MSB
2:1	RW	0x0	source TS Output Source Select TS source for TS out. 00: non-PID-filtered TS packets in PTI0; 01: PID filtered TS packets in PTI0; 10: non-PID-filtered TS packets in PTI1; 11: PID-filtered TS packets in PTI1;

Bit	Attr	Reset Value	Description
0	RW	0x0	start TS out start 1: to start TS out function ; 0: to stop TS out function;

TSP_PVR_TOP_ADDR
Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvr_top_addr top address in pvr mode

TSP_PVR_WRITE_ADDR
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pvr_write_addr The core will update this register to show the PVR write addr

TSP_PTIx_CTRL
Address: Operational Base + offset (0x0100)
PTI Channel Control Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	tsi_sdi_sel TS Serial Data Input Select 1: bit[0] use as serial input data 0: bit[7] use as serial input data
20:19	RW	0x0	tsi_error_handle TS ERROR Handle 00: don't output 01: set the error indicator to 1 10: don't care
18	RW	0x0	clk_phase_sel ts input clock phase select 1'b0: ts input clock 1'b1: inverse of ts input clock

Bit	Attr	Reset Value	Description
17:16	RW	0x0	demux_burst_mode Demux DMA Burst Mode Demux DMA Mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserved
15	RW	0x0	sync_bypass Bypass mode Selection 1'b1: Bypass mode, indicating that input TS packets will not be resynchronized and directly fed into the following modules; 1'b0: Synchronous mode, default, indicating that input TS packets will be resynchronized;
14	RW	0x0	cw_byteorder Control Word format Configuration 0: Default: first byte of the word is the highest byte 1: first byte of the word is the lowest byte
13	RW	0x0	cm_on CSA Conformance Mechanism Configuration CSA Conformance Mechanism 0: CM turned off 1: CM turned on
12:11	RW	0x0	tsi_mode TSI Input Mode Selection Input mode selection: 00: Serial Sync/valid Mode 01: Parallel Sync/valid Mode 10: Parallel Sync/burst Mode 11: Parallel Nosync/valid Mode
10	RW	0x0	tsi_bit_order input serial data order Indicates that the input serial data byte order, ignored in the parallel mode: 0: MSB to LSB 1: LSB to MSB
9	RW	0x0	tsi_sel TS Input Source Select Select input TS source 1'b1: HSADC ; 1'b0: internel memory ;
8	RW	0x0	out_byteswap Output byteswap function When enabled, the word to be transferred to memory buffer "B4B3B2B1" is performed byteswapping to "B1B2B3B4".

Bit	Attr	Reset Value	Description
7	RW	0x0	in_byteswap Input TS Word Byteswap When enabled, the input TS word "B4B3B2B1" is perfomed byteswapping to "B1B2B3B4".
6:4	RW	0x0	unsync_times TS Header Unsynchronized Times If synchronous mode is selected. This field sets the successive times of TS packet header error to re-lock TS header when TS is in locked status;
3:1	RW	0x0	sync_times TS Header Synchronized Times If synchronous mode is selected. This field sets the successive times of finding TS packet header to lock the TS header when TS is in unlocked status;
0	R/W SC	0x0	clear Software clear signal It will reset the core register . It will table several cycles. After reset done, soft_reset will be low. 1. reset; 0. no effect.

TSP_PTIx_LLP_CFG

Address: Operational Base + offset (0x0104)

LLP DMA Control Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:8	RW	0x0	threshold LLP Transfer Threshold The depth for LLP descriptors is 64. An interrupt will be asserted when transfer reaches the threshold set if DMA transfer interrupt is enabled. 00: 1/1 depth 01: 1/2 depth 10: 1/4 depth 11: 1/8 depth
7:6	RW	0x0	burst_mode LLP DMA Burst Mode LLP DMA Burst Mode 2'b00: INCR4 2'b01: INCR8 2'b10: INCR16 2'b11: Reserverd

Bit	Attr	Reset Value	Description
5	RW	0x0	hw_trigger Hardware Trigger Select 1. hardware trigger; 0. software trigger;
4	RW	0x0	fix_addr_en Fix Address Mode Select 1: fixed address mode; 0: incrementing address mode;
3	W1C	0x0	cfg_done LLP DMA Configuration Done When all descriptors of LLP are configured, write 1 to to this bit. The core will clear this bit when llp transction is finished ;
2	RW	0x0	pause LLP DMA Pause Write 1 to Pause DMA channel . DMA will complete current burst transfer and then pause. All register stay unchange. If software write 0 later , It will continue to work. It may take several cycles to pause. 1: pause; 0: continue to work ;
1	W1C	0x0	stop LLP DMA Stop Write 1 to stop DMA channel. DMA will complete current burst transter and then stop. It may takes several cycles. 1: stop ; 0: no effect ;
0	W1C	0x0	start LLP DMA start Write 1 to start DMA Channel , self clear after 1 cycle. 1: start ; 0: no effect

TSP_PTIx_LLP_BASE

Address: Operational Base + offset (0x0108)

LLP Descriptor BASE Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	addr LLP Descriptor BASE Address LLP Descriptor BASE address

TSP_PTIx_LLP_WRITE

Address: Operational Base + offset (0x010c)

LLP DMA Writing Software Descriptor Counter

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	counter LLP DMA Writing Software Descriptor Counter LLP DMA Writing Software Descriptor Counter

TSP_PTIx_LLP_READ

Address: Operational Base + offset (0x0110)

LLP DMA Reading Hardware Descriptor Counter

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	counter LLP DMA Reading Hardware Descriptor Counter LLP DMA Reading Hardware Descriptor Counter

TSP_PTIx_PID_STS0

Address: Operational Base + offset (0x0114)

PTI PID Channel Status 0 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_done PID31 Channel Status 1 means done
30	W1C	0x0	pid30_done PID30 Channel Status 1 means done
29	W1C	0x0	pid29_done PID29 Channel Status 1 means done
28	W1C	0x0	pid28_done PID28 Channel Status 1 means done
27	W1C	0x0	pid27_done PID27 Channel Status 1 means done
26	W1C	0x0	pid26_done PID26 Channel Status 1 means done
25	W1C	0x0	pid25_done PID25 Channel Status 1 means done
24	W1C	0x0	pid24_done PID24 Channel Status 1 means done

Bit	Attr	Reset Value	Description
23	W1C	0x0	pid23_done PID23 Channel Status 1 means done
22	W1C	0x0	pid22_done PID22 Channel Status 1 means done
21	W1C	0x0	pid21_done PID21 Channel Status 1 means done
20	W1C	0x0	pid20_done PID20 Channel Status 1 means done
19	W1C	0x0	pid19_done PID19 Channel Status 1 means done
18	W1C	0x0	pid18_done PID18 Channel Status 1 means done
17	W1C	0x0	pid17_done PID17 Channel Status 1 means done
16	W1C	0x0	pid16_done PID16 Channel Status 1 means done
15	W1C	0x0	pid15_done PID15 Channel Status 1 means done
14	W1C	0x0	pid14_done PID14 Channel Status 1 means done
13	W1C	0x0	pid13_done PID13 Channel Status 1 means done
12	W1C	0x0	pid12_done PID12 Channel Status 1 means done
11	W1C	0x0	pid11_done PID11 Channel Status 1 means done
10	W1C	0x0	pid10_done PID10 Channel Status 1 means done
9	W1C	0x0	pid9_done PID9 Channel Status 1 means done

Bit	Attr	Reset Value	Description
8	W1C	0x0	pid8_done PID8 Channel Status 1 means done
7	W1C	0x0	pid7_done PID7 Channel Status 1 means done
6	W1C	0x0	pid6_done PID6 Channel Status 1 means done
5	W1C	0x0	pid5_done PID5 Channel Status 1 means done
4	W1C	0x0	pid4_done PID4 Channel Status 1 means done
3	W1C	0x0	pid3_done PID3 Channel Status 1 means done
2	RW	0x0	pid2_done PID2 Channel Status 1 means done
1	W1C	0x0	pid1_done PID1 Channel Status 1 means done
0	W1C	0x0	pid0_done PID0 Channel Status 1 means done

TSP_PTIx_PID_STS1
Address: Operational Base + offset (0x0118)
PTI PID Channel Status 1 Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	pid63_done PID63 Channel Status 1 means done
30	W1C	0x0	pid62_done PID62 Channel Status 1 means done
29	W1C	0x0	pid61_done PID61 Channel Status 1 means done
28	W1C	0x0	pid60_done PID60 Channel Status 1 means done

Bit	Attr	Reset Value	Description
27	W1C	0x0	pid59_done PID59 Channel Status 1 means done
26	W1C	0x0	pid58_done PID58 Channel Status 1 means done
25	W1C	0x0	pid57_done PID57 Channel Status 1 means done
24	W1C	0x0	pid56_done PID56 Channel Status 1 means done
23	W1C	0x0	pid55_done PID55 Channel Status 1 means done
22	W1C	0x0	pid54_done PID54 Channel Status 1 means done
21	W1C	0x0	pid53_done PID53 Channel Status 1 means done
20	W1C	0x0	pid52_done PID52 Channel Status 1 means done
19	W1C	0x0	pid51_done PID51 Channel Status 1 means done
18	W1C	0x0	pid50_done PID51 Channel Status 1 means done
17	W1C	0x0	pid49_done PID49 Channel Status 1 means done
16	W1C	0x0	pid48_done PID48 Channel Status 1 means done
15	W1C	0x0	pid47_done PID47 Channel Status 1 means done
14	W1C	0x0	pid46_done PID46 Channel Status 1 means done
13	W1C	0x0	pid45_done PID45 Channel Status 1 means done

Bit	Attr	Reset Value	Description
12	W1 C	0x0	pid44_done PID44 Channel Status 1 means done
11	W1 C	0x0	pid43_done PID43 Channel Status 1 means done
10	W1 C	0x0	pid42_done PID42 Channel Status 1 means done
9	W1 C	0x0	pid41_done PID41 Channel Status 1 means done
8	W1 C	0x0	pid40_done PID40 Channel Status 1 means done
7	W1 C	0x0	pid39_done PID39 Channel Status 1 means done
6	W1 C	0x0	pid38_done PID38 Channel Status 1 means done
5	W1 C	0x0	pid37_done PID37 Channel Status 1 means done
4	W1 C	0x0	pid36_done PID36 Channel Status 1 means done
3	RW	0x0	pid35_done PID35 Channel Status 1 means done
2	W1 C	0x0	pid34_done PID34 Channel Status 1 means done
1	W1 C	0x0	pid33_done PID33 Channel Status 1 means done
0	RW	0x0	pid32_done PID32 Channel Status 1 means done

TSP_PTIx_PID_STS2
Address: Operational Base + offset (0x011c)
PTI PID Channel Status 2 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_error PID31 Error Interrupt Status 1 means error detected
30	W1C	0x0	pid30_error PID30 Error Interrupt Status 1 means error detected
29	W1C	0x0	pid29_error PID29 Error Interrupt Status 1 means error detected
28	W1C	0x0	pid28_error PID28 Error Interrupt Status 1 means error detected
27	W1C	0x0	pid27_error PID27 Error Interrupt Status 1 means error detected
26	W1C	0x0	pid26_error PID26 Error Interrupt Status 1 means error detected
25	W1C	0x0	pid25_error PID25 Error Interrupt Status 1 means error detected
24	W1C	0x0	pid24_error PID24 Error Interrupt Status 1 means error detected
23	W1C	0x0	pid23_error PID23 Error Interrupt Status 1 means error detected
22	W1C	0x0	pid22_error PID22 Error Interrupt Status 1 means error detected
21	W1C	0x0	pid21_error PID21 Error Interrupt Status 1 means error detected
20	W1C	0x0	pid20_error PID20 Error Interrupt Status 1 means error detected
19	W1C	0x0	pid19_error PID19 Error Interrupt Status 1 means error detected
18	W1C	0x0	pid18_error PID18 Error Interrupt Status 1 means error detected
17	W1C	0x0	pid17_error PID17 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
16	W1C	0x0	pid16_error PID16 Error Interrupt Status 1 means error detected
15	W1C	0x0	pid15_error PID15 Error Interrupt Status 1 means error detected
14	W1C	0x0	pid14_error PID14 Error Interrupt Status 1 means error detected
13	W1C	0x0	pid13_error PID13 Error Interrupt Status 1 means error detected
12	W1C	0x0	pid12_error PID12 Error Interrupt Status 1 means error detected
11	W1C	0x0	pid11_error PID11 Error Interrupt Status 1 means error detected
10	W1C	0x0	pid10_error PID10 Error Interrupt Status 1 means error detected
9	W1C	0x0	pid9_error PID9 Error Interrupt Status 1 means error detected
8	W1C	0x0	pid8_error PID8 Error Interrupt Status 1 means error detected
7	W1C	0x0	pid7_error PID7 Error Interrupt Status 1 means error detected
6	W1C	0x0	pid6_error PID6 Error Interrupt Status 1 means error detected
5	W1C	0x0	pid5_error PID5 Error Interrupt Status 1 means error detected
4	W1C	0x0	pid4_error PID4 Error Interrupt Status 1 means error detected
3	W1C	0x0	pid3_error PID3 Error Interrupt Status 1 means error detected
2	W1C	0x0	pid2_error PID2 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
1	W1C	0x0	pid1_error PID1 Error Interrupt Status 1 means error detected
0	W1C	0x0	pid0_error PID0 Error Interrupt Status 1 means error detected

TSP_PTIx_PID_STS3
Address: Operational Base + offset (0x0120)
PTI PID Channel Status 3 Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	pid63_error PID63 Error Interrupt Status 1 means error detected
30	W1C	0x0	pid62_error PID62 Error Interrupt Status 1 means error detected
29	W1C	0x0	pid61_error PID61 Error Interrupt Status 1 means error detected
28	W1C	0x0	pid60_error PID60 Error Interrupt Status 1 means error detected
27	W1C	0x0	pid59_error PID59 Error Interrupt Status 1 means error detected
26	W1C	0x0	pid58_error PID58 Error Interrupt Status 1 means error detected
25	W1C	0x0	pid57_error PID57 Error Interrupt Status 1 means error detected
24	W1C	0x0	pid56_error PID56 Error Interrupt Status 1 means error detected
23	W1C	0x0	pid55_error PID55 Error Interrupt Status 1 means error detected
22	W1C	0x0	pid54_error PID54 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
21	W1C	0x0	pid53_error PID53 Error Interrupt Status 1 means error detected
20	W1C	0x0	pid52_error PID52 Error Interrupt Status 1 means error detected
19	W1C	0x0	pid51_error PID51 Error Interrupt Status 1 means error detected
18	W1C	0x0	pid50_error PID50 Error Interrupt Status 1 means error detected
17	W1C	0x0	pid49_error PID49 Error Interrupt Status 1 means error detected
16	W1C	0x0	pid48_error PID48 Error Interrupt Status 1 means error detected
15	W1C	0x0	pid47_error PID47 Error Interrupt Status 1 means error detected
14	W1C	0x0	pid46_error PID46 Error Interrupt Status 1 means error detected
13	W1C	0x0	pid45_error PID45 Error Interrupt Status 1 means error detected
12	W1C	0x0	pid44_error PID44 Error Interrupt Status 1 means error detected
11	W1C	0x0	pid43_error PID43 Error Interrupt Status 1 means error detected
10	W1C	0x0	pid42_error PID42 Error Interrupt Status 1 means error detected
9	W1C	0x0	pid41_error PID41 Error Interrupt Status 1 means error detected
8	W1C	0x0	pid40_error PID40 Error Interrupt Status 1 means error detected

Bit	Attr	Reset Value	Description
7	W1C	0x0	pid39_error PID39 Error Interrupt Status 1 means error detected
6	W1C	0x0	pid38_error PID38 Error Interrupt Status 1 means error detected
5	W1C	0x0	pid37_error PID37 Error Interrupt Status 1 means error detected
4	W1C	0x0	pid36_error PID36 Error Interrupt Status 1 means error detected
3	W1C	0x0	pid35_error PID35 Error Interrupt Status 1 means error detected
2	W1C	0x0	pid34_error PID34 Error Interrupt Status 1 means error detected
1	W1C	0x0	pid33_error PID33 Error Interrupt Status 1 means error detected
0	W1C	0x0	pid32_error PID32 Error Interrupt Status 1 means error detected

TSP_PTIx_PID_INT_ENA0
Address: Operational Base + offset (0x0124)
PID Interrupt Enable Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_done_ena PID31 Done Enable 1:enabled 0:disabled
30	RW	0x0	pid30_done_ena PID30 Done Enable 1:enabled 0:disabled
29	RW	0x0	pid29_done_ena PID29 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
28	RW	0x0	pid28_done_ena PID28 Done Enable 1:enabled 0:disabled
27	RW	0x0	pid27_done_ena PID27 Done Enable 1:enabled 0:disabled
26	RW	0x0	pid26_done_ena PID26 Done Enable 1:enabled 0:disabled
25	RW	0x0	pid25_done_ena PID25 Done Enable 1:enabled 0:disabled
24	RW	0x0	pid24_done_ena PID24 Done Enable 1:enabled 0:disabled
23	RW	0x0	pid23_done_ena PID23 Done Enable 1:enabled 0:disabled
22	RW	0x0	pid22_done_ena PID22 Done Enable 1:enabled 0:disabled
21	RW	0x0	pid21_done_ena PID21 Done Enable 1:enabled 0:disabled
20	RW	0x0	pid20_done_ena PID20 Done Enable 1:enabled 0:disabled
19	RW	0x0	pid19_done_ena PID19 Done Enable 1:enabled 0:disabled
18	RW	0x0	pid18_done_ena PID18 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
17	RW	0x0	pid17_done_ena PID17 Done Enable 1:enabled 0:disabled
16	RW	0x0	pid16_done_ena PID16 Done Enable 1:enabled 0:disabled
15	RW	0x0	pid15_done_ena PID15 Done Enable 1:enabled 0:disabled
14	RW	0x0	pid14_done_ena PID14 Done Enable 1:enabled 0:disabled
13	RW	0x0	pid13_done_ena PID13 Done Enable 1:enabled 0:disabled
12	RW	0x0	pid12_done_ena PID12 Done Enable 1:enabled 0:disabled
11	RW	0x0	pid11_done_ena PID11 Done Enable 1:enabled 0:disabled
10	RW	0x0	pid10_done_ena PID10 Done Enable 1:enabled 0:disabled
9	RW	0x0	pid9_done_ena PID9 Done Enable 1:enabled 0:disabled
8	RW	0x0	pid8_done_ena PID8 Done Enable 1:enabled 0:disabled
7	RW	0x0	pid7_done_ena PID7 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
6	RW	0x0	pid6_done_ena PID6 Done Enable 1:enabled 0:disabled
5	RW	0x0	pid5_done_ena PID5 Done Enable 1:enabled 0:disabled
4	RW	0x0	pid4_done_ena PID4 Done Enable 1:enabled 0:disabled
3	RW	0x0	pid3_done_ena PID3 Done Enable 1:enabled 0:disabled
2	RW	0x0	pid2_done_ena PID2 Done Enable 1:enabled 0:disabled
1	RW	0x0	pid1_done_ena PID1 Done Enable 1:enabled 0:disabled
0	RW	0x0	pid0_done_ena PID0 Done Enable 1:enabled 0:disabled

TSP_PTIx_PID_INT_ENA1

Address: Operational Base + offset (0x0128)

PID Interrupt Enable Register 1

Bit	Attr	Reset Value	Description
31	RW	0x0	pid63_done PID63 Done Enable 1:enabled 0:disabled
30	RW	0x0	pid62_done PID62 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
29	RW	0x0	pid61_done PID61 Done Enable 1:enabled 0:disabled
28	RW	0x0	pid60_done PID60 Done Enable 1:enabled 0:disabled
27	RW	0x0	pid59_done PID59 Done Enable 1:enabled 0:disabled
26	RW	0x0	pid58_done PID58 Done Enable 1:enabled 0:disabled
25	RW	0x0	pid57_done PID57 Done Enable 1:enabled 0:disabled
24	RW	0x0	pid56_done PID56 Done Enable 1:enabled 0:disabled
23	RW	0x0	pid55_done PID55 Done Enable 1:enabled 0:disabled
22	RW	0x0	pid54_done PID54 Done Enable 1:enabled 0:disabled
21	RW	0x0	pid53_done PID53 Done Enable 1:enabled 0:disabled
20	RW	0x0	pid52_done PID52 Done Enable 1:enabled 0:disabled
19	RW	0x0	pid51_done PID51 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
18	RW	0x0	pid50_done PID50 Done Enable 1:enabled 0:disabled
17	RW	0x0	pid49_done PID49 Done Enable 1:enabled 0:disabled
16	RW	0x0	pid48_done PID48 Done Enable 1:enabled 0:disabled
15	RW	0x0	pid47_done PID47 Done Enable 1:enabled 0:disabled
14	RW	0x0	pid46_done PID46 Done Enable 1:enabled 0:disabled
13	RW	0x0	pid45_done PID45 Done Enable 1:enabled 0:disabled
12	RW	0x0	pid44_done PID44 Done Enable 1:enabled 0:disabled
11	RW	0x0	pid43_done PID43 Done Enable 1:enabled 0:disabled
10	RW	0x0	pid42_done PID42 Done Enable 1:enabled 0:disabled
9	RW	0x0	pid41_done PID41 Done Enable 1:enabled 0:disabled
8	RW	0x0	pid40_done PID40 Done Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
7	RW	0x0	pid39_done PID39 Done Enable 1:enabled 0:disabled
6	RW	0x0	pid38_done PID38 Done Enable 1:enabled 0:disabled
5	RW	0x0	pid37_done PID37 Done Enable 1:enabled 0:disabled
4	RW	0x0	pid36_done PID36 Done Enable 1:enabled 0:disabled
3	RW	0x0	pid35_done PID35 Done Enable 1:enabled 0:disabled
2	RW	0x0	pid34_done PID34 Done Enable 1:enabled 0:disabled
1	RW	0x0	pid33_done PID33 Done Enable 1:enabled 0:disabled
0	RW	0x0	pid32_done PID32 Done Enable 1:enabled 0:disabled

TSP_PTIx_PID_INT_ENA2

Address: Operational Base + offset (0x012c)

PID Interrupt Enable Register 2

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_error PID31 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
30	RW	0x0	pid30_error PID30 Error Interrupt Enable 1:enabled 0:disabled
29	RW	0x0	pid29_error PID29 Error Interrupt Enable 1:enabled 0:disabled
28	RW	0x0	pid28_error PID28 Error Interrupt Enable 1:enabled 0:disabled
27	RW	0x0	pid27_error PID27 Error Interrupt Enable 1:enabled 0:disabled
26	RW	0x0	pid26_error PID26 Error Interrupt Enable 1:enabled 0:disabled
25	RW	0x0	pid25_error PID25 Error Interrupt Enable 1:enabled 0:disabled
24	RW	0x0	pid24_error PID24 Error Interrupt Enable 1:enabled 0:disabled
23	RW	0x0	pid23_error PID23 Error Interrupt Enable 1:enabled 0:disabled
22	RW	0x0	pid22_error PID22 Error Interrupt Enable 1:enabled 0:disabled
21	RW	0x0	pid21_error PID21 Error Interrupt Enable 1:enabled 0:disabled
20	RW	0x0	pid20_error PID20 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
19	RW	0x0	pid19_error PID19 Error Interrupt Enable 1:enabled 0:disabled
18	RW	0x0	pid18_error PID18 Error Interrupt Enable 1:enabled 0:disabled
17	RW	0x0	pid17_error PID17 Error Interrupt Enable 1:enabled 0:disabled
16	RW	0x0	pid16_error PID16 Error Interrupt Enable 1:enabled 0:disabled
15	RW	0x0	pid15_error PID15 Error Interrupt Enable 1:enabled 0:disabled
14	RW	0x0	pid14_error PID14 Error Interrupt Enable 1:enabled 0:disabled
13	RW	0x0	pid13_error PID13 Error Interrupt Enable 1:enabled 0:disabled
12	RW	0x0	pid12_error PID12 Error Interrupt Enable 1:enabled 0:disabled
11	RW	0x0	pid11_error PID11 Error Interrupt Enable 1:enabled 0:disabled
10	RW	0x0	pid10_error PID10 Error Interrupt Enable 1:enabled 0:disabled
9	RW	0x0	pid9_error PID9 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
8	RW	0x0	pid8_error PID8 Error Interrupt Enable 1:enabled 0:disabled
7	RW	0x0	pid7_error PID7 Error Interrupt Enable 1:enabled 0:disabled
6	RW	0x0	pid6_error PID6 Error Interrupt Enable 1:enabled 0:disabled
5	RW	0x0	pid5_error PID5 Error Interrupt Enable 1:enabled 0:disabled
4	RW	0x0	pid4_error PID4 Error Interrupt Enable 1:enabled 0:disabled
3	RW	0x0	pid3_error PID3 Error Interrupt Enable 1:enabled 0:disabled
2	RW	0x0	pid2_error PID2 Error Interrupt Enable 1:enabled 0:disabled
1	RW	0x0	pid1_error PID1 Error Interrupt Enable 1:enabled 0:disabled
0	RW	0x0	pid0_error PID0 Error Interrupt Enable 1:enabled 0:disabled

TSP_PTIx_PID_INT_ENA3
Address: Operational Base + offset (0x0130)
PID Interrupt Enable Register 3

Bit	Attr	Reset Value	Description
31	RW	0x0	pid63_error PID63 Error Interrupt Enable 1:enabled 0:disabled
30	RW	0x0	pid62_error PID62 Error Interrupt Enable 1:enabled 0:disabled
29	RW	0x0	pid61_error PID61 Error Interrupt Enable 1:enabled 0:disabled
28	RW	0x0	pid60_error PID60 Error Interrupt Enable 1:enabled 0:disabled
27	RW	0x0	pid59_error PID59 Error Interrupt Enable 1:enabled 0:disabled
26	RW	0x0	pid58_error PID58 Error Interrupt Enable 1:enabled 0:disabled
25	RW	0x0	pid57_error PID57 Error Interrupt Enable 1:enabled 0:disabled
24	RW	0x0	pid56_error PID56 Error Interrupt Enable 1:enabled 0:disabled
23	RW	0x0	pid55_error PID55 Error Interrupt Enable 1:enabled 0:disabled
22	RW	0x0	pid54_error PID54 Error Interrupt Enable 1:enabled 0:disabled
21	RW	0x0	pid53_error PID53 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
20	RW	0x0	pid52_error PID52 Error Interrupt Enable 1:enabled 0:disabled
19	RW	0x0	pid51_error PID51 Error Interrupt Enable 1:enabled 0:disabled
18	RW	0x0	pid50_error PID50 Error Interrupt Enable 1:enabled 0:disabled
17	RW	0x0	pid49_error PID49 Error Interrupt Enable 1:enabled 0:disabled
16	RW	0x0	pid48_error PID48 Error Interrupt Enable 1:enabled 0:disabled
15	RW	0x0	pid47_error PID47 Error Interrupt Enable 1:enabled 0:disabled
14	RW	0x0	pid46_error PID46 Error Interrupt Enable 1:enabled 0:disabled
13	RW	0x0	pid45_error PID45 Error Interrupt Enable 1:enabled 0:disabled
12	RW	0x0	pid44_error PID44 Error Interrupt Enable 1:enabled 0:disabled
11	RW	0x0	pid43_error PID43 Error Interrupt Enable 1:enabled 0:disabled
10	RW	0x0	pid42_error PID42 Error Interrupt Enable 1:enabled 0:disabled

Bit	Attr	Reset Value	Description
9	RW	0x0	pid41_error PID41 Error Interrupt Enable 1:enabled 0:disabled
8	RW	0x0	pid40_error PID40 Error Interrupt Enable 1:enabled 0:disabled
7	RW	0x0	pid39_error PID39 Error Interrupt Enable 1:enabled 0:disabled
6	RW	0x0	pid38_error PID38 Error Interrupt Enable 1:enabled 0:disabled
5	RW	0x0	pid37_error PID37 Error Interrupt Enable 1:enabled 0:disabled
4	RW	0x0	pid36_error PID36 Error Interrupt Enable 1:enabled 0:disabled
3	RW	0x0	pid35_error PID35 Error Interrupt Enable 1:enabled 0:disabled
2	RW	0x0	pid34_error PID34 Error Interrupt Enable 1:enabled 0:disabled
1	RW	0x0	pid33_error PID33 Error Interrupt Enable 1:enabled 0:disabled
0	RW	0x0	pid32_error PID32 Error Interrupt Enable 1:enabled 0:disabled

TSP_PTIx_PCR_INT_STS
Address: Operational Base + offset (0x0134)
PTI PCR Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	pcr7_done PCR7 Status 1: done; 0: not done;
6	W1C	0x0	pcr6_done PCR6 Status 1: done; 0: not done;
5	W1C	0x0	pcr5_done PCR5 Status 1: done; 0: not done;
4	W1C	0x0	pcr4_done PCR4 Status 1: done; 0: not done;
3	W1C	0x0	pcr3_done PCR3 Status 1: done; 0: not done;
2	W1C	0x0	pcr2_done PCR2 Status 1: done; 0: not done;
1	W1C	0x0	pcr1_done PCR1 Status 1: done; 0: not done;
0	W1C	0x0	pcr0_done PCR0 Status 1: done; 0: not done;

TSP_PTIx_PCR_INT_ENA

Address: Operational Base + offset (0x0138)

PTI PCR Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	pcr7_done_ena pcr7 done interrupt enable 1: enabled; 0: disabled;

Bit	Attr	Reset Value	Description
6	RW	0x0	pcr6_done_ena pcr6 done interrupt enable 1: enabled; 0: disabled;
5	RW	0x0	pcr5_done_ena pcr5 done interrupt enable 1: enabled; 0: disabled;
4	RW	0x0	pcr4_done_ena pcr4 done interrupt enable 1: enabled; 0: disabled;
3	RW	0x0	pcr3_done_ena pcr3 done interrupt enable 1: enabled; 0: disabled;
2	RW	0x0	pcr2_done_ena pcr2 done interrupt enable 1: enabled; 0: disabled;
1	RW	0x0	pcr1_done_ena pcr1 done interrupt enable 1: enabled; 0: disabled;
0	RW	0x0	pcr0_done_ena pcr0 done interrupt enable 1: enabled; 0: disabled;

TSP_PTIx_PCRn_CTRL

Address: Operational Base + offset (0x013c)

PID PCR Control Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:1	RW	0x0000	pid PCR Extraction PID number This 13-bit field sets the PID number that needs PCR extraction.
0	RW	0x0	on PCR Extraction Switch 1'b1: PCR extraction switched on ; 1'b0: PCR extraction switched off ;

TSP_PTIx_PCRn_H

Address: Operational Base + offset (0x015c)
High Order PCR value

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	pcr PCR[32] pcr[32]

TSP_PTIx_PCRn_L
Address: Operational Base + offset (0x0160)
Low Order PCR value

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pcr pcr[31:0] pcr[31:0]

TSP_PTIx_DMA_STS
Address: Operational Base + offset (0x019c)
LLP DMA Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	llp_error LLP DMA Error Status 1: error response during DMA transaction; 0: no error response during DMA transaction;
0	W1C	0x0	llp_done LLP DMA Done Status 1: DMA transaction completed; 0: DMA transaction not completed;

TSP_PTIx_DMA_ENA
Address: Operational Base + offset (0x01a0)
DMA Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	llp_error_ena LLP DMA Error Interrupt Enable 1: enabled 0: disabled
0	RW	0x0	llp_done_ena LLP DMA Done Interrupt Enable 1: enabled 0: disabled

TSP_PTIx_DATA_FLAG0
Address: Operational Base + offset (0x01a4)
PTI_PID_WRITE Flag 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data_write_flag_0 From PID0 TO PID31

TSP_PTIx_DATA_FLAG1
Address: Operational Base + offset (0x01a8)
PTI_PID_WRITE Flag 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data_write_flag_1 From PID32 TO PID63

TSP_PTIx_LIST_FLAG
Address: Operational Base + offset (0x01ac)
PTIx_LIST_WRITE Flag

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	list_write_flag From PID0 TO PID15

TSP_PTIx_DST_STS0
Address: Operational Base + offset (0x01b0)
PTI Destination Status Register

Bit	Attr	Reset Value	Description
31:0	W1C	0x00000000	demux_dma_status_0 From 0 to 31 channel

TSP_PTIx_DST_STS1
Address: Operational Base + offset (0x01b4)
PTI Destination Status Register

Bit	Attr	Reset Value	Description
31:0	W1C	0x00000000	demux_dma_status_0 From 32 to 63 channel

TSP_PTIx_DST_ENA0
Address: Operational Base + offset (0x01b8)
PTI Destination Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	demux_dma_enable_0 From 0 to 31 channel

TSP_PTIx_DST_ENA1

Address: Operational Base + offset (0x01bc)

PTI Destination Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	demux_dma_enable_1 From 32 to 63 channel

TSP_PTIx_ECWn_H

Address: Operational Base + offset (0x0200)

The Even Control Word High Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ecw_h The Even Control Word High Order ECW[63:32]

TSP_PTIx_ECWn_L

Address: Operational Base + offset (0x0204)

The Even Control Word Low Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ecw_l The Even Control Word Low Order ECW[31:0]

TSP_PTIx_OCWn_H

Address: Operational Base + offset (0x0208)

The Odd Control Word High Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ocw_h The Odd Control Word High order OCW[63:32]

TSP_PTIx_OCWn_L

Address: Operational Base + offset (0x020c)

The Odd Control Word Low Order

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ocw_l The Odd Control Word Low Order OCW[31:0]

TSP_PTIx_PIDn_CTRL
Address: Operational Base + offset (0x0300)
PID Channel Control Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	cw_num Control Word Order Number This fields indicates the corresponding order number of control word to be used to descramble TS packets.
15:3	RW	0x0000	pid PID number This 13-bit sets the desired PID number to be processed by PTI channel.
2	RW	0x0	csa_on Descrambling Switch 1'b1: Descrambling function turned on; 1'b0: Descrambling function turned off;
1	R/W SC	0x0	clear PID Channel Clear Write 1 to clear PID channel. This bit will be set to 0 if the channel is clear.
0	R/W SC	0x0	en PID Channel Enable Write 1 to enable channel. Write 0 to this bit will not take any effect. This bit will be 0 when channel is cleared.

TSP_PTIx_PIDn_BASE
Address: Operational Base + offset (0x0400)
PTI Data Memory Buffer Base Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Base Address PTI Data Memory Buffer Base Address

TSP_PTIx_PIDn_TOP
Address: Operational Base + offset (0x0404)
PTI Data Memory Buffer Top Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Top Address PTI Data Memory Buffer Top Address

TSP_PTIx_PIDn_WRITE

Address: Operational Base + offset (0x0408)

PTI Data Memory Buffer Hardware Writing Address

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	address PTI Data Memory Buffer Hardware Writing Address PTI Data Memory Buffer Hardware Writing Address

TSP_PTIx_PIDn_READ

Address: Operational Base + offset (0x040c)

PTI Data Memory Buffer Software Reading Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Software Reading Address PTI Data Memory Buffer Software Reading Address

TSP_PTIx_LISTn_BASE

Address: Operational Base + offset (0x0800)

PTI List Memory Buffer Base Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI Data Memory Buffer Software Reading Address PTI Data Memory Buffer Software Reading Address

TSP_PTIx_LISTn_TOP

Address: Operational Base + offset (0x0804)

PTI List Memory Buffer Top Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Top Address PTI List Memory Buffer Top Address

TSP_PTIx_LISTn_WRITE

Address: Operational Base + offset (0x0808)

PTI List Memory Buffer Hardware Writing Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Hardware Writing Address PTI List Memory Buffer Hardware Writing Address

TSP_PTIx_LISTn_READ

Address: Operational Base + offset (0x080c)

PTI List Memory Buffer Software Reading Address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	address PTI List Memory Buffer Software Reading Address PTI List Memory Buffer Software Reading Address

TSP_PTIx_PIDn_CFG

Address: Operational Base + offset (0x0900)

PID Demux Configure Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	filter_en Filter Byte Enable The proper position of filter byte Enable. For Section filter. the 1st,4th,5th,..18th byte of section header are used to be filtered; For PES filter, the 4th,7th,8th...21th byte of pes header are used to be filtered.
15:12	RO	0x0	reserved
11	RW	0x0	scd_en Start Code Detection Switch Start code detection 1: enabled; 0: disabled; This bit is only valid when n < 16.
10	RW	0x0	cni_on Current Next Indicator Abort when current_next_indicator == 1'b1, 1'b1: abort ; 1'b0: do nothing ;
9:8	RW	0x0	filt_mode Section Filter Mode Filter Mode when the filter mode is configured as section filter. 2'b00: stop per unit; 2'b01: full stop; 2'b10: recycle, update when version number change 2'b11: reserverd

Bit	Attr	Reset Value	Description
7:6	RW	0x0	video_type Video filtering Type 2'b00: MPEG2 2'b01: H264 2'b10: VC-1 2'b11: Reserved
5:4	RW	0x0	filt_type Filter Type 2'b00: section filtering; 2'b01: pes filtering; 2'b10: es filtering; 2'b11: ts filtering; if n>=16, it is reserved as only section filtering, other values are invalid.
3	RW	0x1	cc_abort Continue Counter Error Abort when continuity counter error happens: 1: abort; 0: do nothing;
2	RW	0x0	tei_abort Ts_error_indicator Abort when ts_error_indicator == 1: 1'b1: abort ; 1'b0: do nothing;
1	RW	0x0	crc_abort CRC Error Abort This bit is valid only when crc_on == 1'b1. When crc error happens, 1'b1: abort ; 1'b0: do nothing.
0	RW	0x0	crc_on CRC Check 1'b1: CRC check function turned on 1'b0: CRC check function turned off

TSP_PTIx_PIDn_FILT_0
Address: Operational Base + offset (0x0904)
Fliter Word 0

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 6th byte of section header or 9th byte of pes header

Bit	Attr	Reset Value	Description
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 5th byte of section header or 8th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 4th byte of section header or 7th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 1st byte of section header or 4th byte of pes header

TSP_PTIx_PIDn_FILT_1

Address: Operational Base + offset (0x0908)

Fliter Word 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 10th byte of section header or 13rd byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 9th byte of section header or 12nd byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 8th byte of section header or 11st byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 7th byte of section header or 10th byte of pes header

TSP_PTIx_PIDn_FILT_2

Address: Operational Base + offset (0x090c)

Fliter Word 2

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 14th byte of section header or 17th byte of pes header

Bit	Attr	Reset Value	Description
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 13rd byte of section header or 16th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 12nd byte of section header or 15th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 11st byte of section header or 14th byte of pes header

TSP_PTIx_PIDn_FILT_3
Address: Operational Base + offset (0x0910)
Fliter Word 3

Bit	Attr	Reset Value	Description
31:24	RW	0x00	filt_byte_3 Fliter Byte 2 This byte refers to 18th byte of section header or 21st byte of pes header
23:16	RW	0x00	filt_byte_2 Fliter Byte 2 This byte refers to 17th byte of section header or 20th byte of pes header
15:8	RW	0x00	filt_byte_1 Fliter Byte 1 This byte refers to 16th byte of section header or 19th byte of pes header
7:0	RW	0x00	filt_byte_0 Fliter Byte 0 This byte refers to 15th byte of section header or 18th byte of pes header

15.4.3 MMU Register Summary

Name	Offset	Size	Reset Value	Description
TSP_MMU_DTE_ADDR	0x08800	W	0x00000000	MMU current page Table address
TSP_MMU_STATUS	0x08804	W	0x00000018	MMU status register
TSP_MMU_COMMAND	0x08808	W	0x00000000	MMU command register
TSP_MMU_PAGE_FAULT_ADDR	0x0880c	W	0x00000000	MMU logical address of last page fault
TSP_MMU_ZAP_ONE_LINE	0x08810	W	0x00000000	MMU Zap cache line register

Name	Offset	Size	Reset Value	Description
TSP_MMU_INT_RAWSTAT	0x08814	W	0x00000000	MMU raw interrupt status register
TSP_MMU_INT_CLEAR	0x08818	W	0x00000000	MMU interrupt clear register
TSP_MMU_INT_MASK	0x0881c	W	0x00000000	MMU interrupt mask register
TSP_MMU_INT_STATUS	0x08820	W	0x00000000	MMU interrupt status register
TSP_MMU_AUTO_GATING	0x08824	W	0x00000001	MMU auto gating
TSP_MMU_MISS_CNT	0x08828	W	0x00000000	MMU miss counter
TSP_MMU_BURST_CNT	0x0882c	W	0x00000000	MMU burst counter

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

15.4.4 MMU Detail Register Description

TSP_MMU_DTE_ADDR

Address: Operational Base + offset (0x08800)

MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR MMU dte base addr MMU dte base addr , the address must be 4kb aligned

TSP_MMU_STATUS

Address: Operational Base + offset (0x08804)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID Index of master reponsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x1	REPLAY_BUFFER_EMPTY The MMU replay buffer is empty
3	RO	0x1	MMU_IDLE The MMU is idle when accesses are being translated and there are no unfinished translated accesses.
2	RO	0x0	STAIL_ACTIVE MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE MMU page fault mode currently enabled . The mode is enabled by command.

Bit	Attr	Reset Value	Description
0	RO	0x0	PAGING_ENABLED Paging is enabled

TSP_MMU_COMMAND
Address: Operational Base + offset (0x08808)
MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	WO	0x0	MMU_CMD MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

TSP_MMU_PAGE_FAULT_ADDR
Address: Operational Base + offset (0x0880c)
MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR address of last page fault

TSP_MMU_ZAP_ONE_LINE
Address: Operational Base + offset (0x08810)
MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE address to be invalidated from the page table cache

TSP_MMU_INT_RAWSTAT
Address: Operational Base + offset (0x08814)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error
0	RW	0x0	PAGE_FAULT page fault

TSP_MMU_INT_CLEAR

Address: Operational Base + offset (0x08818)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	WO	0x0	READ_BUS_ERROR read bus error
0	WO	0x0	PAGE_FAULT page fault

TSP_MMU_INT_MASK

Address: Operational Base + offset (0x0881c)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error enable an interrupt source if the corresponding mask bit is set to 1
0	RW	0x0	PAGE_FAULT page fault enable an interrupt source if the corresponding mask bit is set to 1

TSP_MMU_INT_STATUS

Address: Operational Base + offset (0x08820)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error
0	RO	0x0	PAGE_FAULT page fault

TSP_MMU_AUTO_GATING

Address: Operational Base + offset (0x08824)

mmu auto gating

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_gating when it is 1'b1, the mmu will auto gating it self

TSP_MMU_mmu_miss_cnt
Address: Operational Base + offset (0x08828)
Register0000 Abstract

Bit	Attr	Reset Value	Description
31	RW	0x0	cnt_ctrl_sel sel the counter for mmu_miss or mmu_real_miss 1'b0: mmu_real_miss 1'b1: mmu_miss When sel 1'b1, an axi command miss may count for several times; when sel 1'b0, an axi command only count for a time
30	RW	0x0	miss_cnt_overflow_flag miss cnt overflow flag
29:0	RW	0x00000000	miss_cnt count for miss AXI command

TSP_MMU_mmu_burst_cnt
Address: Operational Base + offset (0x0882c)
Register0001 Abstract

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	bust_cnt_overflow_flag
29:0	RW	0x00000000	burst_cnt The AXI input burst counter

15.5 Interface Description

Table 15-1 TSP interface description

Module Pin	Dir	Pad Name	IOMUX Setting
IOMUX0			
tsp_valid	I	IO_TSPvalidm0_CIFvsyncm0_SDMMC0EXTcmd_SPIclkm2_USB3PHYdebug1_I2S2sclkm1_GPIO3A0vccio6	GRF_GPIO3AL_iomux [2:0] = 3'b001
tsp_fail	I	IO_TSPfail_CIFhref_SDMMC0EXTdet_SPItxdm2_USB3PHYdebug2_I2S2sdom1_GPIO3A1vccio6	GRF_GPIO3AL_iomux [5:3] = 3'b001
tsp_clk	I	IO_TSPclk_CIFclkin_SDMMC0EXTclkout_SPIrxdm2_USB3PHYdebug3_I2S2sdim1_GPIO3A2vccio6	GRF_GPIO3AL_iomux [8:6] = 3'b001
tsp_syncm0	I	IO_TSPsync_CIFclkout_SDMMC0EXTwp_GPIO3A3vccio6	GRF_GPIO3AL_iomux [11:9] = 3'b001
tsp_d0	I	IO_TSPd0_CIFda0_SDMMC0EXTd0_UART1tx_USB3PHYdebug4_GPIO3A4vccio6	GRF_GPIO3AL_iomux [14:12] = 3'b001

Module Pin	Dir	Pad Name	IOMUX Setting
tsp_d1	I	IO_TSPd1_CIFdata1_SDMMC0EXTd1_UART1rtsn_USB3PHYdebug5_GPIO3A5vccio6	GRF_GPIO3AH_iomux [2:0] = 3'b001
tsp_d2	I	IO_TSPd2_CIFdata2_SDMMC0EXTd2_UART1rx_USB3PHYdebug6_GPIO3A6vccio6	GRF_GPIO3AH_iomux [5:3] = 3'b001
tsp_d3	I	IO_TSPd3_CIFdata3_SDMMC0EXTd3_UART1ctsn_USB3PHYdebug7_GPIO3A7vccio6	GRF_GPIO3AH_iomux [8:6] = 3'b001
tsp_d4	I	IO_TSPd4_CIFdata4_SPIcsn0m2_I2S2lrcktxm1_USB3PHYdebug8_I2S2lrckrxm1_GPIO3B0vccio6	GRF_GPIO3BL_iomux [2:0] = 3'b001
tsp_d5m0	I	IO_TSPd5m0_CIFdata5m0_GPIO3B1vccio6	GRF_GPIO3BH_iomux [3:2] = 2'b01
tsp_d6m0	I	IO_TSPd6m0_CIFdata6m0_GPIO3B2vccio6	GRF_GPIO3BH_iomux [5:4] = 2'b01
tsp_d7m0	I	IO_TSPd7m0_CIFdata7m0_GPIO3B3vccio6	GRF_GPIO3BH_iomux [7:6] = 2'b01
IOMUX1			
tsp_syncm1	I	IO_I2S1mclk_NOuse0_TSPsyncm1_CIFclkoutm1_GPIO2B7vccio5	GRF_GPIO2BH_iomux [2:0] = 3'b011
tsp_d5m1	I	IO_I2S1lrckrx_NOuse1_TSPd5m1_CIFdata5m1_GPIO2C0vccio5	GRF_GPIO2CL_iomux [2:0] = 3'b011
tsp_d6m1	I	IO_I2S1lrcktx_SPDIFtxm1_TSPd6m1_CIFdata6m1_GPIO2C1vccio5	GRF_GPIO2CL_iomux [5:3] = 3'b011
tsp_d7m1	I	IO_I2S1sclk_PDMclkm0_TSPd7m1_CIFdata7m1_GPIO2C2vccio5	GRF_GPIO2CL_iomux [8:6] = 3'b011

There are two groups of IO for tsp_sync and tsp_data[7:5]. Which group of IO to be used is controlled by GRF_IOMUX_CON[8] , this bit has a default value 1'b0. If this bit is set to 1'b1, the second group of IO is selected.

15.6 Application Notes

15.6.1 Overall Operation Sequence

- Enable desired modules to work by writing correspond bit with ‘1’ in TSP_GCFG. Note: it is important to do this step at first, otherwise writing the corresponding registers will not take effect.
- Set up TS configuration by writing corresponding registers.
- Wait for the interrupts to pick up the desired TS packets following the rules detailed in the following section.

15.6.2 TS Source

TS source can be chosen by writing the bit 9 of TSP_PTIx_CTRL(x=0,1), ‘1’ for demodulator, ‘0’ for local memory.

1.TS_IN Interface

Writing bit 10 of TSP_PTIx_CTRL to choose bit ordering, and writing bit [12:11] to choose input TS mode.

TS_IN interface supports 4 input TS stream mode: sync/valid serial mode, sync/valid parallel mode, sync/burst parallel mode, nosync/valid parallel mode.

2.Local Memory

PTI also can process the TS data read from local memory by using LLP DMA mode.

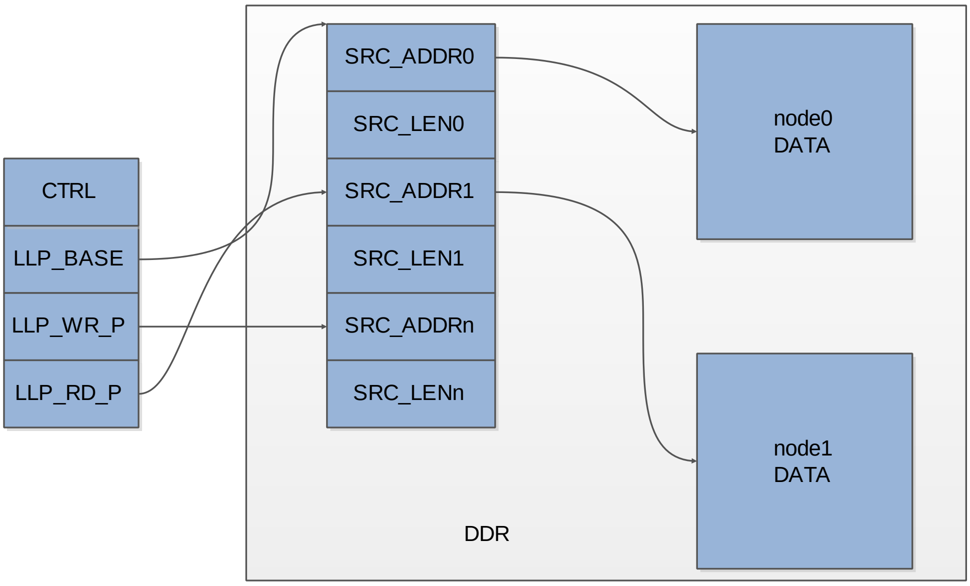


Fig. 1-6LLPaddress architecture

(1) Write PTIx_LLP_BASE with the list base address;
(2) Starting from the list base address, write the list nodes. One list node comprised of two words. The first word describes the TS data base address, the second one describes the length of TS data in unit of word.
(3) Write the PTIx_LLP_WRITE with the number of words that you have written in list memory. Note it is not the number of LLP nodes, so that the number you are writing should be an even one.
(4) Write PTIx_LLP_CFG with the configuration you want. Write the bit 0 with 1 to start LLP DMA. If all the list nodes are written, don't forget to write 1 to bit 3 to tell DMAC that the configuration is finished.

Note:

- *The MSB(bit7) of the 8-bit pointer in the PTIx_LLP_Write and PTIx_LLP_Read is used as the flag bit, and remaining 7 bits are used for addressing. Therefore the the pointer is referred to 7-bit space, not 8-bit space, and remember write the pointer with the correct flag bit. For example, if you have configured 63 LLP nodes and then you have to write the 64th LLP node starting from the list base address,*
- *PTIx_LLP_READ informs that how many words has been processed by LLP DMA. An interrupt may be generated when number of the processed words has reach to the threshold set in the PTIx_LLP_CFG.*
- *If you write the PTIx_LLP_Write several times in a complete DMA transaction, it is important to notice the flag bit of PTIx_LLP_Write, and never make the writing pointer catch up with the reading pointer.*

15.6.3 TS Synchronous Operation

Synchronous mode and Bypass mode can be switched by writing bit 15 of TSP_PTIx_CTRL. In the synchronous mode, 188/192/204 byte TS packets are supported and self-adjusted. Set up locked times in TSP_PTIx_CTRL to inform the successive times of TS packet header detection needs to lock the header of TS packets when in the unlocked mode, and set up unlocked times to informs the successive times of TS packet header error needs to re-lock

header of TS packets in the locked mode. It is recommended to use 2-3 as the locked times to quickly and correctly locked the header, and 2-3 as unlocked times to avoid unnecessarily entering into unlocked searching mode.
In the bypass mode, the input TS data will not be re-synchronized and directly fed into the PTI channel.

15.6.4 Descrambling Operation

Descrambler can achieve PES or TS level descrambling which conforms to the CSA v2.0.
Enable the channel you want by writing 1 to bit 0 of TSP_PTIx_PIDn_CTRL (x=0~1, n= 0~64);
Set the desired PID number
Turn on descrambling function by setting 1 to bit 2. If the corresponding CW is available or TS is required to be left undescrambled, CSA_ON bit is set to 0;
Choose corresponding Control Word by setting bit[19:16], and 16 set Control Word are available to be chosen. Don't forget Control Word should be preprared before the descrambling function is enabled.

Note: If the enabled channel is needed to be disabled, write the CLEAR bit to disabled the channel rather than write '0' to EN bit.

15.6.5 Demux Operation

Refer to TSP_PTIx_PIDn_CFG for Demux operation. The software users should be familiar with the demux knowledge.
Users should create a separate memory buffer to receive the processed data for each desired PID channel, and write the base and top address information of the memory buffer into TSP_PTIx_PIDn_BASE and TSP_PTIx_PIDn respectively. Also initial writing address and reading address, normally the same as base address, are also needed to be written into TSP_PTIx_PIDn_WRITE and TSP_PTIx_PIDn_READ respectively. For ES/PES filter, another separate memory needs to be created to store list data, which is used to assist obtaining PES/ES data. List base address, top address, initial writing address and reading address are also needed to write into corresponding registers.

Note:

For channel whose PID channel number larger than 15, the channels can only be used section filter. For others, there is no such limit. They can be configured as section filter, pes filter, es filter or ts filter.

Data memory address boundary should be aligned with word-size, and list memory address boundary should be aligned with word size. If the memory buffer is not larger to store processed data so that writing address reaches the top address, TSP will return to the base address to write data. So fetch the data in time, don't make the writing address catches up with reading address. The list memory buffer has the same issue.

Demux data obtain

A. TS filter

To obtain TS data and section data, when an desired PID done interrupt is generated, read TSP_PTIx_PIDn_READ firstly to know the address that last reading stops, and then read TSP_PTIx_PIDn_WRITE to know the address that hardware has reached. For ts data, start from the TSP_PTIx_PIDn_READ address to get the TS packet data, and stop at the address you want. However, the ending address should not catch up with writing address. It is recommended to obtain the TS data in the unit of TS packet which is 47-word size. At last, don’t forget to write the ending address into TSP_PTIx_PIDn_READ to leave a hint where current reading stops.

B. Section filter

Section filter can run three mode to meet different needs: stop-per-unit; full stop; recycle , update when version number change. The PID done interrupt will be generated after each part of a complete section is processed in the first mode, and the PID done will be generated only after the whole section is completed in the last two modes. In the frist two mode, the PID channel will be disabled after the whole section is completed. In the recycle mode, the channel will remain active and start a new section processing when the version number changes. Section filter also supports 16-byte filtering function, which can assign 1st , 4th to 18th byte to be filtered.

The process to obtain section data is similar to the process for TS data. After a PID done interrupt done is generated, refer to the corresponding PID error status register to check if the section data is correct. Read the frist word of the section start address to know the total length of the section according to the format of section data.

Section Length = {First Word[11:8], First Word[23:16]};

Total Length = Section Length;

Then start to fetch section data according to the total length. Again don’t forget to write the stopped address.

C. PES/ES filter

PES filter supports 16-byte filtering function, which can assign 4th, 7th to 21st byte to be filtered.

ES filter supports start code detection, including MPEG2 start code 0x000001b3, 0x00000100, VC-1 start code 0x0000010d, 0x000010f, H264 start code 0x00001.

To obtain the pes/es data, the assistant of list descriptor is needed.

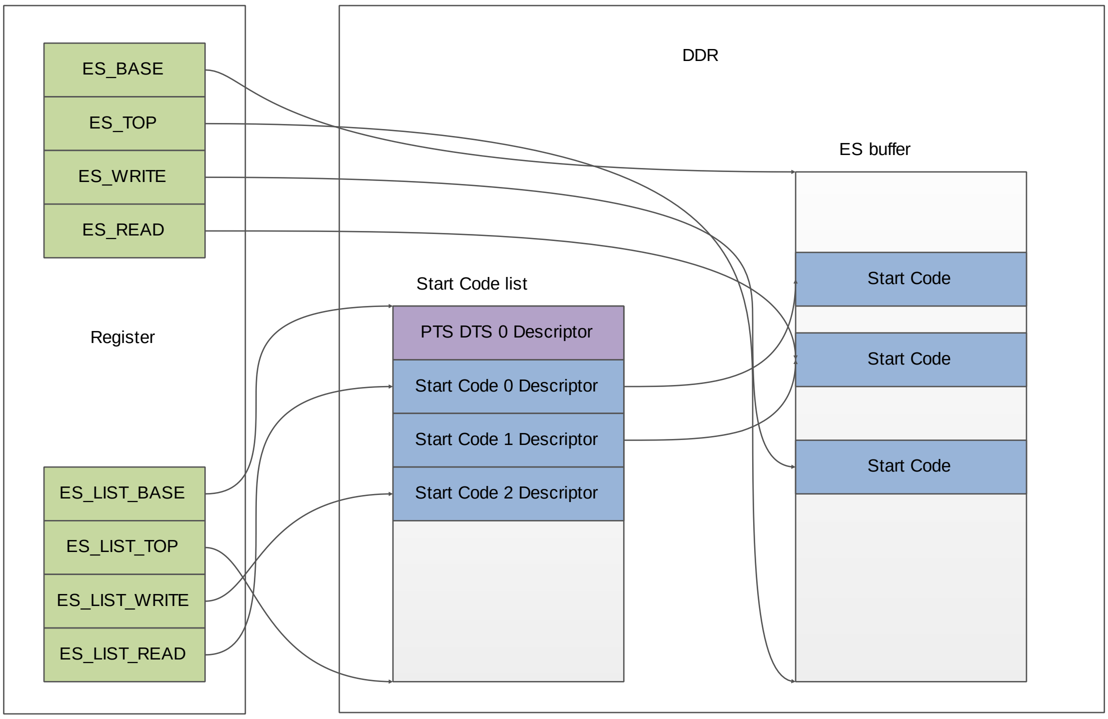


Fig. 1-7LLPmemory architecture

List memory buffer contains descriptors which contains information to obtain es/pes data which are stored in data memory buffer.
The descriptor stored in list memory buffer can be separated into two groups: PTS_DTS Descriptor and Start Code Descriptor. The descriptor is composed by 4 word content, word_0, word_1, word_2 and word_3. The word_x (x means the sequence number in a descriptor, and they are stored in the memory in sequence order). The format of the 4 words are listed as follows:

start code descriptor

Word_0:
Word_0[29:28] indicates the attributes of the bytes of the pointed word. 2’b00 means the whole word belongs to the new ES/PES packet; 2’b01 means that word[7:0] belongs to the previous packet, and the remaining bytes belong to the new packet; 2’b10 means means that word[15:0] belongs to the previous packet, and the remaining bytes belong to the new packet; 2’b11 means ’b10 means means that word[23:0] belongs to the previous packet, and the remaining bytes belong to the new packet. This pointed word is the word where start code starts, word_2 describes the location of start code.
Word_0[27:24] is equal to 0x0 in the start code descriptor. Users can used to tell two kinds of descriptor.
If the video type is H.264, word_0[23:8] means first_mb_in slice, and word_0 means nal_nuit_type.

Word_1:
the start code of stream.

Word_2:
DDR offset address in the DDR of the word where the start code is located.

Word_3:
0x0

PTS_DTS Descriptor

Word_0:
Word_0[29:28]: the same as start code descriptor
Word_0[27:24]: 0x1 in PTS_DTS descriptor.
Word_0[3] : PTS[32];
Word_0[2] : DTS[32];
Word_0[1:0] : pts_dts_flag;

Word_1:
DDR offset address of the word that valid data starts.

Word_2:
PTS[31:0]

Word_3
DTS[31:0]

To obtain PES data or ES data when start code detection is disabled, use PTS_DTS descriptor. To obtain ES data when start code detection is enabled, use start code descriptor.
When a PID done interrupt is generated, make sure there is no corresponding PID error generated. Read the TSP_PTIx_LISTn_READ to know the list reading address in the last time. Start from here, read the 4-word descriptor one by one to know the offset of the packets. Refer to the offset in the DDR where in the data memory buffer to obtain data. Finally write TSP_PTIx_LISTn_READ and TSP_PTIx_PIDn_READ with corresponding reading address.

15.6.6 PVR

PVR module provide you with the function to record the programs you want. The 4 sources can be assigned with PVR, and they are the same as TS out interface.
Assign the PVR length and PVR address, and then configure TSP_PVR_CTRL to start PVR module. If you want to stop PVR function during recording, write '1' to STOP bit (bit 0) to to TSP_PVR_CTRL to stop it. Remember to take care of the status of PVR_ON bit of TSP_GFCG when programming the PVR-related registers.

15.6.7 PCR extraction

PCR extraction can be enabled by configure PTIx_PCRn_CTRL. Then if the PID-matched TS data contain PCR field, the 33-bit PCR_base field will be written corresponding PTIx_PCRn_H and PTIx_PCRn_L registers. An interrupt will be asserted if PCR interrupt is enabled.

Chapter 16 Pulse Width Modulation (PWM)

16.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.

The PWM Module supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
  - Measures the high/low polarity effective cycles of this input waveform
  - Generates a single interrupt at the transition of input waveform polarity
  - 32-bit high polarity capture register
  - 32-bit low polarity capture register
  - 32-bit current value register
  - The capture result of channel 3 can be stored in a FIFO. The depths of FIFO is 8, and 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.
- Configurable to operate in continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - Configurable PWM output polarity in inactive state and duty period pulse polarity
  - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
  - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
  - 8-bit repeat counter for one-shot operation. One-shot operation will produce N + 1 periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
  - Continuous mode generates the waveform continuously, and does not generates any interrupts
- pre-scaled operation to bus clock and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

16.2 Block Diagram

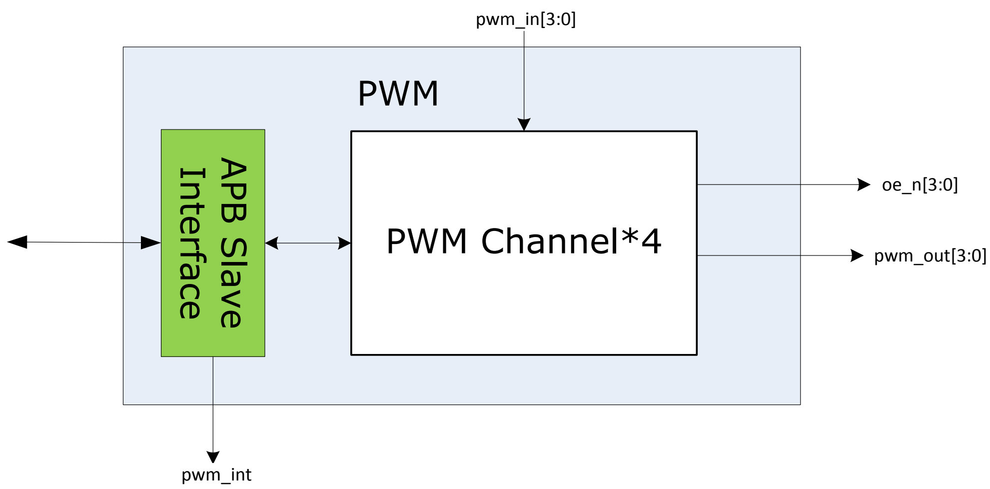


Fig. 16-1 PWM Block Diagram

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.

16.3 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.

16.3.1 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.

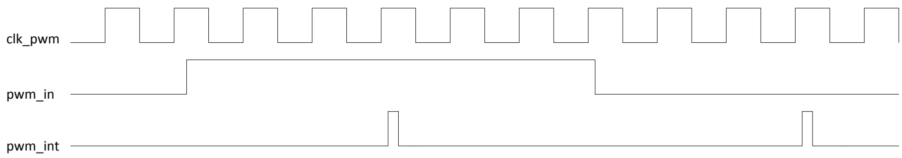


Fig. 16-2 PWM Capture Mode

16.3.2 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.

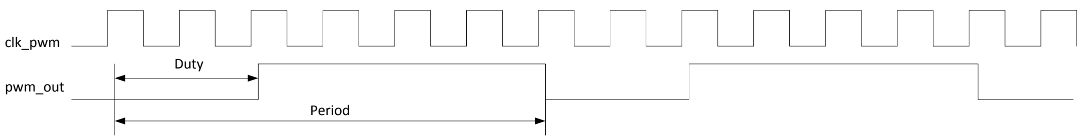


Fig. 16-3 PWM Continuous Left-aligned Output Mode

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.

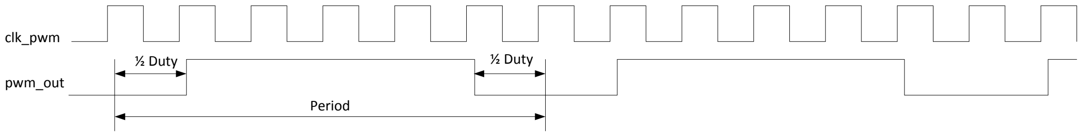


Fig. 16-4 PWM Continuous Center-aligned Output Mode

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).

16.3.3 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.

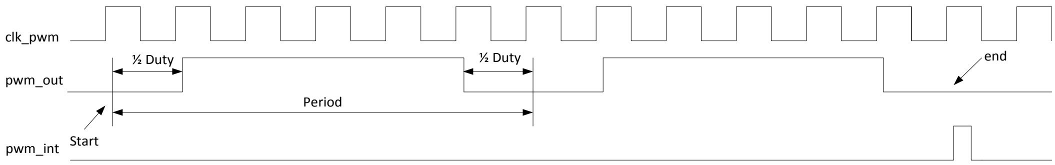


Fig. 16-5 PWM One-shot Center-aligned Output Mode

16.4 Register Description

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register

Name	Offset	Size	Reset Value	Description
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
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_PWM_FIFO_CTRL	0x0050	W	0x00000000	PWM Channel 3 FIFO Mode Control Register
PWM_PWM_FIFO_INTSTS	0x0054	W	0x00000000	FIFO Interrupts Status Register
PWM_PWM_FIFO_TOUTTHR	0x0058	W	0x00000000	FIFO Timeout Threshold Register
PWM_PWM_FIFO	0x0060	W	0x00000000	FIFO Register

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

16.4.2 Detail Register Description

PWM_PWM0_CNT

Address: Operational Base + offset (0x0000)

PWM Channel 0 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR

Address: Operational Base + offset (0x0004)

PWM Channel 0 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_DUTY_LPR

Address: Operational Base + offset (0x0008)

PWM Channel 0 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_CTRL

Address: Operational Base + offset (0x000c)

PWM Channel 0 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7	RO	0x0	reserved
6	RW	0x0	conlock pwm configure lock pwm period and duty lock to previous configuration 0: disable lock 1: enable lock
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt . 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: 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)

PWM Channel 1 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_PERIOD_HPR

Address: Operational Base + offset (0x0014)

PWM Channel 1 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

PWM Channel 1 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001c)

PWM Channel 1 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source

Bit	Attr	Reset Value	Description
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7	RO	0x0	reserved
6	RW	0x0	conlock pwm configure lock pwm period and duty lock to previous configuration 1: enable lock
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: 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)

PWM Channel 2 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR

Address: Operational Base + offset (0x0024)

PWM Channel 2 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_DUTY_LPR

Address: Operational Base + offset (0x0028)

PWM Channel 2 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_CTRL

Address: Operational Base + offset (0x002c)
PWM Channel 2 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7	RO	0x0	reserved
6	RW	0x0	conlock pwm configure lock pwm period and duty lock to previous configuration 1: enable lock
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode

Bit	Attr	Reset Value	Description
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: 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)

PWM Channel 3 Counter Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT Timer Counter The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR

Address: Operational Base + offset (0x0034)

PWM Channel 3 Period Register/High Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR Output Waveform Period/Input Waveform High Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_DUTY_LPR

Address: Operational Base + offset (0x0038)

PWM Channel 3 Duty Register/Low Polarity Capture Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR Output Waveform Duty Cycle/Input Waveform Low Polarity Cycle If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_CTRL

Address: Operational Base + offset (0x003c)

PWM Channel 3 Control Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt Repeat Counter This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale Scale Factor This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14:12	RW	0x0	prescale Prescale Factor This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel Clock Source Select 0: non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source 1: scaled clock is selected as PWM clock source
8	RW	0x0	lp_en Low Power Mode Enable 0: disabled 1: enabled When PWM channel is inactive state and Low Power Mode is enabled, the path to PWM Clock prescale module is blocked to reduce power consumption.
7	RO	0x0	reserved
6	RW	0x0	conlock pwm configure lock pwm period and duty lock to previous configuration 1: enable lock
5	RW	0x0	output_mode PWM Output mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol Inactive State Output Polarity This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol Duty Cycle Output Polarity This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode PWM Operation Mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: 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)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	CH3_Pol Channel 3 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.
10	RO	0x0	CH2_Pol Channel 2 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.

Bit	Attr	Reset Value	Description
9	RO	0x0	CH1_Pol Channel 1 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.
8	RO	0x0	CH0_Pol Channel 0 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7:4	RO	0x0	reserved
3	R/W SC	0x0	CH3_IntSts Channel 3 Interrupt Status 0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated
2	W1 C	0x0	CH2_IntSts Channel 2 Interrupt Status 0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated
1	W1 C	0x0	CH1_IntSts Channel 1 Interrupt Status 0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated
0	W1 C	0x0	CH0_IntSts Channel 0 Raw Interrupt Status 0: Channel 0 Interrupt not generated 1: Channel 0 Interrupt generated

PWM_INT_EN

Address: Operational Base + offset (0x0044)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	CH3_Int_en Channel 3 Interrupt Enable 0: Channel 3 Interrupt disabled 1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en Channel 2 Interrupt Enable 0: Channel 2 Interrupt disabled 1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en Channel 1 Interrupt Enable 0: Channel 1 Interrupt disabled 1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en Channel 0 Interrupt Enable 0: Channel 0 Interrupt disabled 1: Channel 0 Interrupt enabled

PWM_PWM_FIFO_CTRL

Address: Operational Base + offset (0x0050)

PWM Channel 3 FIFO Mode Control Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	timeout_en fifo timeout enable
8	RW	0x0	dma_mode_en dma mode enable 1'b1: enable 1'b0: disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost full Watermark level
3	RW	0x0	watermark_int_en Watermark full interrupt
2	RW	0x0	overflow_int_en FIFO Overflow Interrupt Enable When high, an interrupt asserts when the channel 3
1	RW	0x0	full_int_en FIFO Full Interrupt Enable When high, an interrupt asserts when the channel 3 FIFO is full.
0	RW	0x0	fifo_mode_sel FIFO MODE Sel When high, PWM FIFO mode is activated

PWM_PWM_FIFO_INTSTS

Address: Operational Base + offset (0x0054)

FIFO Interrupts Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	fifo_empty_status FIFO empty Status This bit indicates the FIFO is empty
3	W1C	0x0	timieout_intsts Timeout interrupt Timeout interrupt
2	W1C	0x0	fifo_watermark_full_intsts FIFO Watermark Full Interrupt Status This bit indicates the FIFO is Watermark Full
1	W1C	0x0	fifo_overflow_intsts FIFO Overflow Interrupt Status This bit indicates the FIFO is overflow
0	W1C	0x0	fifo_full_intsts FIFO Full Interrupt Status This bit indicates the FIFO is full

PWM_PWM_FIFO_TOUTTHR

Address: Operational Base + offset (0x0058)

FIFO Timeout Threshold Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	timeout_threshold FIFO Timeout value(unit pwmclk)

PWM_PWM_FIFO

Address: Operational Base + offset (0x0060)

FIFO Register

Bit	Attr	Reset Value	Description
31	RO	0x0	pol Polarity This bit indicates the polarity of the lower 31-bit counter. 0: Low 1: High
30:0	RO	0x00000000	cycle_cnt High/Low Cycle Counter This 31-bit counter indicates the effective cycles of high/low waveform.

16.5 Interface Description

Table 16-1 PWM Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
PWM0	I/O	IO_PWM0_I2C1sda_GPIO2A4vccio5	GRF_GPIO2A_IOMUX[9:8]=2’b01
PWM1	I/O	IO_PWM1_I2C1scl_GPIO2A5vccio5	GRF_GPIO2A_IOMUX[11:10]=2’b01
PWM2	I/O	IO_PWM2_GPIO2A6vccio5	GRF_GPIO2A_IOMUX[13:12]=2’b01
PWM3	I/O	IO_PWMir_POWERstate2_GPIO2A2vccio5	GRF_GPIO2A_IOMUX[5:4]=2’b01

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

16.6 Application Notes

16.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to ‘0’ to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Configure the channel to work in the capture mode.
4. Enable the INT_EN.chx_int_en to enable the interrupt generation.
5. Enable the channel by writing ‘1’ to PWMx_CTRL.pwm_en bit to start the channel.
6. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWMx_PERIOD_HPC register to know the effective high cycles of input waveforms, otherwise turn to PWMx_DUTY_LPC register to know the effective low cycles.
7. Write ‘0’ to PWMx_CTRL.pwm_en to disable the channel.

16.6.2 PWM Capture DMA Mode Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to ‘0’ to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting 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 to tansfer data from PWM to DDR.
6. Enable the channel by writing ‘1’ to PWMx_CTRL.pwm_en bit to start the channel.
7. When an dma_req is asserted, DMAC transfer the data of effective high cycles and low cycles of input waveforms to DDR.
8. Write ‘0’ to PWMx_CTRL.pwm_en to disable the channel.

16.6.3 PWM One-shot Mode/Continuous Standard Usage Flow

1. Set PWMx_CTRL.pwm_en to ‘0’ to disable the PWM channel.

2. Choose the prescale factor and the scale factor for pclk by programming PWMx_CTRL.prescale and PWMx_CTRL.scale, and select the clock needed by setting PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWMx_CTRL.duty_pol and PWMx_CTRL.inactive_pol.
4. Set the 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 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 PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working in, write '0' to PWMx_CTRL.pwm_en bit to disable the PWM channel.

16.6.4 Low-power mode

Setting PWMx_CTRL.lp_en to '1' makes the channel enter the low-power mode. When the PWM channel is inactive, the APB bus clock to the clock prescale module is gated in order to reduce the power consumption. It is recommended to disable the channel before entering the low-power mode, and quit the low-power mode before enabling the channel.

16.6.5 Other notes

When the channel is active to produce waveforms, it is free to program the PWMx_PERIOD_HPC and PWMx_DUTY_LPC register. 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 17 UART Interface

17.1 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 3 independent UART controller: UART0, UART1, UART2
- UART0/UART1/UART2 all contain two 64Bytes FIFOs for data receive and transmit
- UART0/UART1/UART2 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

17.2 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

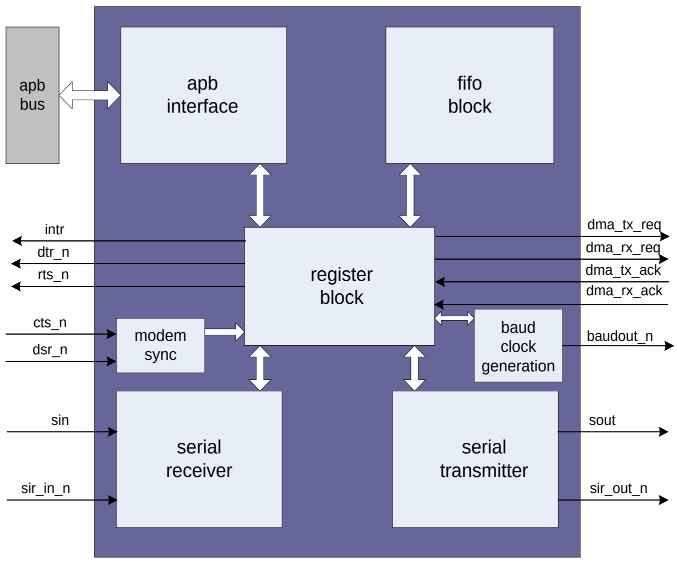


Fig. 17-1 UART Architecture

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 clockalong with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.

Serial Receiver

Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

17.3 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.

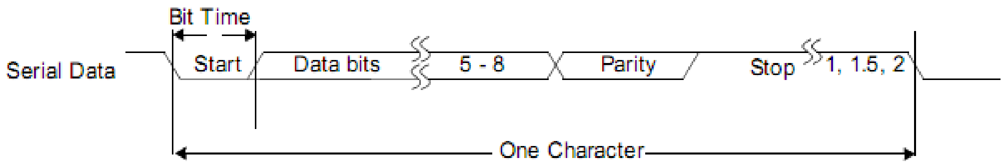


Fig. 17-2 UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional datacommunications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

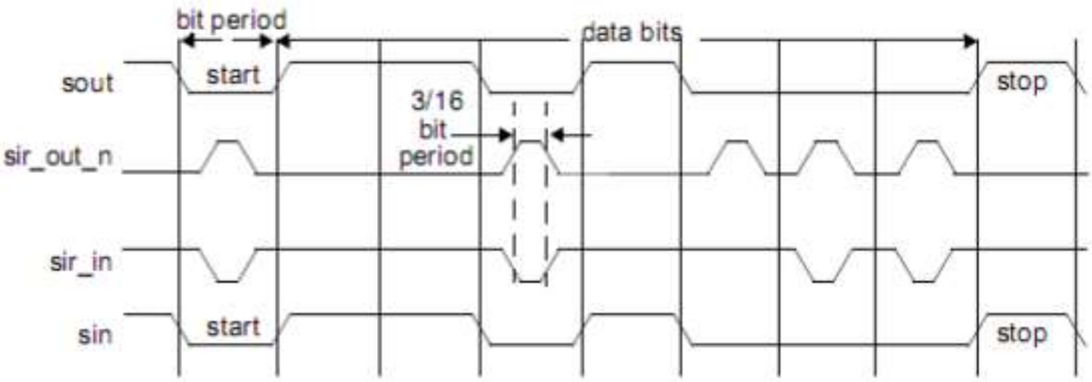


Fig. 17-3 IrDA 1.0

Baud Clock

The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

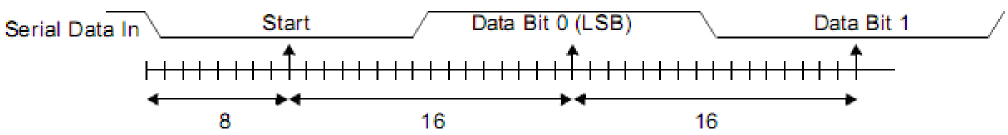


Fig. 17-4 UART baud rate

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 UART0/UART1/UART2is 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.

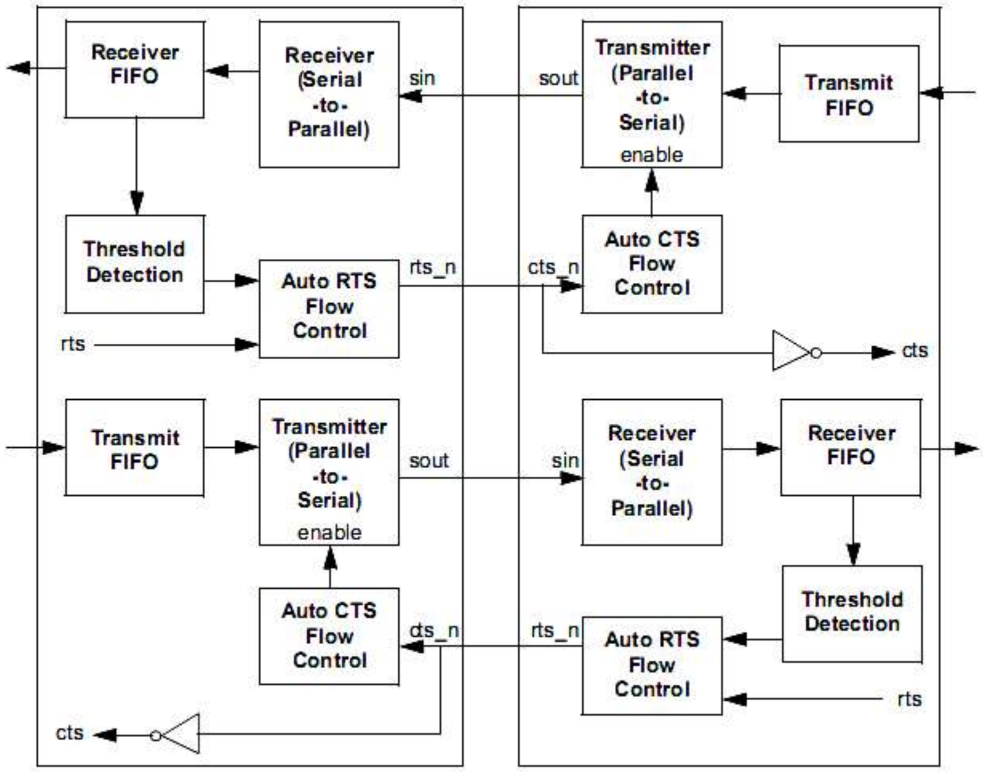


Fig. 17-5 UART Auto flow control block diagram

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)

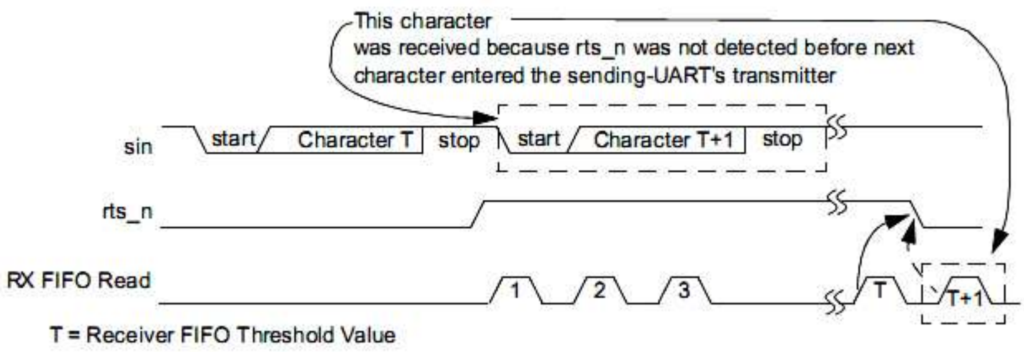


Fig. 17-6 UART AUTO RTS TIMING

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)

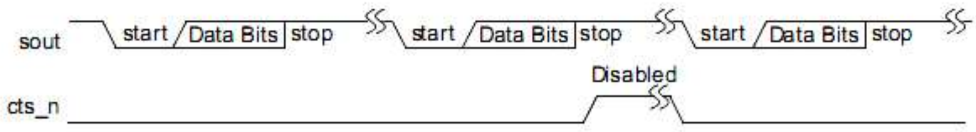


Fig. 17-7 UART AUTO CTS TIMING

17.4 Register Description

This section describes the control/status registers of the design. There are 3 UARTs in RK3328, and each one has its own base address.

17.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000000	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000000	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

Name	Offset	Size	Reset Value	Description
UART_STHR	0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000000	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x0330372a	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*

17.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_THR

Address: Operational Base + offset (0x0000)

Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

UART_DLH

Address: Operational Base + offset (0x0004)

Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER

Address: Operational Base + offset (0x0004)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
6:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt.
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

UART_IIR

Address: Operational Base + offset (0x0008)

Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
5:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RO	0x0	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

UART_FCR

Address: Operational Base + offset (0x0008)

FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO 1/4 full 10 = FIFO 1/2 full 11 = FIFO 2 less than ful
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO 1/4 full 11 = FIFO 1/2 full

Bit	Attr	Reset Value	Description
3	WO	0x0	dma_mode DMA Mode This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 0 = mode 0 1 = mode 11100 = character timeout.
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
0	WO	0x0	fifo_en FIFO Enable. FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset (0x000c)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.

Bit	Attr	Reset Value	Description
6	RW	0x0	break_ctrl Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	RO	0x0	reserved
4	RW	0x0	even_parity_sel Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked.
3	RW	0x0	parity_en Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
2	RW	0x0	stop_bits_num Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, twostop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	data_length_sel Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART_MCR

Address: Operational Base + offset (0x0010)

Modem Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	sir_mode_en SIR Mode Enable. SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode . 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)

Bit	Attr	Reset Value	Description
2	RW	0x0	out1 OUT1 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)
1	RW	0x0	req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.
0	RW	0x0	data_terminal_ready Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

UART_LSR

Address: Operational Base + offset (0x0014)

Line Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	RO	0x0	trans_empty Transmitter Empty bit. Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.

Bit	Attr	Reset Value	Description
5	RO	0x0	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

UART_MSR

Address: Operational Base + offset (0x0018)

Modem Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	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.
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR

Address: Operational Base + offset (0x001c)

Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space.

UART_SRBR

Address: Operational Base + offset (0x0030)

Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR

Address: Operational Base + offset (0x006c)

Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR.

UART_FAR

Address: Operational Base + offset (0x0070)

FIFO Access Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled

UART_TFR

Address: Operational Base + offset (0x0074)

Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	RO	0x0	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.

UART_RFW

Address: Operational Base + offset (0x0078)

Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).

Bit	Attr	Reset Value	Description
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR

Address: Operational Base + offset (0x007c)

UART Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x0	trasn_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	RO	0x0	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.

Bit	Attr	Reset Value	Description
0	RO	0x0	uart_busy UART Busy. UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 0 = UART is idle or inactivenot busy 1 = UART is busy (actively transferring data)

UART_TFL

Address: Operational Base + offset (0x0080)

Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL

Address: Operational Base + offset (0x0084)

Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR

Address: Operational Base + offset (0x0088)

Software Reset Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).

Bit	Attr	Reset Value	Description
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)

Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR

Address: Operational Base + offset (0x0090)

Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE

Address: Operational Base + offset (0x0098)

Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT

Address: Operational Base + offset (0x009c)

Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET

Address: Operational Base + offset (0x00a0)

Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX

Address: Operational Base + offset (0x00a4)

Halt TX

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

Bit	Attr	Reset Value	Description
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled

UART_DMASA

Address: Operational Base + offset (0x00a8)

DMA Software Acknowledge

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_CPR

Address: Operational Base + offset (0x00f4)

Component Parameter Register

UART_CPR is UART0's own unique register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	RO	0x0	reserved
13	RO	0x0	DMA_EXTRA 0 = FALSE 1 = TRUE
12	RO	0x0	UART_ADD_ENCODED_PARAMS 0 = FALSE 1 = TRUE
11	RO	0x0	SHADOW 0 = FALSE 1 = TRUE

Bit	Attr	Reset Value	Description
10	RO	0x0	FIFO_STAT 0 = FALSE 1 = TRUE
9	RO	0x0	FIFO_ACCESS 0 = FALSE 1 = TRUE
8	RO	0x0	NEW_FEAT 0 = FALSE 1 = TRUE
7	RO	0x0	SIR_LP_MODE 0 = FALSE 1 = TRUE
6	RO	0x0	SIR_MODE 0 = FALSE 1 = TRUE
5	RO	0x0	THRE_MODE 0 = FALSE 1 = TRUE
4	RO	0x0	AFCE_MODE 0 = FALSE 1 = TRUE
3:2	RO	0x0	reserved
1:0	RO	0x0	APB_DATA_WIDTH 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

UART_UCV

Address: Operational Base + offset (0x00f8)

UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x0330372a	ver ASCII value for each number in the version

UART_CTR

Address: Operational Base + offset (0x00fc)

Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

17.5 Interface Description

Table 17-1 UART Interface Description

Modulepin	Dir	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	IO_UART0rx_GMACtxd1m1_GPIO1B0vccio4	GRF_GPIO1B_IOMUX[1:0]=2'b01
uart0_sout	O	IO_UART0tx_GMACtxd0m1_GPIO1B1vccio4	GRF_GPIO1B_IOMUX[3:2]=2'b01
uart0_cts_n	I	IO_UART0ctsn_GMACrxd0m1_GPIO1B3vccio4	GRF_GPIO1B_IOMUX[7:6]=2'b01
uart0_rts_n	O	IO_UART0rtsn_GMACrxd1m1_GPIO1B2vccio4	GRF_GPIO1B_IOMUX[5:4]=2'b01
UART1 Interface			
uart1_sin	I	IO_TSPd2_CIFdata2_SDMMC0EXTd2_UART1rx_USB3PHYdebug6_GPIO3A6vccio6	GRF_GPIO3A_IOMUX[5:3]= 3'b100
uart1_sout	O	IO_TSPd0_CIFda0_SDMMC0EXTd0_UART1tx_USB3PHYdebug4_GPIO3A4vccio6	GRF_GPIO3A_IOMUX[14:12]=3'b100
uart1_cts_n	I	IO_TSPd3_CIFdata3_SDMMC0EXTd3_UART1ctsn_USB3PHYdebug7_GPIO3A7vccio6	GRF_GPIO3A_IOMUX[7:6]= 3'b100
uart1_rts_n	O	IO_TSPd1_CIFdata1_SDMMC0EXTd1_UART1rtsn_USB3PHYdebug5_GPIO3A5vccio6	GRF_GPIO3A_IOMUX[2:0]=3'b100
UART2m0 Interface			
uart2m0_sin	I	IO_SDMMC0d1_UART2DBGrxm0_GPIO1A1vccio3	GRF_GPIO1A_IOMUX[3:2]=2'b10
uart2m0_sout	O	IO_SDMMC0d0_UART2DBGtxm0_GPIO1A0vccio3	GRF_GPIO1A_IOMUX[1:0]=2'b10
UART2m1 Interface			
uart2m1_sin	I	IO_UART2DBGrxm1_POWERstate1_GPIO2A1vccio5	GRF_GPIO2A_IOMUX[3:2]=2'b01
uart2m1_sout	O	IO_UART2DBGtxm1_POWERstate0_GPIO2A0vccio5	GRF_GPIO2A_IOMUX[1:0]=2'b01

The I/O interface of UART2 can be chosen by setting GRF_CON_IOMUX[0]bit, if this bit is set to 1, UART2 uses the UART2m1 I/O interface.

17.6 Application Notes

17.6.1 None FIFO Mode Transfer Flow

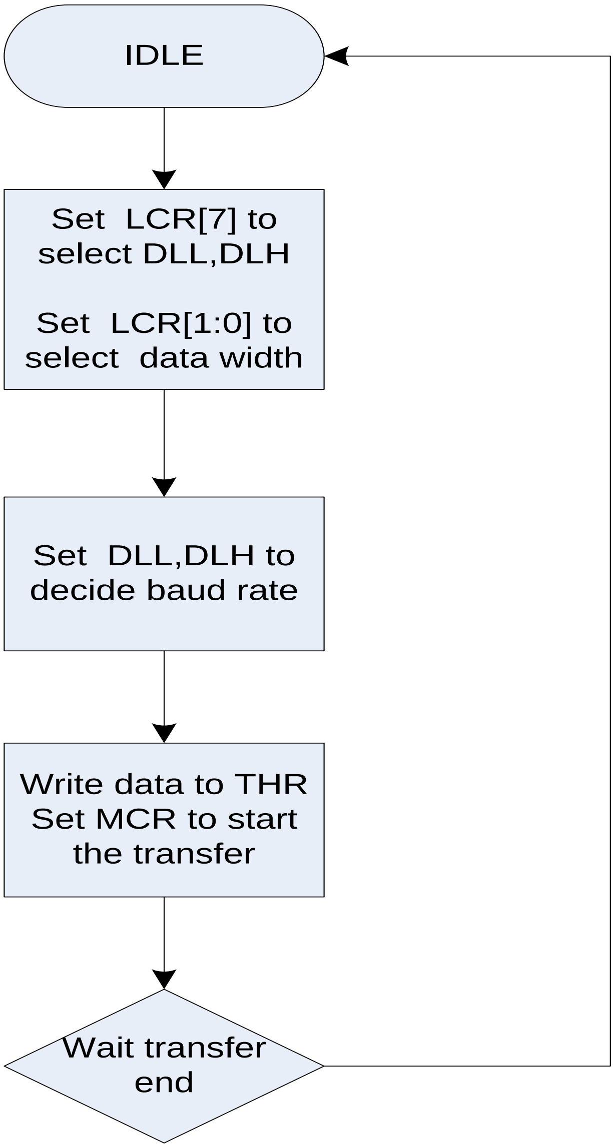


Fig. 17-8 UART none fifo mode

17.6.2 FIFO Mode Transfer Flow

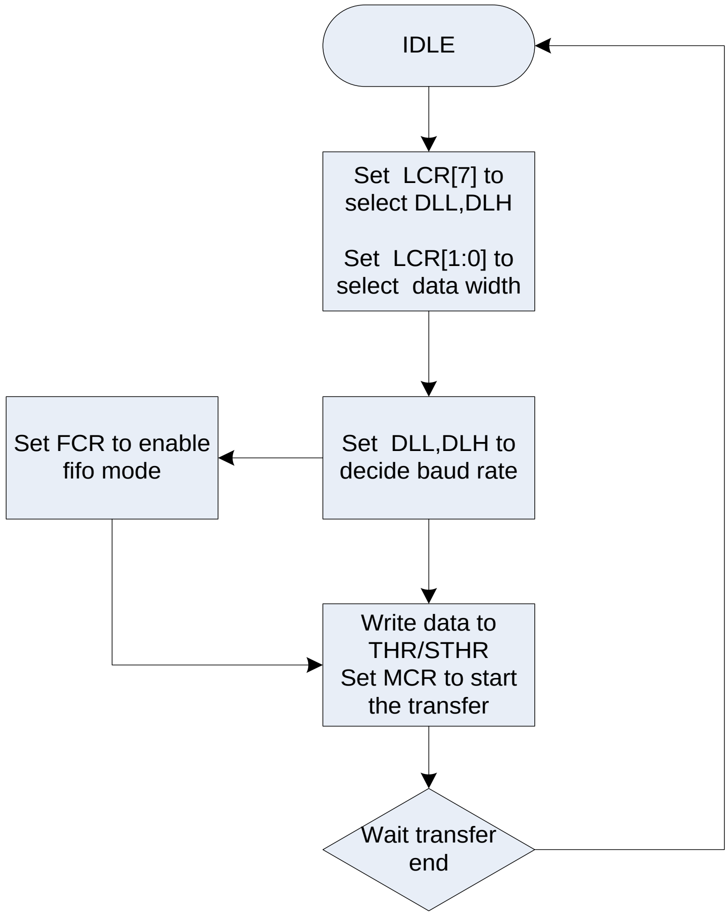


Fig. 17-9 UART fifo mode

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.

17.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.

UART0,UART1 and UART2 source clocks can be selected from three PLL outputs (CODEC PLL/GENERAL PLL/USBPHY_480M). 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.

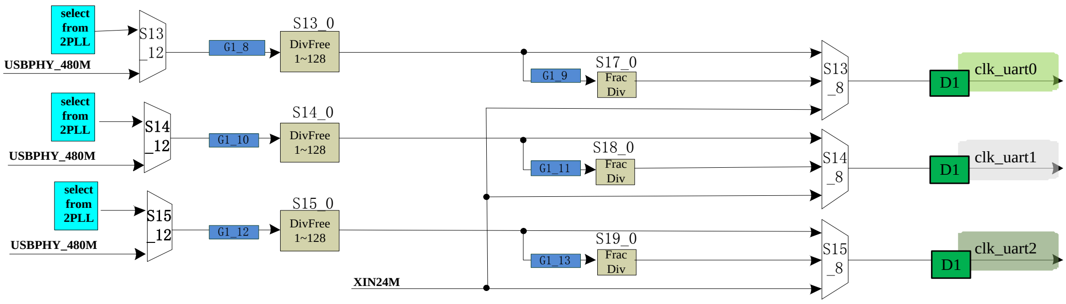


Fig. 17-10 UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 17-2 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 46875/72 to get 1.8432MHz clock; Configure UART_DLL to 1.
460.8 Kbps	Configure GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 46875/288 to get 7.3728MHz clock; Configure UART_DLL to 1.
921.6 Kbps	Configure GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 46875/576 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 50 to get 24MHz clock; Configure UART_DLL to 1.
3 Mbps	Choose GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 1200/48 to get 48MHz clock; Configure UART_DLL to 1.
4 Mbps	Configure GENERAL PLL to get 1200MHz clock output; Divide 1200MHz clock by 1200/64 to get 64MHz clock; Configure UART_DLL to 1.

17.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

- GRF_SOC_CON3[2:0] (grf_uart_cts_sel[2:0]) used to configure the polarity of cts_n.Every bit for one UART, bit2 is for UART2,bit1 is for UART1, bit0 is for UART0
- GRF_SOC_CON3[5:3] (grf_uart_rts_sel[2:0]) used to configure the polarity of rts_n.Every bit for one UART, bit2 is for UART2,bit1 is for UART1, bit0 is for UART0.
- When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, lowactive.
- When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, lowactive.

Chapter 18 GPIO

18.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is an APB slave device. GPIO controls the output data and direction of external I/O pads. It also can read back the data on external pads using memory-mapped registers.

GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode

18.2 Block Diagram

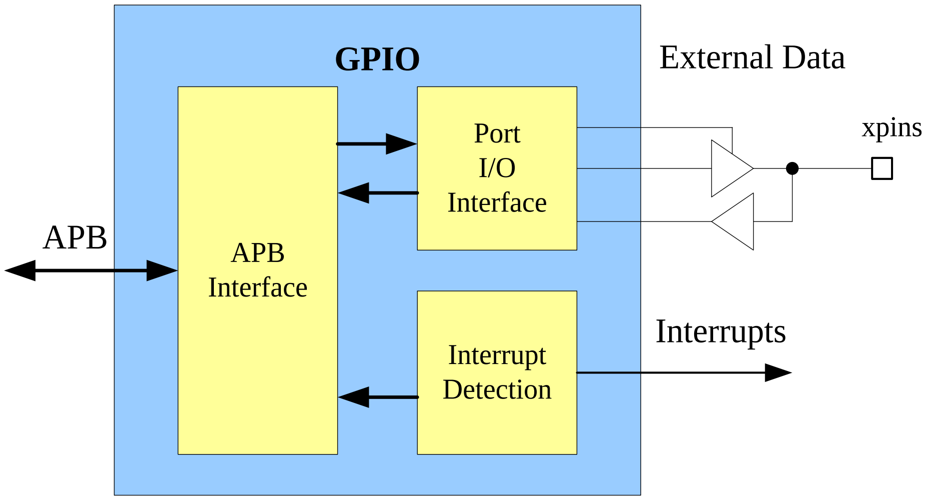


Fig. 18-1 GPIO block diagram

Block descriptions:
APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

Port I/O Interface
External data Interface to or from I/O pads.
Interrupt Detection
Interrupt interface to or from interrupt controller.

18.3 Function Description

18.3.1 Operation

Control Mode (software)

Under software control, the data and direction control for the signal are sourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR).

The direction of the external I/O pad is controlled by a write to the Porta data direction register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls the direction of an external I/O pad.

The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad. External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register(GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only, meaning that it cannot be written from the APB software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.

An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost. However, no new interrupts are generated.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

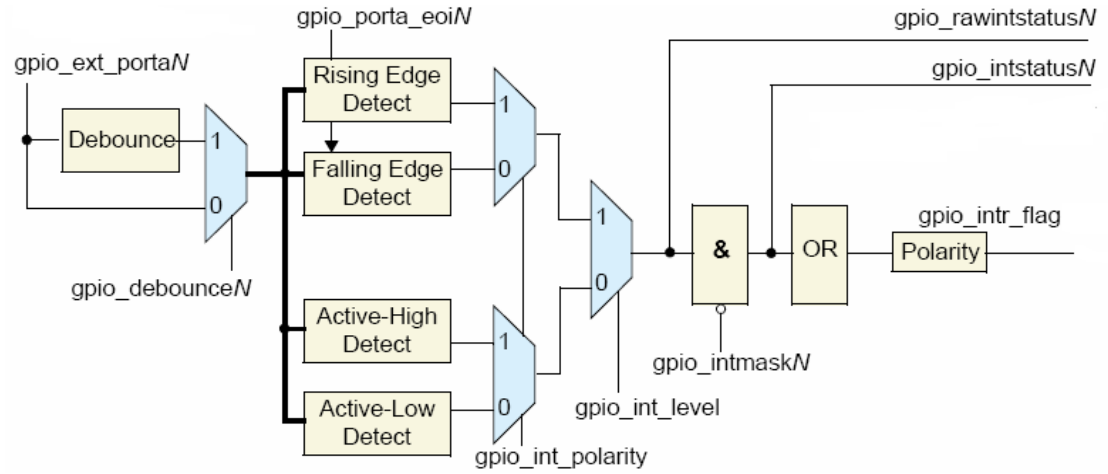


Fig. 18-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When input interrupt signals are debounced using a debounce clock (pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one rising edge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization to pclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional and under software control (GPIO_LS_SYNC).

18.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitive interrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A in order to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on a level-sensitive interrupt.

GPIOs' hierarchy in the chip

GPIO0, GPIO1, GPIO2, GPIO3 are in PD_BUS subsystem.

18.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 4 GPIOs (GPIO0 ~ GPIO3), and each of them has same register group. Therefore, 4 GPIOs' register groups have 4 different base addresses.

18.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register

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

18.4.2 Detail Register Description

GPIO_SWPORTA_DR

Address: Operational Base + offset (0x0000)

Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode.The value read back is equal to the last value written to this register.

GPIO_SWPORTA_DDR

Address: Operational Base + offset (0x0004)

Port A data direction register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 0: Input (default) 1: Output

GPIO_INTEN

Address: Operational Base + offset (0x0030)

Interrupt enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 0: Configure Port A bit as normal GPIO signal (default) 1: Configure Port A bit as interrupt

GPIO_INTMASK

Address: Operational Base + offset (0x0034)

Interrupt mask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 0: Interrupt bits are unmasked (default) 1: Mask interrupt

GPIO_INTTYPE_LEVEL

Address: Operational Base + offset (0x0038)

Interrupt level register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 0: Level-sensitive (default) 1: Edge-sensitive

GPIO_INT_POLARITY

Address: Operational Base + offset (0x003c)

Interrupt polarity register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 0: Active-low (default) 1: Active-high

GPIO_INT_STATUS

Address: Operational Base + offset (0x0040)

Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS

Address: Operational Base + offset (0x0044)

Raw Interrupt status of port A

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits

GPIO_DEBOUNCE

Address: Operational Base + offset (0x0048)

Debounce enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0: No debounce (default) 1: Enable debounce

GPIO_PORTA_EOI

Address: Operational Base + offset (0x004c)

Port A clear interrupt register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0: No interrupt clear (default) 1: Clear interrupt

GPIO_EXT_PORTA

Address: Operational Base + offset (0x0050)

Port A external port register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.

GPIO_LS_SYNC

Address: Operational Base + offset (0x0060)

Level_sensitive synchronization enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0: No synchronization to pclk_intr (default) 1: Synchronize to pclk_intr

18.5 Interface Description

Table 18-1 GPIO interface description

Module Pin	Dir	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_porta[7:0]	I/O	GPIO0_A[7:0]	GRF_GPIO0A_IOMUX[15:0]=16'h0
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0]=16'h0
gpio0_porta[23:16]	I/O	GPIO0_C[7:0]	GRF_GPIO0C_IOMUX[15:0]=16'h0
gpio0_porta[31:24]	I/O	GPIO0_D[7:0]	GRF_GPIO0D_IOMUX[15:0]=16'h0
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0]=16'h0
gpio1_porta[15:8]	I/O	GPIO1_B[7:0]	GRF_GPIO1B_IOMUX[15:0]=16'h0
gpio1_porta[23:16]	I/O	GPIO1_C[7:0]	GRF_GPIO1C_IOMUX[15:0]=16'h0
gpio1_porta[31:24]	I/O	GPIO1_D[7:0]	GRF_GPIO1D_IOMUX[15:0]=16'h0
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0]=16'h0
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2BL_IOMUX[15:0]=16'h0 GRF_GPIO2BH_IOMUX[15:0]=16'h0
gpio2_porta[23:16]	I/O	GPIO2_C[7:0]	GRF_GPIO2CL_IOMUX[15:0]=16'h0 GRF_GPIO2CH_IOMUX[15:0]=16'h0
gpio2_porta[31:24]	I/O	GPIO2_D[7:0]	GRF_GPIO2D_IOMUX[15:0]=16'h0
GPIO3 Interface			
gpio3_porta[7:0]	I/O	GPIO3_A[7:0]	GRF_GPIO3AL_IOMUX[15:0]=16'h0

Module Pin	Dir	Pad Name	IOMUX Setting
			GRF_GPIO3AH_IOMUX[15:0]=16'h0
gpio3_porta[15:8]	I/O	GPIO3_B[7:0]	GRF_GPIO3BL_IOMUX[15:0]=16'h0 GRF_GPIO3BH_IOMUX[15:0]=16'h0
gpio3_porta[23:16]	I/O	GPIO3_C[7:0]	GRF_GPIO3C_IOMUX[15:0]=16'h0
gpio3_porta[31:24]	I/O	GPIO3_D[7:0]	GRF_GPIO3D_IOMUX[15:0]=16'h0

18.6 Application Notes

Steps to set GPIO's direction

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction and Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Default GPIO's direction is input direction.

Steps to set GPIO's level

- Write GPIO_SWPORT_DDR[x] as 1 to set this gpio as output direction.
- Write GPIO_SWPORT_DR[x] as v to set this GPIO's value.

Steps to get GPIO's level

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Read from GPIO_EXT_PORT[x] to get GPIO's value

Steps to set GPIO as interrupt source

- Write GPIO_SWPORT_DDR[x] as 0 to set this gpio as input direction.
- Write GPIO_INTTYPE_LEVEL[x] as v1 and write GPIO_INT_POLARITY[x] as v2 to set interrupt type
- Write GPIO_INTEN[x] as 1 to enable GPIO's interrupt

Note: Please switch iomux to GPIO mode first!

Chapter 19 I2C Interface

19.1 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:

- 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

19.2 Block Diagram

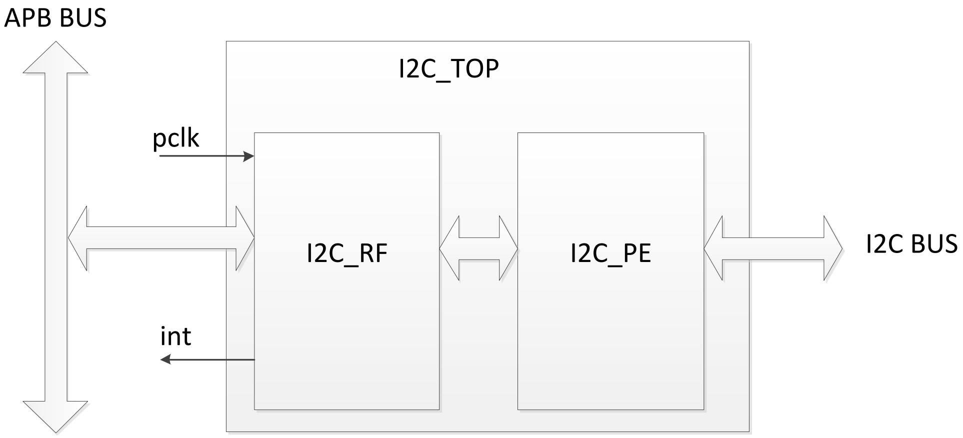


Fig. 19-1 I2C architecture

19.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

19.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

19.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

19.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller supports only Masterfunction. Itsupports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

19.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the working clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

19.3.2 Master Mode Programming

- SCL Clock

  When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:

  SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)

  SCL = PCLK/ SCLK Divisor

- Data Receiver Register Access

  When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.

  When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register

  Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.

  When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command

  Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command

  Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode

  There are four i2c operation modes.

  - When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
  - When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
  - When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
  - When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command
  - When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.
  - In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].
  - In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition

  There are 7 interrupt bits in I2C_ISR register related to master mode.

  - Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed transmitting a byte.
  - Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
  - MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
  - MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
  - Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
  - Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
  - NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

- Last byte acknowledge control
  - If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
  - If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

- How to handle NAK handshake received
  - If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
  - If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

- I2C controller data transfer waveform
  - Bit transferring
    - Data Validity

      The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

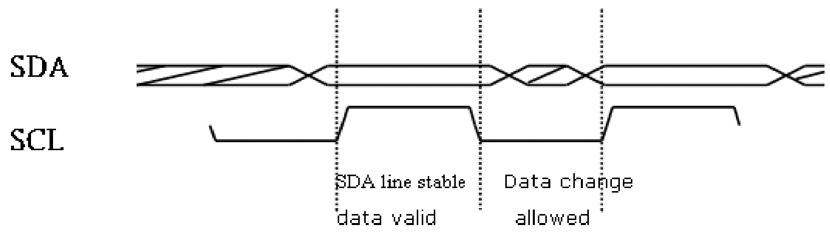


Fig. 19-2 I2C DATA Validity

- 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.

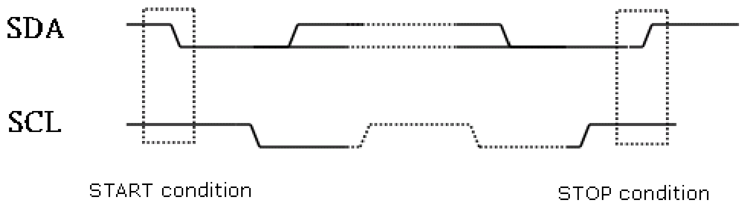


Fig. 19-3 I2C Start and stop conditions

- 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".

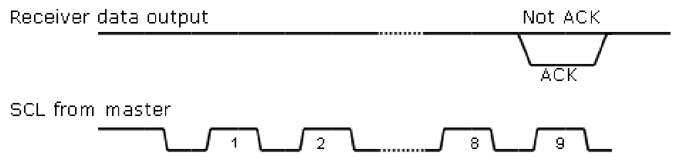


Fig. 19-4 I2C Acknowledge

- 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.

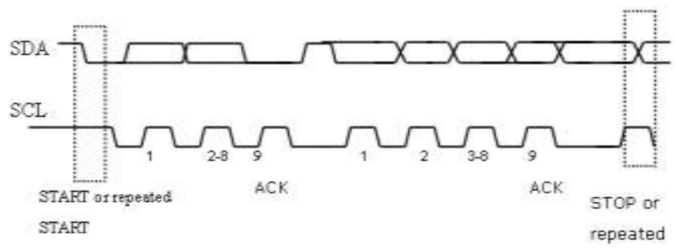


Fig. 19-5 I2C byte transfer

19.4 Register Description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00000000	control register
RKI2C_CLKDIV	0x0004	W	0x00000001	clock divider register
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
RKI2C_MRXCNT	0x0014	W	0x00000000	master rx count
RKI2C_IEN	0x0018	W	0x00000000	interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	interrupt pending register
RKI2C_FCNT	0x0020	W	0x00000000	finished count
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

Name	Offset	Size	Reset Value	Description
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

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

19.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)

control register

Bit	Attr	Reset Value	Description
31:7	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 stop enable, when this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start start enable start enable, when this bit is written to 1, I2C will generate start signal.

Bit	Attr	Reset Value	Description
2:1	RW	0x0	i2c_mode i2c mode select 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable 1'b0:not enable 1'b1:enable

RKI2C_CLKDIV

Address: Operational Base + offset (0x0004)

clock divider register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count T(SCL_HIGH) = T(PCLK) * (CLKDIVH + 1) * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count T(SCL_LOW) = T(PCLK) * (CLKDIVL + 1) * 8

RKI2C_MRXADDR

Address: Operational Base + offset (0x0008)

the slave address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid 1'b0:invalid 1'b1:valid

Bit	Attr	Reset Value	Description
25	RW	0x0	addmvld address middle byte valid 1'b0:invalid 1'b1:valid
24	RW	0x0	addlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read 24 bits address register

RKI2C_MRXRADDR

Address: Operational Base + offset (0x000c)

the slave register address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid 1'b0:invalid 1'b1:valid
25	RW	0x0	sraddmvld address middle byte valid 1'b0:invalid 1'b1:valid
24	RW	0x0	sraddlvld address low byte valid 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	sraddr slave register address accessed 24 bits register address

RKI2C_MTXCNT

Address: Operational Base + offset (0x0010)

master transmit count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt master transmit count 6 bits counter

RKI2C_MRXCNT

Address: Operational Base + offset (0x0014)

masterrx count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master rx count 6 bits counter

RKI2C_IEN

Address: Operational Base + offset (0x0018)

interrupt enable register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
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)

interrupt pending register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	W1C	0x0	nakrcvipd NAK handshake received interrupt pending bit 1'b0:no interrupt available 1'b1:NAK handshake received interrupt appear, write 1 to clear
5	W1C	0x0	stopipd stop operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:stop operation finished interrupt appear, write 1 to clear
4	W1C	0x0	startipd start operation finished interrupt pending bit 1'b0:no interrupt available 1'b1:start operation finished interrupt appear, write 1 to clear
3	W1C	0x0	mbrfipd MRXCNT data received finished interrupt pending bit 1'b0:no interrupt available 1'b1:MRXCNT data received finished interrupt appear, write 1 to clear
2	W1C	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit 1'b0:no interrupt available 1'b1:MTXCNT data transfer finished interrupt appear, write 1 to clear

Bit	Attr	Reset Value	Description
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)

finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

RKI2C_TXDATA0

Address: Operational Base + offset (0x0100)

I2C tx data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 data0 to be transmitted 32 bits data

RKI2C_TXDATA1

Address: Operational Base + offset (0x0104)

I2C tx data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 data1 to be transmitted 32 bits data

RKI2C_TXDATA2

Address: Operational Base + offset (0x0108)

I2C tx data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 data2 to be transmitted 32 bits data

RKI2C_TXDATA3

Address: Operational Base + offset (0x010c)

I2C tx data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 data3 to be transmitted 32 bits data

RKI2C_TXDATA4

Address: Operational Base + offset (0x0110)

I2C tx data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 data4 to be transmitted 32 bits data

RKI2C_TXDATA5

Address: Operational Base + offset (0x0114)

I2C tx data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 data5 to be transmitted 32 bits data

RKI2C_TXDATA6

Address: Operational Base + offset (0x0118)

I2C tx data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 data6 to be transmitted 32 bits data

RKI2C_TXDATA7

Address: Operational Base + offset (0x011c)

I2C tx data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 data7 to be transmitted 32 bits data

RKI2C_RXDATA0

Address: Operational Base + offset (0x0200)

I2C rx data register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0 data0 received 32 bits data

RKI2C_RXDATA1

Address: Operational Base + offset (0x0204)

I2C rx data register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1 data1 received 32 bits data

RKI2C_RXDATA2

Address: Operational Base + offset (0x0208)

I2C rx data register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2 data2 received 32 bits data

RKI2C_RXDATA3

Address: Operational Base + offset (0x020c)

I2C rx data register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3 data3 received 32 bits data

RKI2C_RXDATA4

Address: Operational Base + offset (0x0210)

I2C rx data register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4 data4 received 32 bits data

RKI2C_RXDATA5

Address: Operational Base + offset (0x0214)

I2C rx data register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5 data5 received 32 bits data

RKI2C_RXDATA6

Address: Operational Base + offset (0x0218)

I2C rx data register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6 data6 received 32 bits data

RKI2C_RXDATA7

Address: Operational Base + offset (0x021c)

I2C rx data register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7 data7 received 32 bits data

19.5 Interface Description

Table 19-1 I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	IO_I2C0scl_FEPHYled_linkm1_GPIO2D0vccio5	GRF_GPIO2D_IOMUX[3:2]=2'b01
i2c0_scl	I/O	IO_I2C0sda_FEPHYLEDrxm1_FEPHYLEDtxm1_GPIO2D1vccio5	GRF_GPIO2D_IOMUX[1:0]=2'b01
I2C1 Interface			
i2c1_sda	I/O	IO_PWM0_I2C1sda_GPIO2A4vccio5	GRF_GPIO2A_IOMUX[9:8]=2'b10
i2c1_scl	I/O	IO_PWM1_I2C1scl_GPIO2A5vccio5	GRF_GPIO2A_IOMUX[5:4]=2'b10
I2C2 Interface			
i2c2_sda	I/O	IO_ I2C2sda_TSADCshut_GPIO2B5vccio5	GRF_GPIO2B_IOMUX[11:10]=2'b01
i2c2_scl	I/O	IO_ I2C2scl_GPIO2B6vccio5	GRF_GPIO2B_IOMUX[13:12]=2'b01
I2C3 Interface			
i2c3_sda	I/O	IO_HDMIscl_I2C3scl_GPIO0A5pmuio	GRF_GPIO0A_IOMUX[13:12]=2'b10
		IO_I2C3scl5v_HDMISCLpmuio5v	GRF_CON_I2C3_SCL5V=1
i2c3_scl	I/O	IO_HDMIsda_I2C3sda_GPIO0A6pmuio	GRF_GPIO0A_IOMUX[11:10]=2'b10
		IO_I2Csda5v_HDMISDApmuio5v	GRF_CON_I2C3_SDA5V=1

19.6 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)

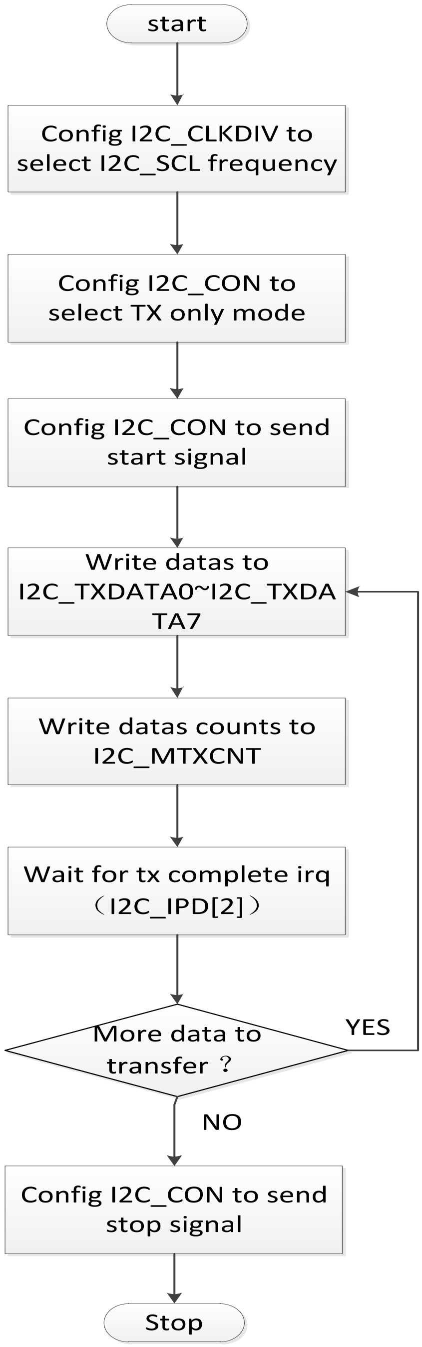


Fig. 19-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

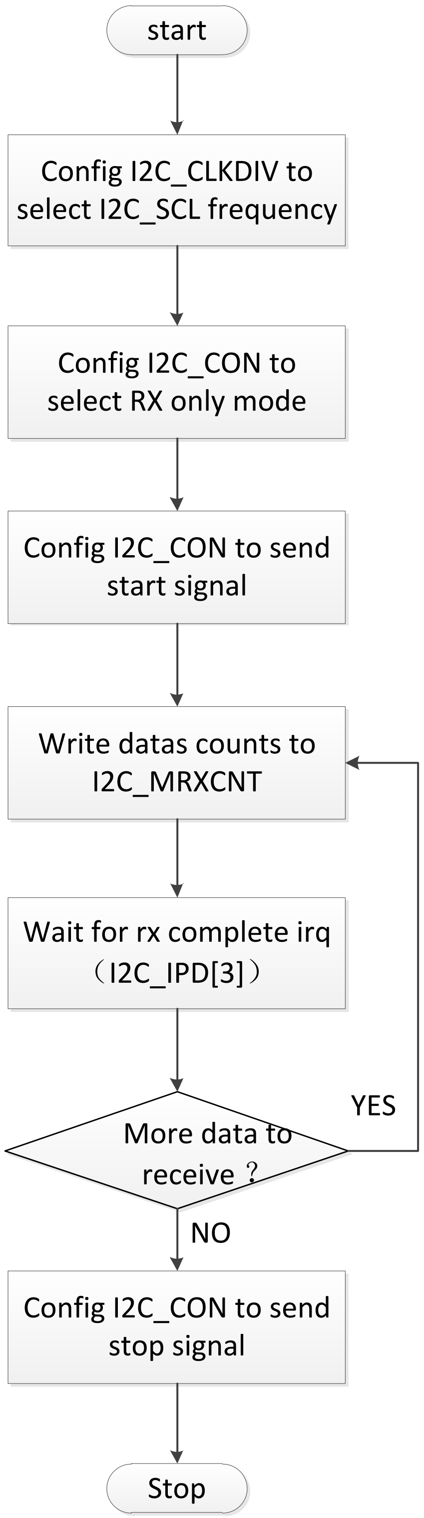


Fig. 19-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

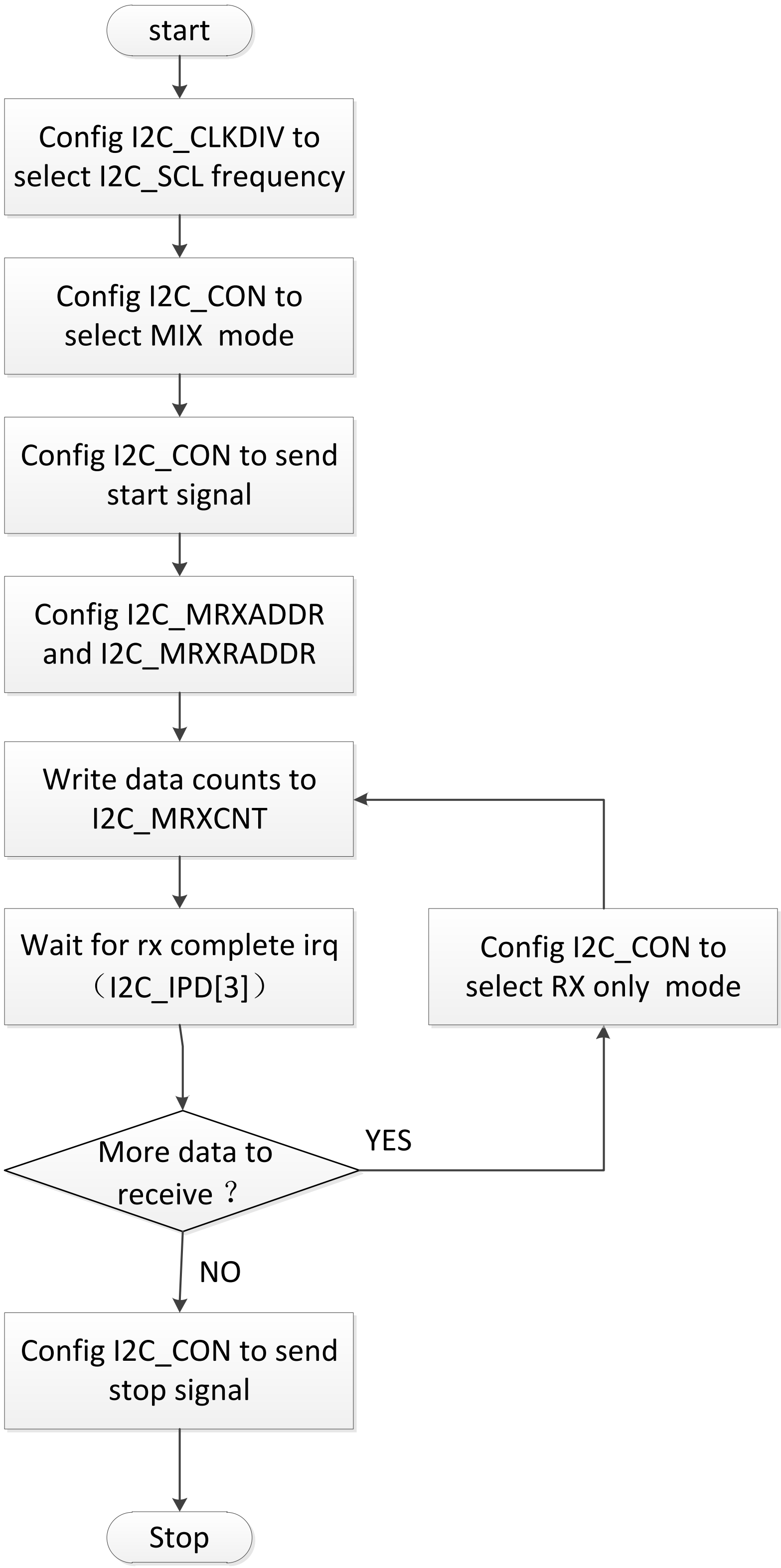


Fig. 19-8 I2C Flow chat for mix mode

Chapter 20 Serial Peripheral Interface (SPI)

20.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

20.2 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

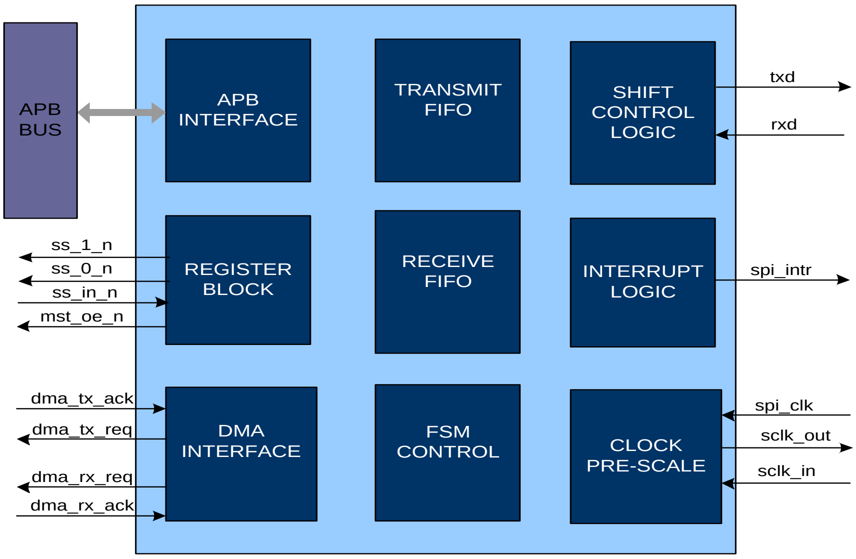


Fig. 20-1 SPI Controller Block diagram

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 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

20.3 Function Description

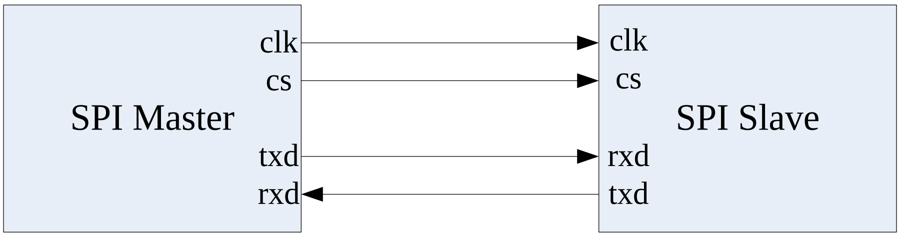


Fig. 20-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1). Transmit and Receive

When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.

2).Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3).Receive Only

When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})

When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

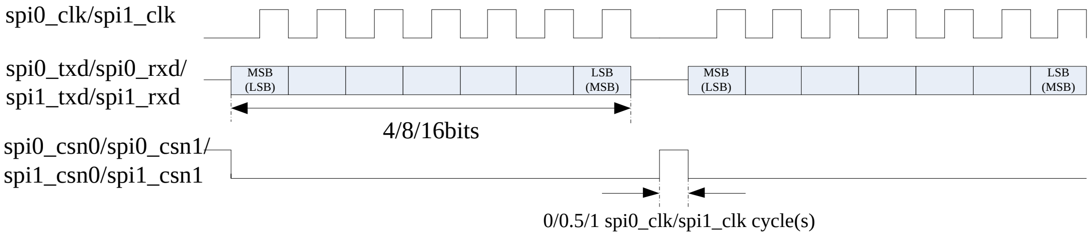


Fig. 20-3 SPI Format (SCPH=0 SCPOL=0)

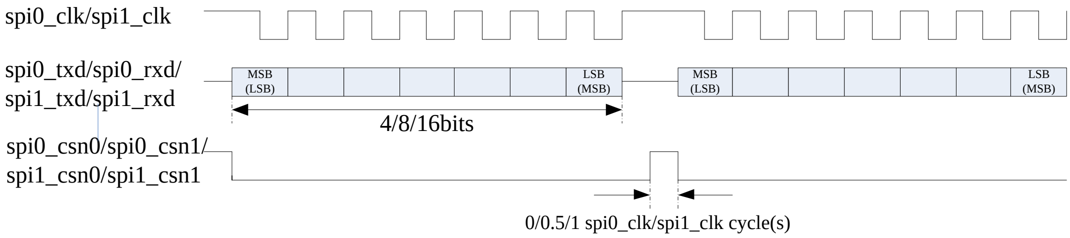


Fig. 20-4 SPI Format (SCPH=0 SCPOL=1)

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.

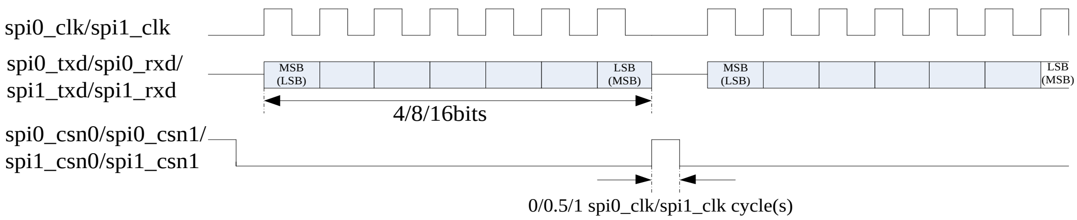


Fig. 20-5 SPI Format (SCPH=1 SCPOL=0)

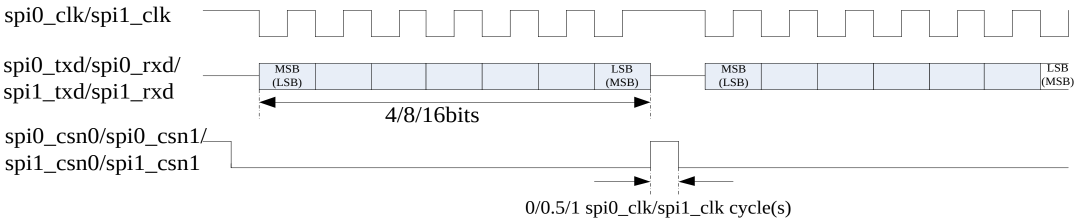


Fig. 20-6 SPI Format (SCPH=1 SCPOL=1)

20.4 Register Description

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0

Name	Offset	Size	Reset Value	Description
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable
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	0x0000000c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0048	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x004c	W	0x00000000	Receive FIFO Data

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

20.4.2 Detail Register Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)

Control Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Microwire Transfer Mode Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM Operation Mode 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	XFM Transfer Mode 2'b00 :Transmit & Receive 2'b01 : Transmit Only 2'b10 : Receive Only 2'b11 :reserved

Bit	Attr	Reset Value	Description
17:16	RW	0x0	FRF Frame Format 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd Sample Delay When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform Valid when data frame size is 8bit. 1'b0:apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First Bit Mode 1'b0:first bit is MSB 1'b1:first bit is LSB
11	RW	0x0	EM Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0:little endian 1'b1:big endian
10	RW	0x0	SSD ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle.

Bit	Attr	Reset Value	Description
9:8	RW	0x0	CSM Chip Select Mode Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01:ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11:reserved
7	RW	0x0	SCPOL Serial Clock Polarity Valid when the frame format is set to Motorola SPI. 1'b0: Inactive state of serial clock is low 1'b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase Valid when the frame format is set to Motorola SPI. 1'b0: Serial clock toggles in middle of first data bit 1'b1: Serial clock toggles at start of first data bit
5:2	RW	0x0	CFS Control Frame Size Selects the length of the control word for the Microwire frame format. 4'b0000~0010:reserved 4'b0011:4-bit serial data transfer 4'b0100:5-bit serial data transfer 4'b0101:6-bit serial data transfer 4'b0110:7-bit serial data transfer 4'b0111:8-bit serial data transfer 4'b1000:9-bit serial data transfer 4'b1001:10-bit serial data transfer 4'b1010:11-bit serial data transfer 4'b1011:12-bit serial data transfer 4'b1100:13-bit serial data transfer 4'b1101:14-bit serial data transfer 4'b1110:15-bit serial data transfer 4'b1111:16-bit serial data transfer
1:0	RW	0x2	DFS Data Frame Size Selects the data frame length. 2'b00:4bit data 2'b01:8bit data 2'b10:16bit data 2'b11:reserved

SPI_CTRLR1
Address: Operational Base + offset (0x0004)
Control Register 1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM Number of Data Frames When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR
Address: Operational Base + offset (0x0008)
SPI Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ENR SPI Enable 1’b1: Enable all SPI operations. 1’b0: Disable all SPI operations Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER
Address: Operational Base + offset (0x000c)
Slave Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	SER1 Slave 1 Select Enable 1’b1: Enable chip select 1 1’b0: Disable chip select 1 This register is valid only when SPI is configured as a master device.
0	RW	0x0	SER0 Slave Select Enable 1’b1: Enable chip select 0 1’b0: Disable chip select 0 This register is valid only when SPI is configured as a master device.

SPI_BAUDR

Address: Operational Base + offset (0x0010)

Baud Rate Select

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	BAUDR Baud Rate Select 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)

Transmit FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TXFTLR Transmit FIFO Threshold Level When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR

Address: Operational Base + offset (0x0018)

Receive FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RXFTLR Receive FIFO Threshold Level When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR

Address: Operational Base + offset (0x001c)

Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TXFLR Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR

Address: Operational Base + offset (0x0020)

Receive FIFO Level

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

SPI_SR

Address: Operational Base + offset (0x0024)

SPI Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF Receive FIFO Full 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RO	0x1	RFE Receive FIFO Empty 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE Transmit FIFO Empty 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF Transmit FIFO Full 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full

Bit	Attr	Reset Value	Description
0	RO	0x0	BSF SPI Busy Flag When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR

Address: Operational Base + offset (0x0028)

Interrupt Polarity

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	IPR Interrupt Polarity 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)

Interrupt Mask

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM Receive FIFO Full Interrupt Mask 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM Receive FIFO Overflow Interrupt Mask 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM Receive FIFO Underflow Interrupt Mask 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM Transmit FIFO Overflow Interrupt Mask 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM Transmit FIFO Empty Interrupt Mask 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR

Address: Operational Base + offset (0x0030)

Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS Receive FIFO Full Interrupt Status 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS Receive FIFO Overflow Interrupt Status 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking
2	RO	0x0	RFUIS Receive FIFO Underflow Interrupt Status 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS Transmit FIFO Overflow Interrupt Status 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS Transmit FIFO Empty Interrupt Status 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR

Address: Operational Base + offset (0x0034)

Raw Interrupt Status

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFRIS Receive FIFO Full Raw Interrupt Status 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS Receive FIFO Overflow Raw Interrupt Status 1'b0 = spi_rxo_intr interrupt is not active prior to masking 1'b1 = spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS Receive FIFO Underflow Raw Interrupt Status 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking

Bit	Attr	Reset Value	Description
1	RO	0x0	TFORIS Transmit FIFO Overflow Raw Interrupt Status 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking
0	RO	0x1	TFERIS Transmit FIFO Empty Raw Interrupt Status 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR

Address: Operational Base + offset (0x0038)

Interrupt Clear

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	WO	0x0	CTFOI Clear Transmit FIFO Overflow Interrupt Write 1 to Clear Transmit FIFO Overflow Interrupt
2	WO	0x0	CRFOI Clear Receive FIFO Overflow Interrupt Write 1 to Clear Receive FIFO Overflow Interrupt
1	WO	0x0	CRFUI Clear Receive FIFO Underflow Interrupt Write 1 to Clear Receive FIFO Underflow Interrupt
0	WO	0x0	CCI Clear Combined Interrupt Write 1 to Clear Combined Interrupt

SPI_DMACR

Address: Operational Base + offset (0x003c)

DMA Control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR

Address: Operational Base + offset (0x0040)
DMA Transmit Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR
Address: Operational Base + offset (0x0044)
DMA Receive Data Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR
Address: Operational Base + offset (0x0048)
Transmit FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR Transimt FIFO Data Register. When it is written to, data are moved into the transmit FIFO.

SPI_RXDR
Address: Operational Base + offset (0x004c)
Receive FIFO Data

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR Receive FIFO Data Register. When the register is read, data in the receive FIFO is accessed.

20.5 Interface Description

Table 20-1 1SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_clk	I/O	IO_SPIclkm0_GPIO2B0vccio5	GRF_GPIO2B_IOMUX[1:0]=2’b01
spi0_rxd	I	IO_SPIrxdm0_GPIO2B2vccio5	GRF_GPIO2B_IOMUX[5:4]=2’b01
spi0_txd	O	IO_SPItxdm0_GPIO2B1vccio5	GRF_GPIO2B_IOMUX[3:2]=2’b01
spi0_csn0	I/O	IO_SPIcsn0m0_GPIO2B3vccio5	GRF_GPIO2B_IOMUX[7:6]=2’b01
spi0_csn1	O	IO_SPIcsn1m0_FLASHvol_sel_GPIO2B4vccio5	GRF_GPIO2B_IOMUX[9:8]=2’b01
spi1_clk	I/O	IO_FLASHcs1_SPIclkm1_GPIO3C7vccio2	GRF_GPIO3C_IOMUX[15:14]=2’b10
spi1_rxd	I	IO_FLASHale_SPIrxdm1_GPIO3D0vccio2	GRF_GPIO3D_IOMUX[1:0]=2’b10
spi1_txd	O	IO_FLASHcle_SPItxdm1_GPIO3D1vccio2	GRF_GPIO3D_IOMUX[3:2]=2’b10
spi1_csn0	I/O	IO_FLASHwrn_SPIcsn0m1_GPIO3D2vccio2	GRF_GPIO3D_IOMUX[5:4]=2’b10
spi1_csn1	O	IO_FLASHcs0_SPIcsn1m1_GPIO3Dvccio2	GRF_GPIO3D_IOMUX[7:6]=2’b10
spi2_clk	I/O	IO_TSPvalid_CIFvsync_SDMMC0EXTcmd_SPIclkm2_USB3PHYdebug1_I2S2sclkm1_GPIO3A0vccio6	GRF_GPIO3AL_IOMUX[2:0]=3’b100
spi2_rxd	I	IO_TSLclk_CIFclkin_SDMMC0EXTclkout_SPIrxdm2_USB3PHYdebug3_I2S2sdim1_GPIO3A2vccio6	GRF_GPIO3AL_IOMUX[5:3]=3’b100
spi2_txd	O	IO_TSPfail_CIFhref_SDMMC0EXTdet_SPItxdm2_USB3PHYdebug2_I2S2sdom1_GPIO3A1vccio6	GRF_GPIO3AL_IOMUX[8:6]=3’b100
spi2_csn0	I/O	IO_TSPd4_CIFdata4_SPIcsn0m2_I2S2lrcktxm1_USB3PHYdebug8_I2S2lrckrxm1_GPIO3B0vccio6	GRF_GPIO3BL_IOMUX[2:0]=3’b011

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

20.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)

When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

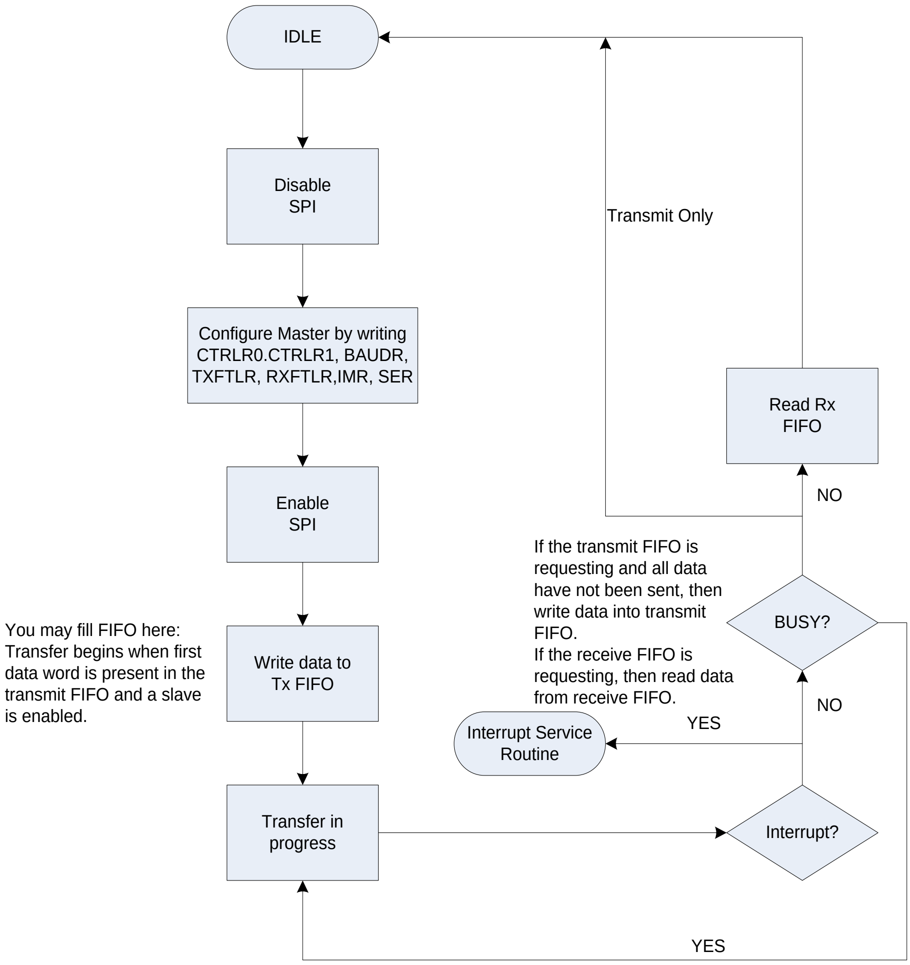


Fig. 20-7 SPI Master transfer flow diagram

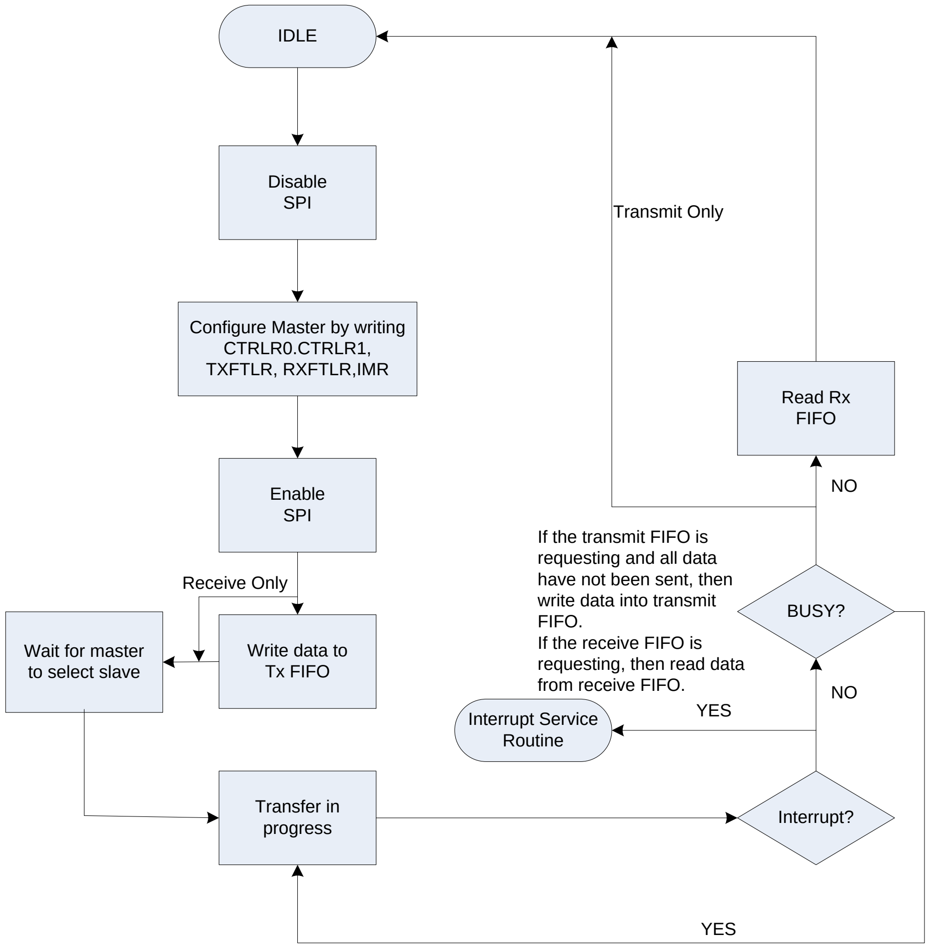


Fig. 20-8 SPI Slave transfer flow diagram

Chapter 21 SPDIF Transmitter

21.1 Overview

The SPDIF transmitter is a self-clocking, serial and unidirectional interface for the interconnection of digital audio equipment in consumer and professional applications which uses linear PCM coded audio samples.
When used in professional application, the interface is primarily intended to carry monophonic or stereophonic programmes at a 48 kHz sampling frequency with a resolution of up to 24bits per sample. It may alternatively be used to carry signals sampled at 32 kHz or 44.1 kHz.
When used in consumer application, the interface is primarily intended to carry stereophonic programmes with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.
When used for other purposes, the interface is primarily intended to carry audio data coded other than linear PCM coded audio samples. Provision is also made to allow the interface to carry data related to computer software or signals coded using non-linear PCM. The maximum sample frequency can be up to 192 kHz for the non-linear PCM mode.
In all cases, the clock references and auxiliary information are transmitted along with the programme.

- Supports one internal 32-bit wide and 32-location deep sample data buffer
- Supports two 16-bit audio data store together in one 32-bit wide location
- Supports AHB bus interface
- Supports biphase format stereo audio data output
- Supports DMA handshake interface and configurable DMA water level
- Supports sample data buffer empty, block terminate and user data interrupt
- Supports combine interrupt output
- Supports 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 16, 20, 24 bits audio data transfer in linear PCM mode
- Support non-linear PCM transfer

21.2 Block Diagram

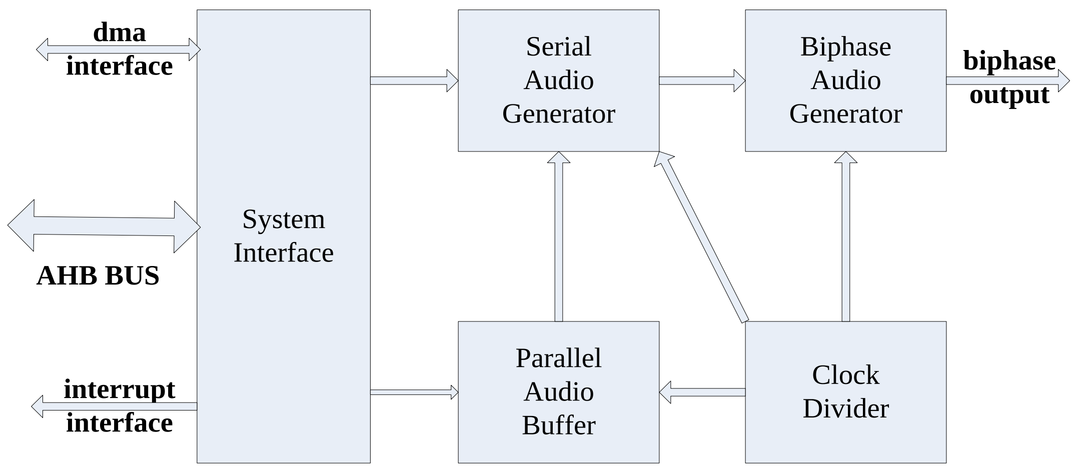


Fig.21-1 SPDIF transmitter Block Diagram

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 handshake interface.

Clock Divider
The Clock Divider implements clock generation function. The input source clock to the module is MCLK. By the divider of the module, the clock divider generates work clock for digital audio data transformation.
Parallel Audio Buffer
The Parallel Audio Buffer is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.
Serial Audio Converter
The Serial Audio Converter reads parallel audio data from the Parallel Audio Buffer and converts it to serial audio data.
Biphase Audio Generator
The Biphase Audio Generator reads serial audio data from the Serial Audio Converter and generates biphase audio data based on IEC-60958 standard.

21.3 Function description

21.3.1 Frame Format

A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of transmission of frames corresponds exactly to the source sampling frequency.
In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. The first sub-frame(left channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame to identify the start of the block structure used to organize the channel status information. The second sub-frame (right in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

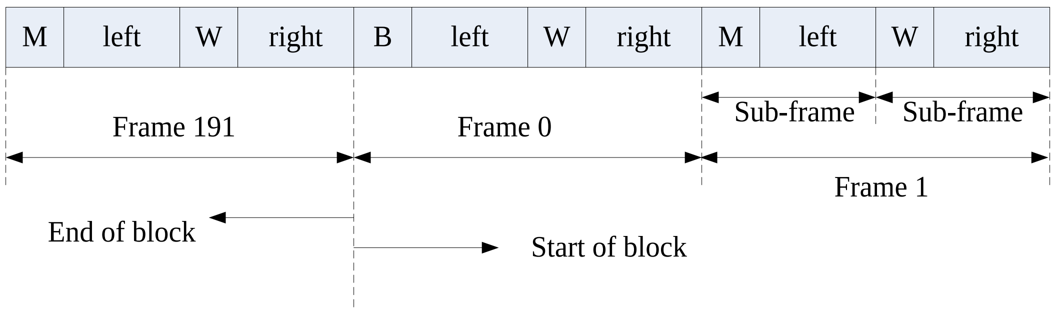


Fig.21-2 SPDIF Frame Format

In the single channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried only in the first sub-frame and may be duplicated in the second sub-frame. If the second sub-frame is not carrying duplicate data, then time slot 28 (validity flag) shall be set to logical '1' (not valid).

21.3.2 Sub-frame Format

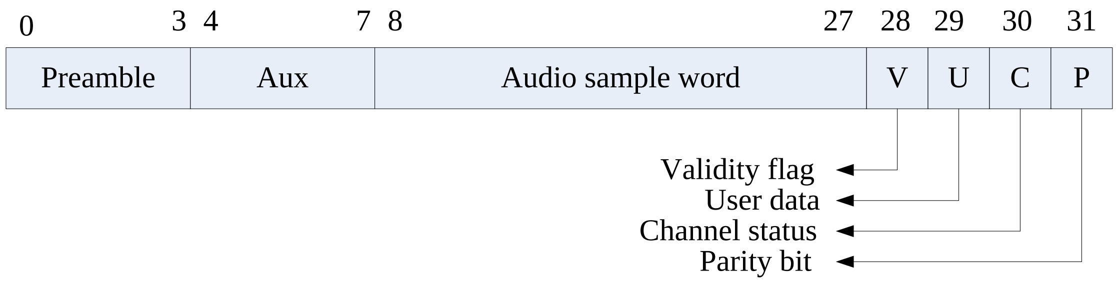


Fig.21-3 SPDIF Sub-frame Format

Each sub-frame is divided into 32 time slots, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. Time slot 4 to 27 carry the audio sample word in linear 2's complement representation. The MSB is carried by time slot 27. When a 24-bit coding range is used, the LSB is in time slot 4. When a 20-bit coding range is used, time slot 8 to 27 carry the audio sample word with the LSB in time slot 8.Time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 24 or 20), the unused LSBs are set to a logical '0'. For a non-linear PCM audio application or a data application the main data field may carry any other information. Time slot 28 carries the validity flag associated with the main data field. Time slot 29 carries 1 bit of the user data associated with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit of the channel status words associated with the main data field channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

21.3.3 Channel Coding

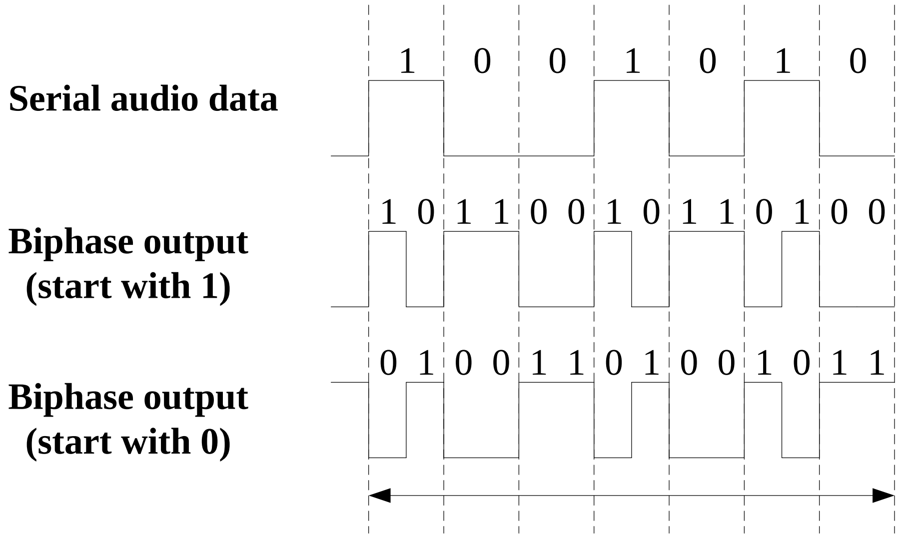


Fig.21-4 SPDIF Channel Coding

To minimize the direct current component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphase-mark.

Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical '0'.However, it is different from the first if the bit is logical '1'.

21.3.4 Preamble

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks.
To achieve synchronization within one sampling period and to make this process completely reliable, these patterns violate the biphase-mark code rules, thereby avoiding the possibility of data imitating the preambles.
A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

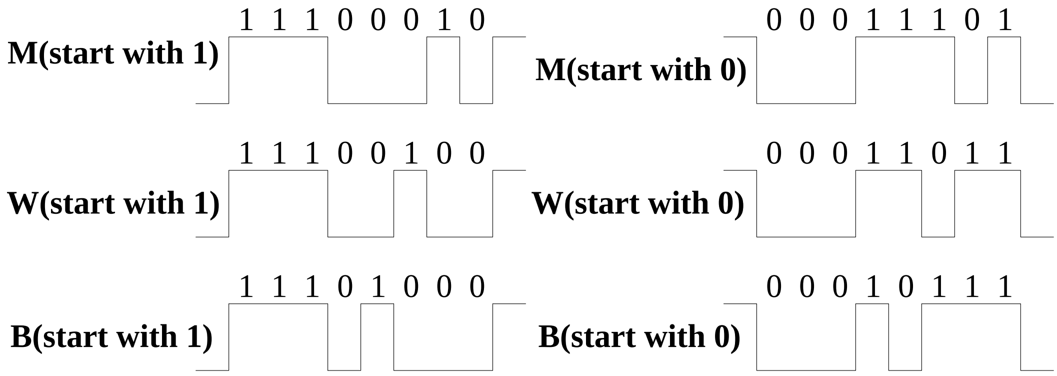


Fig.21-5 SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

21.3.5 NON-LINEAR PCM ENCODED SOURCE(IEC 61937)

The non-linear PCM encoded audio bitstream is transferred using the basic 16-bit data area of the IEC 60958subframes, i.e. in time slots 12 to 27. Each IEC 60958 frame transfers 32-bit of the non-PCM data in consumer application mode.
If the SPDIF bitstream conveys linear PCM audio, the symbol frequency is 64 times the PCM sampling frequency(32 time slots per PCM sample times two channels). If a non-linear PCM encoded audio bitstream is conveyed by the interface, the symbol frequency is 64 times the sampling rate of the encoded audio within that bitstream. But in the case where a non-linear PCM encoded audio bitstream is conveyed by the interface containing audio with low sampling frequency, the symbol frequency is 128 times the sampling rate of the encoded audio within that bitstream.
Each data burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc, Pd), followed by the burst payload which contains data of an encoded audio frame.
The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word. Pc gives information about the type of data and some information/control for the receiver. Pd gives the length of the burst payload, the number of bits or number of bytes according to data-type.
The four preamble words are contained in two sequential SPDIF frames. The frame beginning the data-burst contains preamble word Pa in subframe 0 and Pb in subframe 1. The next frame contains Pc in subframe 0 and Pd in subframe 1. When placed into a SPDIF subframe, the MSB of a 16-bit burst-preamble is placed into timeslot 27 and the LSB is placed into time slot 12.

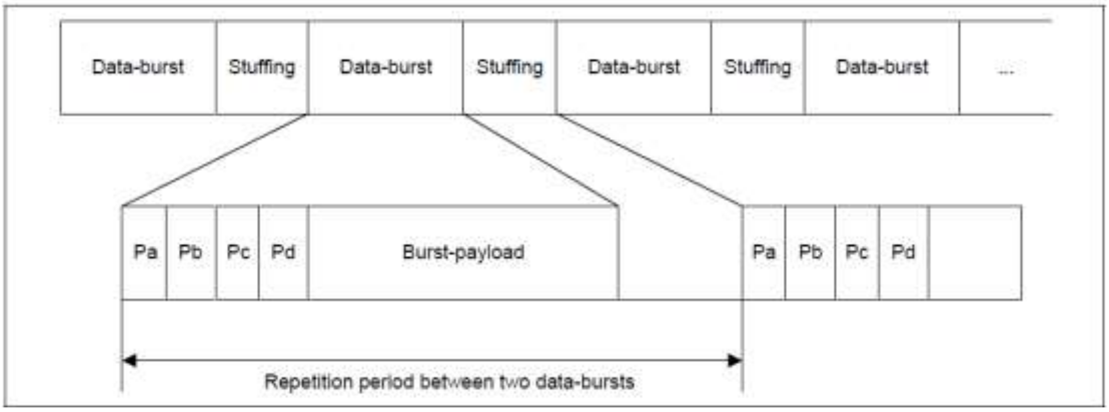


Fig.21-6 Format of Data-burst

21.4 Register description

21.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SPDIF_CFGR	0x0000	W	0x00000000	Transfer Configuration Register
SPDIF_SDBLR	0x0004	W	0x00000000	Sample Date Buffer Level Register
SPDIF_DMACR	0x0008	W	0x00000000	DMA Control Register
SPDIF_INTCR	0x000c	W	0x00000000	Interrupt Control Register
SPDIF_INTSR	0x0010	W	0x00000000	Interrupt Status Register
SPDIF_XFER	0x0018	W	0x00000000	Transfer Start Register
SPDIF_SMPDR	0x0020	W	0x00000000	Sample Data Register
SPDIF_VLDFRn	0x0060	W	0x00000000	Validity Flag Register n
SPDIF_USRDRn	0x0090	W	0x00000000	User Data Register n
SPDIF_CHNSRn	0x00c0	W	0x00000000	Channel Status Register n
SPDIF_BURTSINFO	0x0100	W	0x00000000	Channel Burst Info Register
SPDIF_REPETTION	0x0104	W	0x00000000	Channel Repetition Register
SPDIF_BURTSINFO_SHD	0x0108	W	0x00000000	Shadow Channel Burst Info Register
SPDIF_REPETTION_SHD	0x010c	W	0x00000000	Shadow Channel Repetition Register
SPDIF_USRDR_SHDn	0x0190	W	0x00000000	Shadow User Data Register n

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

21.4.2 Detail Register Description

SPDIF_CFGR

Address: Operational Base + offset (0x0000)

Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MCD mclk divider Fmclk/Fsdo This parameter can be calculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	PCMTYPE PCM type 0: linear PCM 1: non-linear PCM
7	WO	0x0	CLR mclk domain logic clear Write 1 to clear mclk domain logic. Read return zero.
6	RW	0x0	CSE Channel status enable 0: disable 1: enable The bit should be set to 1 when the channel conveys non-linear PCM
5	RW	0x0	UDE User data enable 0: disable 1: enable
4	RW	0x0	VFE Validity flag enable 0: disable 1: enable
3	RW	0x0	ADJ audio data justified 0: Right justified 1: Left justified
2	RW	0x0	HWT Halfword word transform enable 0: disable 1: enable It is valid only when the valid data width is 16bit.
1:0	RW	0x0	VDW Valid data width 00: 16bit 01: 20bit 10: 24bit 11: reserved The valid data width is 16bit only for non-linear PCM

SPDIF_SDBLR
Address: Operational Base + offset (0x0004)
Sample Date Buffer Level Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	SDBLR Sample Date Buffer Level Register Contains the number of valid data entries in the sample data buffer.

SPDIF_DMACR

Address: Operational Base + offset (0x0008)

DMA Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	TDE Transmit DMA Enable 0: Transmit DMA disabled 1: Transmit DMA enabled
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 Sample Date Buffer is equal to or below this field value

SPDIF_INTCR

Address: Operational Base + offset (0x000c)

Interrupt Control Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	W1C	0x0	UDTIC User Data Interrupt Clear Write '1' to clear the user data interrupt.
16	W1C	0x0	BTTIC Block/Data burst transfer finish interrupt clear Write 1 to clear the interrupt.
15:10	RO	0x0	reserved
9:5	RW	0x00	SDBT Sample Date Buffer Threshold Sample Date Buffer Threshold for empty interrupt
4	RW	0x0	SDBEIE Sample Date Buffer empty interrupt enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
3	RW	0x0	BTTIE Block transfer/repetition period end interrupt enable When enabled, an interrupt will be asserted when the block transfer is finished if the channel conveys linear PCM or when the repetition period is reached if the channel conveys non-linear PCM. 0: disable 1: enable
2	RW	0x0	UDTIE User Data Interrupt 0: disable 1: enable If enabled, an interrupt will be asserted when the content of the user data register is fed into the corresponding shadow register
1:0	RO	0x0	reserved

SPDIF_INTSR
Address: Operational Base + offset (0x0010)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	SDBEIS Sample Date Buffer empty interrupt status 0: inactive 1: active
3	RW	0x0	BTTIS Block/Data burst transfer interrupt status 0: inactive 1: active
2	RW	0x0	UDTIS User Data Interrupt Status 0: inactive 1: active
1:0	RO	0x0	reserved

SPDIF_XFER
Address: Operational Base + offset (0x0018)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	XFER Transfer Start Register Transfer Start Register

SPDIF_SMPDR

Address: Operational Base + offset (0x0020)

Sample Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SMPDR Sample Data Register Sample Data Register

SPDIF_VLDFRn

Address: Operational Base + offset (0x0060)

Validity Flag Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	VLDFR_SUB_1 Validity Flag Subframe 1 Validity Flag Register 0
15:0	RW	0x0000	VLDFR_SUB_0 Validity Flag Subframe 0 Validity Flag for Subframe 0

SPDIF_USRDRn

Address: Operational Base + offset (0x0090)

User Data Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RW	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

SPDIF_CHNSRn

Address: Operational Base + offset (0x00c0)

Channel Status Register n

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CHNSR_SUB_1 Channel Status Subframe 1 Channel Status Bit for Subframe 1
15:0	RW	0x0000	CHNSR_SUB_0 Channel Status Subframe 0 Channel Status Bit for Subframe 0

SPDIF_BURTSINFO

Address: Operational Base + offset (0x00d0)

Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RW	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RW	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RW	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RW	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION

Address: Operational Base + offset (0x0104)

Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	REPETTION Repetition This define the repetition period when the channel conveys non-linear PCM

SPDIF_BURTSINFO_SHD
Address: Operational Base + offset (0x0108)
Shadow Channel Burst Info Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	PD pd Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RO	0x0	BSNUM Bitstream Number This field indicates the bitstream number. Usually the birstream number is 0.
12:8	RO	0x00	DATAINFO Data-type-dependent info This field gives the data-type-dependent info
7	RO	0x0	ERRFLAG Error Flag 0: indicates a valid burst-payload 1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RO	0x00	DATATYPE Data type 0000000: null data 0000001: AC-3 data 0000011: Pause data 0000100: MPEG-1 layer 1 data 0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 0000110: MPEG-2 data with extension 0000111: MPEG-2 AAC 0001000: MPEG-2, layer-1 low sampling frequency 0001001: MPEG-2, layer-2 low sampling frequency 0001010: MPEG-2, layer-3 low sampling frequency 0001011: DTS type I 0001100: DTS type II 0001101: DTS type III 0001110: ATRAC 0001111: ATRAC 2/3 0010000: ATRAC-X 0010001: DTS type IV 0010010: WMA professional type I 0110010: WMA professional type II 1010010: WMA professional type III 1110010: WMA professional type IV 0010011: MPEG-2 AAC low sampling frequency 0110011: MPEG-2 AAC low sampling frequency 1010011: MPEG-2 AAC low sampling frequency 1110011: MPEG-2 AAC low sampling frequency 0010100: MPEG-4 AAC 0110100: MPEG-4 AAC 1010100: MPEG-4 AAC 1110100: MPEG-4 AAC 0010101: Enhanced AC-3 0010110: MAT others: reserved

SPDIF_REPETTION_SHD

Address: Operational Base + offset (0x010c)

Shadow Channel Repetition Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	REPETTION Repetition This register provides the repetition of the bitstream when channel conveys non-linear PCM. In the design, it is define the length bwtween Pa of the two consecutive data-burst. For the same audio format, the definition is different. Please convert the actual repetition in order to comply with the design.

SPDIF_USRDR_SHDn
Address: Operational Base + offset (0x0190)
Shadow User Data Register n

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	USR_SUB_1 User Data Subframe 1 User Data Bit for Subframe 1
15:0	RO	0x0000	USR_SUB_0 User Data Subframe 0 User Data Bit for Subframe 0

21.5 Interface description

Table 21-1 SPDIF Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
spdif_8ch_sdo	O	IO_SPDIFtx_GPIO3d3	GRF_GPIO3D_IOMUX[7:6]=2’b01
spdif_8ch_sdo	O	IO_TESTCLKout1_SPDIF1tx_GPIO3d7	GRF_GPIO3D_IOMUX[15:14]=2’b10

The output of SPDIF module which signals as spdif_8ch_sdo is also connected to the audio interface of HDMI.

Table 21-2 Interface Between SPDIF And HDMI

Module Pin	Direction	Module Pin	Direction
mclk_spdif_8ch	O	ispdifclk	I
spdif_8ch_sdo	O	ispdifdata	I

21.6 Application Notes

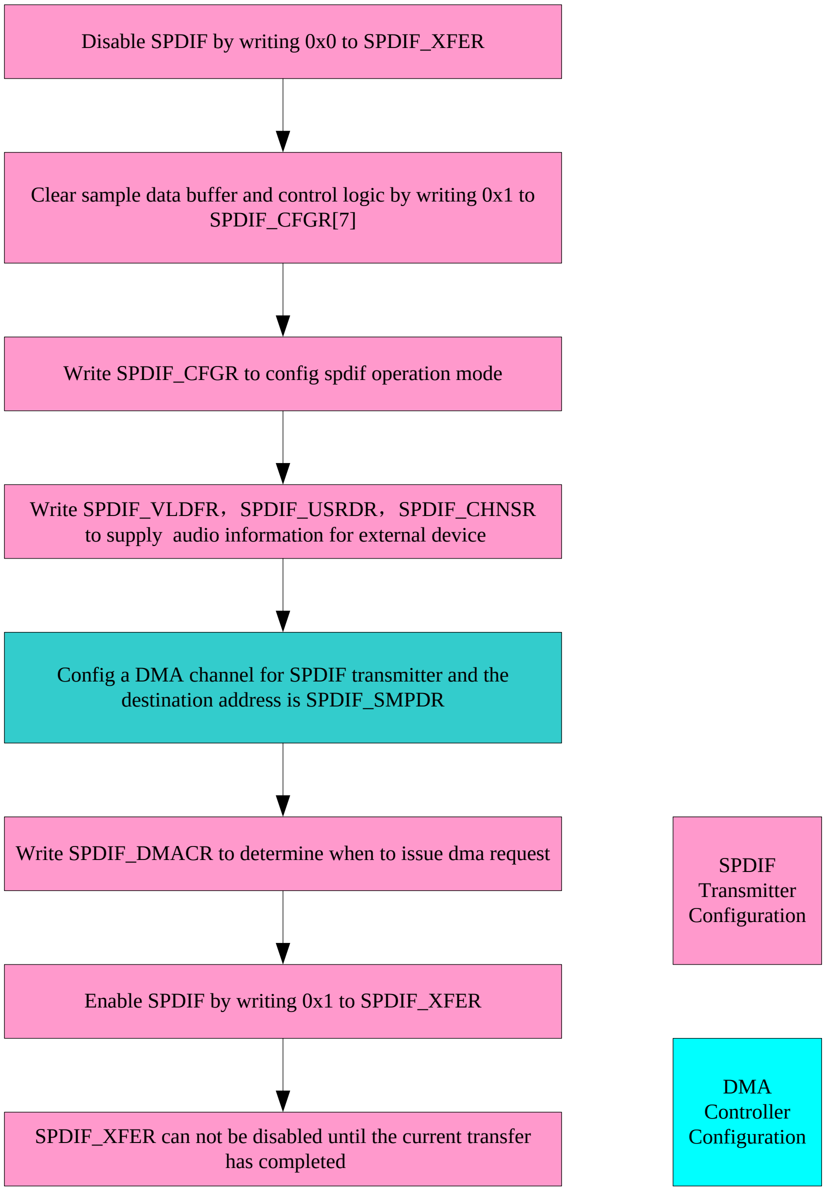


Fig.21-7 SPDIF transmitter operation flow chart

21.6.1 Channel Status Bit and Validity Flag Bit

Normally the channel status bits and validity flag bits are not necessarily updated frequently. If it is desired to change the channel status bits or validity flag, please write to the corresponding register after a block termination interrupt is asserted. The new value will take effect immediately.

21.6.2 User Data Bit

As the user data bits are updated frequently, the design takes use of the shadow register mechanism to store and convey the user data bit. When the SPDIF interface is disabled, the values of the shadow user data registers keeps the same with the corresponding user data registers. After the SPDIF starts, any change of the user data register will not go to the corresponding shadow user data registers until an user data interrupt is asserted. Therefore before the SPDIF transfer starts, prepare the first 384 user data bits by writing them to the SPDIF_USRDR registers. After the SPDIF transfer starts, writing the second

384 user data bits to the SPDIF_USRDR registers. Then wait for the assertion of user data interrupt. The second 384 user data bits goes to the shadow registers, and then third 384 user bits are written to SPDIF_USRDR.

21.6.3 Burst Info and Repetition

The shadow register mechanism is also applied to the data of burst info and repetition as the user data. The difference is that the update of shadow register will be taken after assertion of the block termination interrupt.

It is important to note that the repetition defined in the design is a little different from the repetition defined in IEC-61957. The repetition is always defined as the length (measured in IEC-60958 frame) between Pa of two consecutive data-bursts. Therefore the user needs to calculate the new repetition value if the definition of the repetition is different for some audio formats such as AC-3.

Chapter 22 GMAC Ethernet Interface

22.1 Overview

The GMAC Ethernet Controller 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.

22.1.1 Feature

- Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces
- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
  - Supports CSMA/CD Protocol for half-duplex operation
  - Supports packet bursting and frame extension in 1000 Mbps half-duplex operation
  - Supports IEEE 802.3x flow control for full-duplex operation
  - Optional forwarding of received pause control frames to the user application in full-duplex operation
  - Back-pressure support for half-duplex operation
  - Automatic transmission of zero-quanta pause frame on de-assertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
  - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the GMAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI

Master interface

- Supports internal loopback on the RGMII/RMII for debugging
- Debug status register that gives status of FSMs in Transmit and Receive data-paths and FIFO fill-levels.

22.2 Block Diagram

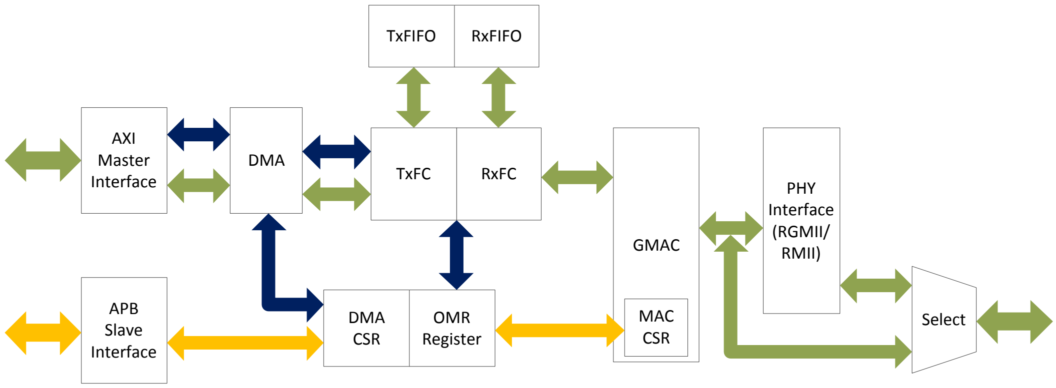


Table 22-1 GMACArchitecture

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 supports the PHY interfaces of reduced GMII (RGMII) and 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.

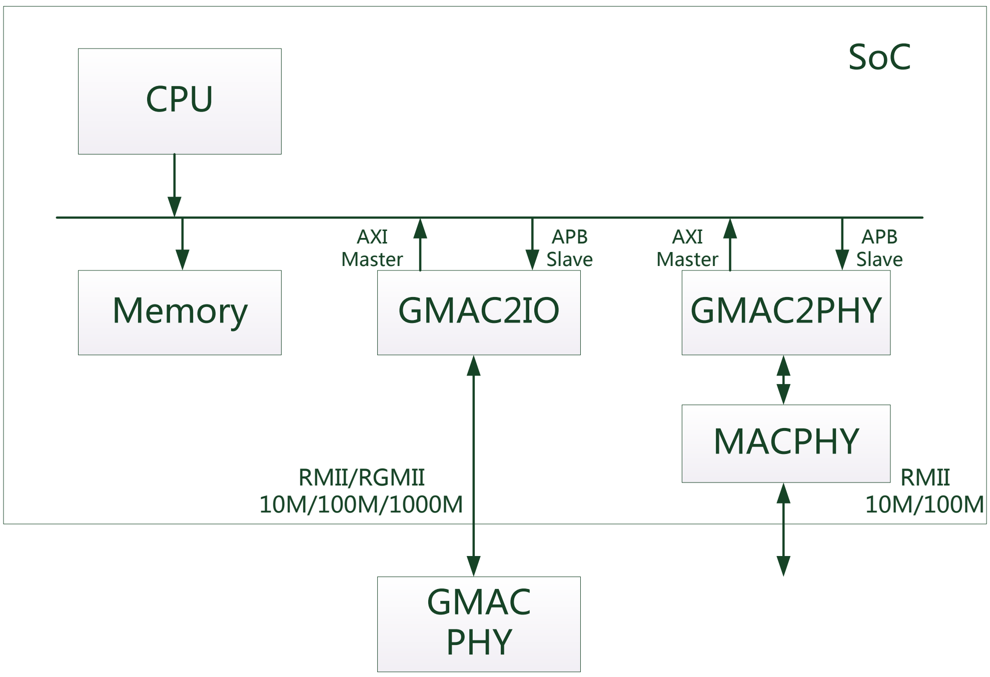


Fig.22-1 GMAC Architecture

There are two independent GMAC controllers named GMAC2IO and GMAC2PHY:

- GMAC2IO Supports 10/100/1000-Mbps data transfer rates with the RGMII interfaces and Supports 10/100-Mbps data transfer rates with the RMII interfaces
- GMAC2PHY Supports 10/100-Mbps data transfer rates with the RMII interfaces

22.3 Function Description

22.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig. 25-2.

Fig.22-2 MAC Block Diagram

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.

22.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the 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.

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.1-3. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

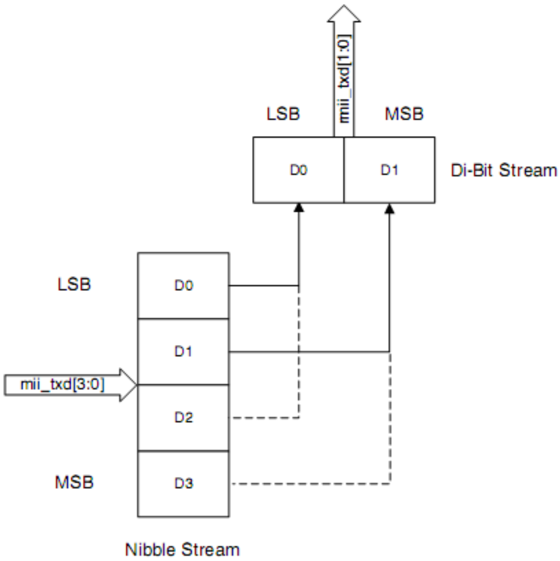


Fig.22-3 RMII transmission bit ordering

RMII Transmit Timing Diagrams

Fig.1-4 through 1-7 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.

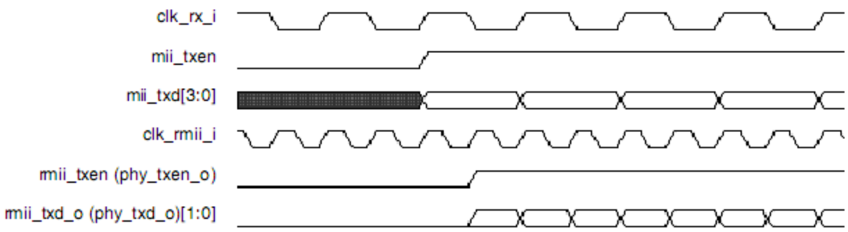


Fig. 22-4 Start of MII and RMII transmission in 100-Mbps mode

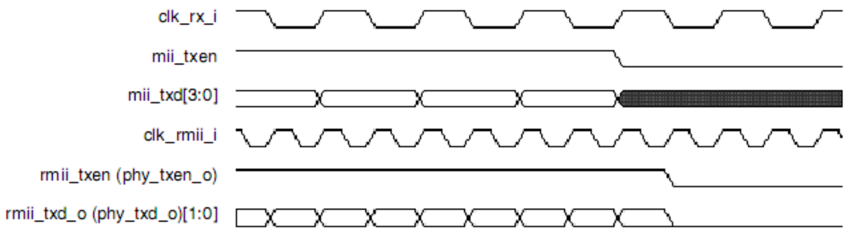


Fig. 22-5 End of MII and RMII Transmission in 100-Mbps Mode

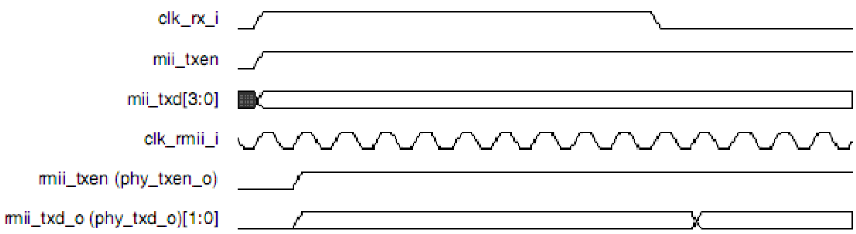


Fig. 22-6 Start of MII and RMII Transmission in 10-Mbps Mode

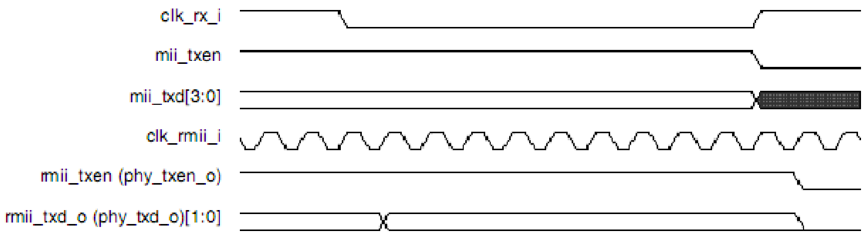


Fig. 22-7 End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-8. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

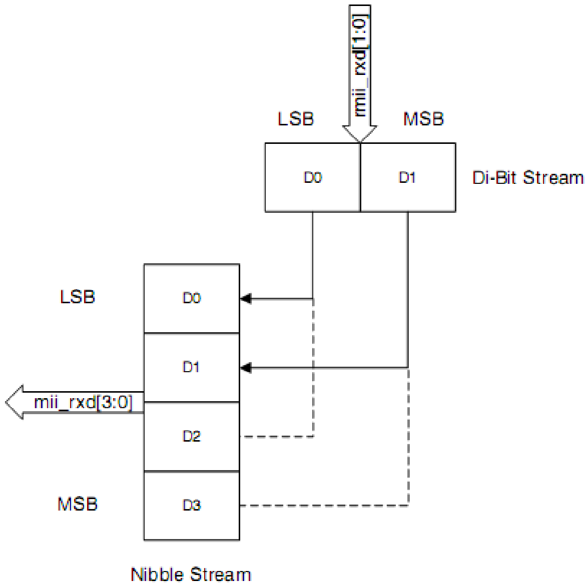


Fig. 22-8 RMII receive bit ordering

22.3.3 RGMII interface

The Reduced Gigabit Media Independent Interface (RGMII) specification reduces the pin count of the interconnection between the GMAC 10/100/1000 controller and the PHY for GMII and MII interfaces. To achieve this, the data path and control signals are reduced and multiplexed together with both the edges of the transmission and receive clocks. For gigabit operation the clocks operate at 125 MHz; for 10/100 operation, the clock rates are 2.5 MHz/25 MHz.

In the GMAC 10/100/1000 controller, the RGMII module is instantiated between the GMAC core's GMII and the PHY to translate the control and data signals between the GMII and RGMII protocols.

The RGMII block has the following characteristics:

- Supports 10-Mbps, 100-Mbps, and 1000-Mbps operation rates.
- For the RGMII block, no extra clock is required because both the edges of the incoming clocks are used.
- The RGMII block extracts the in-band (link speed, duplex mode and link status) status signals from the PHY and provides them to the GMAC core logic for link detection.

22.3.4 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the GMAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.
The GMAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_gmac. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	pclk_gmac	MDC Clock
0000	60-100 MHz	pclk_gmac/42
0001	100-150 MHz	pclk_gmac/62
0010	20-35 MHz	pclk_gmac/16
0011	35-60 MHz	pclk_gmac/26
0100	150-250 MHz	pclk_gmac/102
0101	250-300 MHz	pclk_gmac/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_gmac. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.
The frame structure on the MDIO line is shown below.

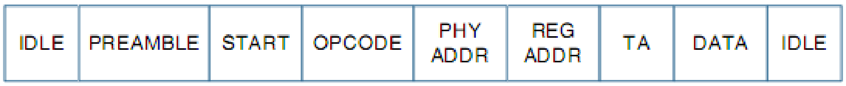


Fig. 22-9 MDIO frame structure

IDLE: The mdio line is three-state; there is no clock on gmii_mdc_o
PREAMBLE: 32 continuous bits of value 1
START: Start-of-frame is 2'b01
OPCODE: 2'b10 for read and 2'b01 for write
PHY ADDR: 5-bit address select for one of 32 PHYs
REG ADDR: Register address in the selected PHY
TA: Turnaround is 2'bZ0 for read and 2'b10 for Write
DATA: Any 16-bit value. In a write operation, the GMAC drives mdio; in a read operation, PHY drives it.

22.3.5 Power Management Block

Power management (PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and Magic Packets received by the GMAC. The PMT block sits on the receiver path of the GMAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.
When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.

Remote Wake-Up Frame Detection

When the GMAC is in sleep mode and the remote wake-up bit is enabled in register GMAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register GMAC_PMT_CTRL_STA.
PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received.
Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.
The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register GMAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the GMAC_PMT_CTRL_STA register to determine reception of a wake-up frame.

Magic Packet Detection

The Magic Packet frame is based on a method that uses Advanced Micro Device’s Magic Packet technology to power up the sleeping device on the network. The GMAC receives a specific packet of information, called a Magic Packet, addressed to the node on the network.
Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a GMAC Address appearing 16 times.
The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register GMAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48’hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the GMAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48’hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48’hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the GMAC address are detected.
If the MAC address of a node is 48'h00_11_22_33_44_55, then the GMAC scans for the data sequence:

Destination Address Source Address ………………………………. FF FFFFFFFFFF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC

Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

22.3.6 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.
The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.
The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

22.4 Register Description

22.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GMAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register
GMAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter
GMAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register
GMAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register
GMAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register
GMAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register
GMAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register
GMAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register
GMAC_DEBUG	0x0024	W	0x00000000	Debug register
GMAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register
GMAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register
GMAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register
GMAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register
GMAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register
GMAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register
GMAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register
GMAC_AN_ADV	0x00c8	W	0x000001e0	Auto Negotiation Advertisement Register
GMAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto Negotiation Link Partner Ability Register

Name	Offset	Size	Reset Value	Description
GMAC_AN_EXP	0x00d0	W	0x00000000	Auto Negotiation Expansion Register
GMAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register
GMAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register
GMAC_MMC_RX_INTR	0x0104	W	0x00000000	MMC Receive Interrupt Register
GMAC_MMC_TX_INTR	0x0108	W	0x00000000	MMC Transmit Interrupt Register
GMAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register
GMAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register
GMAC_MMC_TXOCTETCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
GMAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX Frame Good and Bad Counter
GMAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error
GMAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
GMAC_MMC_TXOCTETCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
GMAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
GMAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
GMAC_MMC_RXOCTETCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
GMAC_MMC_RXOCTETCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter
GMAC_MMC_RXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
GMAC_MMC_RXCRCERR	0x0194	W	0x00000000	MMC RX Carrier
GMAC_MMC_RXLENERR	0x01c8	W	0x00000000	MMC RX Length Error
GMAC_MMC_RXFIFOOVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow
GMAC_MMC_IPC_INT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register
GMAC_MMC_IPC_INTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register
GMAC_MMC_RXIPV4GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
GMAC_MMC_RXIPV4HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
GMAC_MMC_RXIPV6GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
GMAC_MMC_RXIPV6HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame

Name	Offset	Size	Reset Value	Description
GMAC_MMC_RXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
GMAC_MMC_RXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
GMAC_MMC_RXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
GMAC_MMC_RXIPV4HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error
GMAC_MMC_RXIPV6HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
GMAC_MMC_RXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
GMAC_MMC_RXTCPERROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
GMAC_MMC_RXICMPERROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
GMAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register
GMAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register
GMAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register
GMAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register
GMAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register
GMAC_STATUS	0x1014	W	0x00000000	Status Register
GMAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register
GMAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register
GMAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register
GMAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register
GMAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register
GMAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register
GMAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register
GMAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register
GMAC_CUR_HOST_TX_BUF_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register
GMAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Address Register

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

22.4.2 Detail Register Description

GMAC_MAC_CONF

Address: Operational Base + offset (0x0000)

MAC Configuration Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY.
23	RW	0x0	WD Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that.
22	RW	0x0	JD Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.
21	RW	0x0	BE Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in GMII Half-Duplex mode.
20	RO	0x0	reserved
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times

Bit	Attr	Reset Value	Description
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions.
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps
13	RW	0x0	DO Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting.
12	RW	0x0	LM Loopback Mode When this bit is set, the GMAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally.
11	RW	0x0	DM Duplex Mode When this bit is set, the GMAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration.

Bit	Attr	Reset Value	Description
10	RW	0x0	IPC Checksum Offload When this bit is set, the GMAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The GMAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared.
9	RW	0x0	DR Disable Retry When this bit is set, the GMAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the GMAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC will attempt retries based on the settings of BL.
8	RW	0x0	LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up
7	RW	0x0	ACS Automatic Pad/CRC Stripping When this bit is set, the GMAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the GMAC will pass all incoming frames to the Host unmodified.

Bit	Attr	Reset Value	Description
6:5	RW	0x0	BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 = r < 2^k
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmission state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive.
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmission state machine of the GMAC is enabled for transmission on the GMII/MII. When this bit is reset, the GMAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames.
2	RW	0x0	RE Receiver Enable When this bit is set, the receiver state machine of the GMAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the GMAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII.
1:0	RO	0x0	reserved

GMAC_MAC_FRM_FILT

Address: Operational Base + offset (0x0004)

MAC Frame Filter

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the GMAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter.
30:11	RO	0x0	reserved
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter.
9	RW	0x0	SAF Source Address Filter Enable The GMAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the GMAC drops the frame. When this bit is reset, then the GMAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison.
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter.

Bit	Attr	Reset Value	Description
7:6	RW	0x0	PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register GMAC_FLOW_CTRL[2]. 2'b00: GMAC filters all control frames from reaching the application. 2'b01: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: GMAC forwards all control frames to application even if they fail the Address Filter. 2'b11: GMAC forwards control frames that pass the Address Filter.
5	RW	0x0	DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames.
4	RW	0x0	PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit.
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.

Bit	Attr	Reset Value	Description
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

GMAC_HASH_TAB_HI

Address: Operational Base + offset (0x0008)

Hash Table High Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

GMAC_HASH_TAB_LO

Address: Operational Base + offset (0x000c)

Hash Table Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

GMAC_GMII_ADDR

Address: Operational Base + offset (0x0010)

GMII Address Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_gmac frequency used in your design. The suggested range of pclk_gmac frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. **Selection pclk_gmacMDC Clock** 0000 60-100 MHz pclk_gmac/42 0001 100-150 MHz pclk_gmac/62 0010 20-35 MHz pclk_gmac/16 0011 35-60 MHz pclk_gmac/26 0100 150-250 MHz pclk_gmac/102 0101 250-300 MHz pclk_gmac/124 0110, 0111 Reserved When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_gmac is of frequency 100 MHz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks. **Selection MDC Clock** 1000 pclk_gmac/4 1001 pclk_gmac/6 1010 pclk_gmac/8 1011 pclk_gmac/10 1100 pclk_gmac/12 1101 pclk_gmac/14 1110 pclk_gmac/16 1111 pclk_gmac/18
1	RW	0x0	GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register GMAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register GMAC_GMII_DATA.

Bit	Attr	Reset Value	Description
0	W1C	0x0	GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the GMAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the GMAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared.

GMAC_GMII_DATA

Address: Operational Base + offset (0x0014)

GMII Data Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.

GMAC_FLOW_CTRL

Address: Operational Base + offset (0x0018)

Flow Control Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.
6	RO	0x0	reserved
5:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted. **Selection** **Threshold** 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface.
3	RW	0x0	UP Unicast Pause Frame Detect When this bit is set, the GMAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the GMAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard.
2	RW	0x0	RFE Receive Flow Control Enable When this bit is set, the GMAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled.

Bit	Attr	Reset Value	Description
1	RW	0x0	TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the GMAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled.
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register GMAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC will reset this bit to 1'b0. The register GMAC_FLOW_CTRL should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC Core. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function.

GMAC_VLAN_TAG

Address: Operational Base + offset (0x001c)

VLAN Tag Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison.

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the GMAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames.

GMAC_DEBUG

Address: Operational Base + offset (0x0024)

Debug register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission.
24	RW	0x0	TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission.
23	RO	0x0	reserved
22	RW	0x0	TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO.
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission

Bit	Attr	Reset Value	Description
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state.
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO.
3	RO	0x0	reserved
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state.

GMAC_PMT_CTRL_STA

Address: Operational Base + offset (0x002c)

PMT Control and Status Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle.
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the GMAC (DAF) address recognition to be a wake-up frame.
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register.
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register.
4:3	RO	0x0	reserved
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception.
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception.
0	R/W SC	0x0	PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high.

GMAC_INT_STATUS

Address: Operational Base + offset (0x0038)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared.
6	RO	0x0	MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
5	RO	0x0	MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration.
4	RO	0x0	MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration.
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register GMAC_PMT_CTRL_STA.
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register.

GMAC_INT_MASK

Address: Operational Base + offset (0x003c)

Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register GMAC_INT_STATUS.
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register GMAC_INT_STATUS.

GMAC_MAC_ADDR0_HI

Address: Operational Base + offset (0x0040)

MAC Address0 High Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_MAC_ADDR0_LO

Address: Operational Base + offset (0x0044)

MAC Address0 Low Register

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames.

GMAC_AN_CTRL

Address: Operational Base + offset (0x00c0)

AN Control Register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the GMAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation.
11:10	RO	0x0	reserved
9	R/W SC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation.
8:0	RO	0x0	reserved

GMAC_AN_STATUS
Address: Operational Base + offset (0x00c4)
AN Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated.
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the GMAC supports auto-negotiation.
2	R/W SC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down.
1:0	RO	0x0	reserved

GMAC_AN_ADV
Address: Operational Base + offset (0x00c8)
Auto Negotiation Advertisement Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the GMAC does not support the next page.

Bit	Attr	Reset Value	Description
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred.
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the GMAC is capable of configuring the PAUSE function as defined in IEEE 802.3x.
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the GMAC supports Half-Duplex. This bit is tied to low (and RO) when the GMAC is configured for Full-Duplex-only operation.
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the GMAC supports Full-Duplex.
4:0	RO	0x0	reserved

GMAC_AN_LINK_PART_AB
Address: Operational Base + offset (0x00cc)
Auto Negotiation Link Partner Ability Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired.
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the GMAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved.
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner.
11:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x.
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode.
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode.
4:0	RO	0x0	reserved

GMAC_AN_EXP

Address: Operational Base + offset (0x00d0)

Auto Negotiation Expansion Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the GMAC does not support next page function.
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the GMAC. This bit will be cleared when read.
0	RO	0x0	reserved

GMAC_INTF_MODE_STA

Address: Operational Base + offset (0x00d8)

RGMII Status Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)

Bit	Attr	Reset Value	Description
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

GMAC_MMC_CTRL

Address: Operational Base + offset (0x0100)

MMC Control Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2K Bytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2K Bytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W SC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full.
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)

Bit	Attr	Reset Value	Description
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/W SC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

GMAC_MMC_RX_INTR

Address: Operational Base + offset (0x0104)

MMC Receive Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INTR

Address: Operational Base + offset (0x0108)

MMC Transmit Interrupt Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RX_INT_MSK

Address: Operational Base + offset (0x010c)

MMC Receive Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value.
20:19	RO	0x0	reserved
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value.
17:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value.
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value.
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TX_INT_MSK

Address: Operational Base + offset (0x0110)

MMC Transmit Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value.
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value.
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value.
18:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value.
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_TXOCTETCNT_GB

Address: Operational Base + offset (0x0114)

MMC TX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames.

GMAC_MMC_TXFRMCNT_GB

Address: Operational Base + offset (0x0118)

MMC TX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames.

GMAC_MMC_TXUNDFLWERR

Address: Operational Base + offset (0x0148)

MMC TX Underflow Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error.

GMAC_MMC_TXCARERR

Address: Operational Base + offset (0x0160)

MMC TX Carrier Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier).

GMAC_MMC_TXOCTETCNT_G
Address: Operational Base + offset (0x0164)
MMC TX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only.

GMAC_MMC_TXFRMCNT_G
Address: Operational Base + offset (0x0168)
MMC TX Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted.

GMAC_MMC_RXFRMCNT_GB
Address: Operational Base + offset (0x0180)
MMC RX Frame Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received.

GMAC_MMC_RXOCTETCNT_GB
Address: Operational Base + offset (0x0184)
MMC RX OCTET Good and Bad Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames.

GMAC_MMC_RXOCTETCNT_G
Address: Operational Base + offset (0x0188)
MMC RX OCTET Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames.

GMAC_MMC_RXMCFRMCNT_G
Address: Operational Base + offset (0x0190)

MMC RX Mulitcast Frame Good Counter

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received.

GMAC_MMC_RXCRCERR
Address: Operational Base + offset (0x0194)
MMC RX Carrier

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error.

GMAC_MMC_RXLENERR
Address: Operational Base + offset (0x01c8)
MMC RX Length Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠frame size), for all frames with valid length field.

GMAC_MMC_RXFIFOOVRFLW
Address: Operational Base + offset (0x01d4)
MMC RX FIFO Overflow

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow.

GMAC_MMC_IPC_INT_MSK
Address: Operational Base + offset (0x0200)
MMC Receive Checksum Offload Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_IPC_INTR
Address: Operational Base + offset (0x0208)
MMC Receive Checksum Offload Interrupt Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value.
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value.
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value.

Bit	Attr	Reset Value	Description
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.
4:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value.
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value.

GMAC_MMC_RXIPV4GFRM
Address: Operational Base + offset (0x0210)
MMC RX IPV4 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

GMAC_MMC_RXIPV4HDERRFRM
Address: Operational Base + offset (0x0214)
MMC RX IPV4 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

GMAC_MMC_RXIPV6GFRM
Address: Operational Base + offset (0x0224)
MMC RX IPV6 Good Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads.

GMAC_MMC_RXIPV6HDERRFRM
Address: Operational Base + offset (0x0228)
MMC RX IPV6 Head Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch).

GMAC_MMC_RXUDPERRFRM
Address: Operational Base + offset (0x0234)
MMC RX UDP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error.

GMAC_MMC_RXTCPERRFRM
Address: Operational Base + offset (0x023c)
MMC RX TCP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error.

GMAC_MMC_RXICMPERRFRM
Address: Operational Base + offset (0x0244)
MMC RX ICMP Error Frame

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error.

GMAC_MMC_RXIPV4HDERROCT
Address: Operational Base + offset (0x0254)
MMC RX OCTET IPV4 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter.

GMAC_MMC_RXIPV6HDERROCT

Address: Operational Base + offset (0x0268)

MMC RX OCTET IPV6 Head Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter.

GMAC_MMC_RXUDPERROCT

Address: Operational Base + offset (0x0274)

MMC RX OCTET UDP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors.

GMAC_MMC_RXTCPERROCT

Address: Operational Base + offset (0x027c)

MMC RX OCTET TCP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors.

GMAC_MMC_RXICMPERROCT

Address: Operational Base + offset (0x0284)

MMC RX OCTET ICMP Error

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors.

GMAC_BUS_MODE

Address: Operational Base + offset (0x1000)

Bus Mode Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved

Bit	Attr	Reset Value	Description
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address.
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high.
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations.
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x01	PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly. For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less. For RxFIFO, valid PBL range in full duplex mode is all.
7	RO	0x0	reserved
6:2	RW	0x00	DSL Descriptor Skip Length This bit specifies the number of dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode.
1	RO	0x0	reserved
0	R/W SC	0x1	SWR Software Reset When this bit is set, the MAC DMA Controller resets all GMAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion.

GMAC_TX_POLL_DEMAND

Address: Operational Base + offset (0x1004)

Transmit Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register GMAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes.

GMAC_RX_POLL_DEMAND

Address: Operational Base + offset (0x1008)

Receive Poll Demand Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register GMAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register GMAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state.

GMAC_RX_DESC_LIST_ADDR

Address: Operational Base + offset (0x100c)

Receive Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_TX_DESC_LIST_ADDR

Address: Operational Base + offset (0x1010)

Transmit Descriptor List Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only.

GMAC_STATUS

Address: Operational Base + offset (0x1014)

Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI GMAC PMT Interrupt This bit indicates an interrupt event in the GMAC core's PMT module. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
27	RO	0x0	GMI GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC core. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.
26	RO	0x0	GLI GMAC Line interface Interrupt This bit reflects an interrupt event in the GMAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the GMAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the GMAC subsystem (sbd_intr_o) is high when this bit is high.

Bit	Attr	Reset Value	Description
25:23	RO	0x0	EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register GMAC_STATUS[13]) set. This field does not generate an interrupt. Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor.
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory.

Bit	Attr	Reset Value	Description
16	W1C	0x0	NIS Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared.
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive FIFO Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared.
14	W1C	0x0	ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register GMAC_STATUS[6] automatically clears this bit.
13	W1C	0x0	FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses.
12:11	RO	0x0	reserved
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.

Bit	Attr	Reset Value	Description
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received.
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.
7	W1C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register GMAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA.
6	W1C	0x0	RI Receive Interrupt This bit indicates the completion of frame reception. Specific frame status information has been posted in the descriptor. Reception remains in the Running state.
5	W1C	0x0	UNF Transmit Underflow This bit indicates that the Transmit Buffer had an Underflow during frame transmission. Transmission is suspended and an Underflow Error TDES0[1] is set.
4	W1C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11].
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.

Bit	Attr	Reset Value	Description
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped.
0	W1C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor.

GMAC_OP_MODE

Address: Operational Base + offset (0x1018)

Operation Mode Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset.
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.
23:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register GMAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped.
20	W1C	0x0	FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active.
19:17	RO	0x0	reserved
16:14	RW	0x0	TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16

Bit	Attr	Reset Value	Description
13	RW	0x0	ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register GMAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register GMAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state.
12:11	RW	0x0	RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is de-asserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the de-assertion is effective only after flow control is asserted.
10:9	RW	0x0	RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high.
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled.

Bit	Attr	Reset Value	Description
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set.
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01).
5	RO	0x0	reserved
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained.

Bit	Attr	Reset Value	Description
1	RW	0x0	SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register GMAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register GMAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register GMAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state.
0	RO	0x0	reserved

GMAC_INT_ENA

Address: Operational Base + offset (0x101c)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register GMAC_STATUS[0]: Transmit Interrupt Register GMAC_STATUS[2]: Transmit Buffer Unavailable Register GMAC_STATUS[6]: Receive Interrupt Register GMAC_STATUS[14]: Early Receive Interrupt

Bit	Attr	Reset Value	Description
15	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register GMAC_STATUS[1]: Transmit Process Stopped Register GMAC_STATUS[3]: Transmit Jabber Timeout Register GMAC_STATUS[4]: Receive Overflow Register GMAC_STATUS[5]: Transmit Underflow Register GMAC_STATUS[7]: Receive Buffer Unavailable Register GMAC_STATUS[8]: Receive Process Stopped Register GMAC_STATUS[9]: Receive Watchdog Timeout Register GMAC_STATUS[10]: Early Transmit Interrupt Register GMAC_STATUS[13]: Fatal Bus Error
14	RW	0x0	ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled.
13	RW	0x0	FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled.
12:11	RO	0x0	reserved
10	RW	0x0	ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled.
9	RW	0x0	RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled.
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled.

Bit	Attr	Reset Value	Description
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled.
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled.
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled.
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled.
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled.
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled.

GMAC_OVERFLOW_CNT

Address: Operational Base + offset (0x1020)

Missed Frame and Buffer Overflow Counter Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1.
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.

GMAC_REC_INT_WDT_TIMER
Address: Operational Base + offset (0x1024)
Receive Interrupt Watchdog Timer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame.

GMAC_AXI_BUS_MODE
Address: Operational Base + offset (0x1028)
AXI Bus Mode Register

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the GMAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller.
30	RW	0x0	UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received.
29:22	RO	0x0	reserved
21:20	RW	0x1	WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1
19:18	RO	0x0	reserved
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit (register GMAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels.
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16.
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8.

Bit	Attr	Reset Value	Description
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4.
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register GMAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1.

GMAC_AXI_STATUS
Address: Operational Base + offset (0x102c)
AXI Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data.
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data.

GMAC_CUR_HOST_TX_DESC
Address: Operational Base + offset (0x1048)
Current Host Transmit Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_DESC
Address: Operational Base + offset (0x104c)
Current Host Receive Descriptor Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_TX_Buf_ADDR
Address: Operational Base + offset (0x1050)
Current Host Transmit Buffer Address Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

GMAC_CUR_HOST_RX_BUF_ADDR
Address: Operational Base + offset (0x1054)
Current Host Receive Buffer Adderss Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation.

22.5 Interface Description

Table 22-2 M0 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	IO_GMACclkm0_GPIO0D0vccio1	GPIO0D_IOMUX_SEL[1:0]=2’b01
mac_txen	O	IO_GMACtxenm0_GPIO0B4vccio1	GPIO0B_IOMUX_SEL[9:8]=2’b01
mac_txd1	O	IO_GMACtxd1m0_GPIO0C0vccio1	GPIO0C_IOMUX_SEL[1:0]=2’b01
mac_txd0	O	IO_GMACtxd0m0_GPIO0C1vccio1	GPIO0C_IOMUX_SEL[3:2]=2’b01
mac_rxdv	I	IO_GMACrxdvm0_GPIO0D1vccio1	GPIO0D_IOMUX_SEL[3:2]=2’b01
mac_rxer	I	IO_GMACrxerm0_GPIO0B5vccio1	GPIO0B_IOMUX_SEL[11:10]=2’b 01
mac_rxd1	I	IO_GMACrxd1m0_GPIO0B6vccio1	GPIO0B_IOMUX_SEL[13:12]=2’b 01
mac_rxd0	I	IO_GMACrxd0m0_GPIO0B7vccio1	GPIO0B_IOMUX_SEL[15:14]=2’b 01
Management interface			
mac_mdio	I/O	IO_GMACmdiom0_GPIO0B3vccio1	GPIO0B_IOMUX_SEL[7:6]=2’b01
mac_mdc	O	IO_GMACmdcm0_GPIO0C3vccio1	GPIO0C_IOMUX_SEL[7:6]=2’b01

Table 22-3 M0 RGMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RGMII/RMII interface			
mac_clk	I/O	IO_GMACclkm0_GPIO0D0vccio1	GPIO0D_IOMUX_SEL[1:0]=2’b01
mac_txclk	O	IO_GMACtxclkm0_GPIO0B0vccio1	GPIO0B_IOMUX_SEL[1:0]=2’b01
mac_txen	O	IO_GMACtxenm0_GPIO0B4vccio1	GPIO0B_IOMUX_SEL[9:8]=2’b01
mac_txd3	O	IO_GMACtxd3m0_GPIO0C7vccio1	GPIO0C_IOMUX_SEL[15:14]=2’b 01

mac_txd2	O	IO_GMACtxd2m0_GPIO0C6vccio1	GPIO0C_IOMUX_SEL[13:12]=2’b01
mac_txd1	O	IO_GMACtxd1m0_GPIO0C0vccio1	GPIO0C_IOMUX_SEL[1:0]=2’b01
mac_txd0	O	IO_GMACtxd0m0_GPIO0C1vccio1	GPIO0C_IOMUX_SEL[3:2]=2’b01
mac_rxclk	I	IO_GMACrxclkm0_GPIO0B2vccio1	GPIO0B_IOMUX_SEL[5:4]=2’b01
mac_rxdv	I	IO_GMACrxdvm0_GPIO0D1vccio1	GPIO0D_IOMUX_SEL[3:2]=2’b01
mac_rxd3	I	IO_GMACrxd3m0_GPIO0C4vccio1	GPIO0C_IOMUX_SEL[9:8]=2’b01
mac_rxd2	I	IO_GMACrxd2m0_GPIO0C5vccio1	GPIO0C_IOMUX_SEL[11:10]=2’b01
mac_rxd1	I	IO_GMACrxd1m0_GPIO0B6vccio1	GPIO0B_IOMUX_SEL[13:12]=2’b01
mac_rxd0	I	IO_GMACrxd0m0_GPIO0B7vccio1	GPIO0B_IOMUX_SEL[15:14]=2’b01
mac_crs	I	IO_GMACcrsm0_GPIO0B1vccio1	GPIO0B_IOMUX_SEL[3:2]=2’b01
mac_col	I	IO_GMACcolm0_GPIO0C2vccio1	GPIO0C_IOMUX_SEL[5:4]=2’b01
Management interface			
mac_mdio	I/O	IO_GMACmdiom0_GPIO0B3vccio1	GPIO0B_IOMUX_SEL[7:6]=2’b01
mac_mdc	O	IO_GMACmdcm0_GPIO0C3vccio1	GPIO0C_IOMUX_SEL[7:6]=2’b01

Table 22-3 M1 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	IO_I2S2mclk_GMACclkm1_GPIO1C5vccio4	GPIO1C_IOMUX_SEL[11:10]=2’b10
mac_txen	O	IO_I2S2sdom0_GMACtxenm1_PDMsdi2m1_GPIO1D1vccio4	GPIO1D_IOMUX_SEL[3:2]=2’b10
mac_txd1	O	IO_UART0rx_GMACtxd1m1_GPIO1B0vccio4	GPIO1B_IOMUX_SEL[1:0]=2’b10
mac_txd0	O	IO_UART0tx_GMACtxd0m1_GPIO1B1vccio4	GPIO1B_IOMUX_SEL[3:2]=2’b10
mac_rxdv	I	IO_I2S2sclkm0_GMACrxdvm1_PDMclkm1_GPIO1C6vccio4	GPIO1C_IOMUX_SEL[13:12]=2’b10
mac_rxer	I	IO_I2S2sdim0_GMACrxerm1_PDMsdi1m1_GPIO1D0vccio4	GPIO1D_IOMUX_SEL[1:0]=2’b10
mac_rxd1	I	IO_UART0rtsn_GMACrxd1m1_GPIO1B2vccio4	GPIO1B_IOMUX_SEL[5:4]=2’b10
mac_rxd0	I	IO_UART0ctsn_GMACrxd0m1_GPIO1B3vccio4	GPIO1B_IOMUX_SEL[7:6]=2’b10
Management interface			
mac_mdio	I/O	IO_SDMMC1detn_GMACmdiom1_PDMfsyncm1_GPIO1C3vccio4	GPIO1C_IOMUX_SEL[7:6]=2’b10
mac_mdc	O	IO_I2S2lrcktxm0_GMACmdcm1_PDMsdi0m1_GPIO1C7vccio4	GPIO1C_IOMUX_SEL[15:14]=2’b10

Table 22-4 M1 RGMII Interface Description

Module pin	Direction	Pad name	IOMUX setting

RGMII/RMII interface			
mac_clk	I/O	IO_I2S2mclk_GMACclkm1_GPIO1C5vccio4	GPIO1C_IOMUX_SEL[11:10]=2'b10
mac_txclk	O	IO_SDMMC1clkout_GMACtxclkm1_GPIO1B4vccio4	GPIO1B_IOMUX_SEL[9:8]=2'b10
mac_txen	O	IO_I2S2sdom0_GMACtxenm1_PDMsdi2m1_GPIO1D1vccio4	GPIO1D_IOMUX_SEL[3:2]=2'b10
mac_txd3	O	IO_SDMMC1d2_GMACtxd3m1_GPIO1C0vccio4	GPIO1C_IOMUX_SEL[1:0]=2'b10
mac_txd2	O	IO_SDMMC1d3_GMACtxd2m1_GPIO1C1vccio4	GPIO1C_IOMUX_SEL[3:2]=2'b10
mac_txd1	O	IO_UART0rx_GMACtxd1m1_GPIO1B0vccio4	GPIO1B_IOMUX_SEL[1:0]=2'b10
mac_txd0	O	IO_UART0tx_GMACtxd0m1_GPIO1B1vccio4	GPIO1B_IOMUX_SEL[3:2]=2'b10
mac_rxclk	I	IO_SDMMC1cmd_GMACrxclkm1_GPIO1B5vccio4	GPIO1B_IOMUX_SEL[11:10]=2'b10
mac_rxdv	I	IO_I2S2sclkm0_GMACrxdvm1_PDMclkm1_GPIO1C6vccio4	GPIO1C_IOMUX_SEL[13:12]=2'b10
mac_rxd3	I	IO_SDMMC1d0_GMACrxd3m1_GPIO1B6vccio4	GPIO1B_IOMUX_SEL[13:12]=2'b10
mac_rxd2	I	IO_SDMMC1d1_GMACrxd2m1_GPIO1B7vccio4	GPIO1B_IOMUX_SEL[15:14]=2'b10
mac_rxd1	I	IO_UART0rtsn_GMACrxd1m1_GPIO1B2vccio4	GPIO1B_IOMUX_SEL[5:4]=2'b10
mac_rxd0	I	IO_UART0ctsn_GMACrxd0m1_GPIO1B3vccio4	GPIO1B_IOMUX_SEL[7:6]=2'b10
mac_crs	I	IO_SDMMC1pwren_GMACcrsm1_GPIO1C2vccio4	GPIO1C_IOMUX_SEL[5:4]=2'b10
mac_col	I	IO_SDMMC1wp_GMACcolm1_GPIO1C4vccio4	GPIO1C_IOMUX_SEL[9:8]=2'b10
Management interface			
mac_mdio	I/O	IO_SDMMC1detn_GMACmdiom1_PDMfsyncm1_GPIO1C3vccio4	GPIO1C_IOMUX_SEL[7:6]=2'b10
mac_mdc	O	IO_I2S2lrcktxm0_GMACmdcm1_PDMsdi0m1_GPIO1C7vccio4	GPIO1C_IOMUX_SEL[15:14]=2'b10

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

22.6 Application Notes

22.6.1 Descriptors

The DMA in GMAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the Host physical memory space, and consists of an entire frame or part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled The descriptor ring and chain structure is shown in following figure.

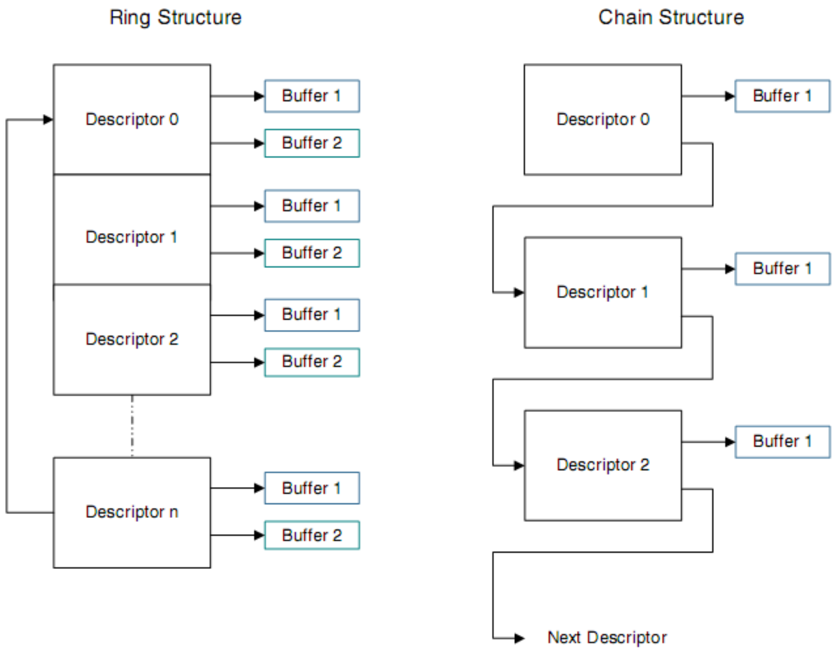


Fig. 22-10 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

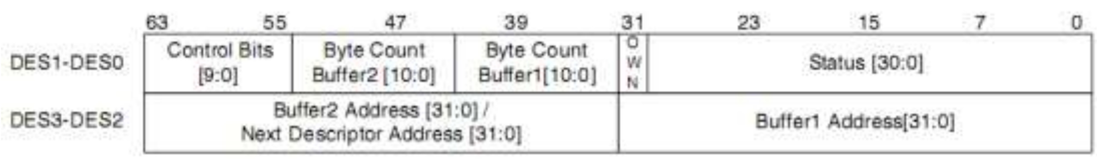


Fig. 22-11 Rx/Tx Descriptors definition

22.6.2 Receive Descriptor

The GMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMAalways attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.

In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)

RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 22-4 Receive Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: • RDES0[0]: Payload Checksum Error • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: IPC Checksum • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC Core.

Bit	Description
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the GMAC Core.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
7	IPC Checksum Error/Giant Frame When IP Checksum Engine is enabled, this bit, when set, indicates that the 16-bit IPv4 Header checksum calculated by the core did not match the received checksum bytes. The Error Summary bit[15] is NOT set when this bit is set in this mode.
6	LC: Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in Half-Duplex mode.
5	FT: Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the LT field is greater than or equal to 16'h0600). When this bit is reset, it indicates that the received frame is an IEEE802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
4	RWT: Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
3	RE: Receive Error When set, this bit indicates that the gmii_rxer_i signal is asserted while gmii_rxdv_i is asserted during frame reception. This error also includes carrier extension error in GMII and Half-duplex mode. Error can be of less/no extension, or error (rxd≠ 0f) during extension.
2	DE: Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in MII Mode.
1	CE: CRC Error

Bit	Description
	When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
0	Rx MAC Address/Payload Checksum Error When set, this bit indicates that the Rx MAC Address registers value (1 to 15) matched the frame's DA field. When reset, this bit indicates that the Rx MAC Address Register 0 value matched the DA field. If Full Checksum Offload Engine is enabled, this bit, when set, indicates the TCP, UDP, or ICMP checksum the core calculated does not match the received encapsulated TCP, UDP, or ICMP segment's Checksum field. This bit is also set when the received number of payload bytes does not match the value indicated in the Length field of the encapsulated IPv4 or IPv6 datagram in the received Ethernet frame.

Receive Descriptor 1 (RDES1)

RDES1 contains the buffer sizes and other bits that control the descriptor chain/ring.

Table 22-5 Receive Descriptor 1

Bit	Description
31	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the GMAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When RDES1[24] is set, RBS2 (RDES1[21-11]) is a "don't care" value. RDES1[25] takes precedence over RDES1[24].
23:22	Reserved.
21:11	RBS2: Receive Buffer 2 Size These bits indicate the second data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES3 (buffer2 address pointer) is not aligned to bus width. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. This field is not valid if RDES1[24] is set.
10:0	RBS1: Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 8 depending upon the bus widths (64), even if the value of RDES2 (buffer1 address pointer) is not aligned. In the case where the buffer size is not a multiple of 8, the resulting behavior is undefined. If this field is 0, the DMA ignores this

	buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 24).

Receive Descriptor 2 (RDES2)
RDES2 contains the address pointer to the first data buffer in the descriptor.

Table 22-6 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)
RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 22-7 Receive Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

22.6.3 Transmit Descriptor

The descriptor addresses must be aligned to the bus width used (64). Each descriptor is provided with two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory-management schemes.

Transmit Descriptor 0 (TDES0)
TDES0 contains the transmitted frame status and the descriptor ownership information.

Table 22-8 Transmit Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:17	Reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.
15	ES: Error Summary Indicates the logical OR of the following bits: • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the GMAC operates in Half-Duplex Mode.
10	NC: No Carrier

Bit	Description
	When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII Mode and 512 byte times including Preamble and Carrier Extension in RGMII Mode). Not valid if Underflow Error is set.
8	EC: Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If the DR (Disable Retry) bit in the GMAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the GMAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the GMAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register GMAC_STATUS[5]) and Transmit Interrupt (Register GMAC_STATUS [0]).
0	DB: Deferred Bit When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 22-9 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
	When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.
26	DC: Disable CRC When set, the GMAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
	These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 22-10 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 22-11 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

22.6.4 Programming Guide

DMA Initialization – Descriptors

The following operations must be performed to initialize the DMA.

1. Provide a software reset. This will reset all of the GMAC internal registers and logic. (GMAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll GMAC_OP_MODE[0], which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the Bus Mode Register by setting values in register GMAC_BUS_MODE
   a. Mixed Burst and AAL
   b. Fixed burst or undefined burst
   c. Burst length values and burst mode values.
   d. Descriptor Length (only valid if Ring Mode is used)
   e. Tx and Rx DMA Arbitration scheme
4. Program the AXI Interface options in the register GMAC_BUS_MODE
   a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])

5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.
6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.
7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register GMAC_RX_DESC_LIST_ADDR and GMAC_TX_DESC_LIST_ADDR).
8. Program the following fields to initialize the mode of operation by setting values in register GMAC_OP_MODE
   a. Receive and Transmit Store And Forward
   b. Receive and Transmit Threshold Control (RTC and TTC)
   c. Hardware Flow Control enable
   d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
   e. Error Frame and undersized good frame forwarding enable
   f. OSF Mode
9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register GMAC_STATUS).
10. Enable the interrupts by programming the interrupt enable register GMAC_INT_ENA.
11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register GMAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow.
1. Program the register GMAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of (GMAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in registerGMAC_GMII_ADDR (bits 15-11).
3. Provide the MAC address registers (GMAC_MAC_ADDR0_HI and GMAC_MAC_ADDR0_LO).
4. If Hash filtering is enabled in your configuration, program the Hash filter register (GMAC_HASH_TAB_HI and GMAC_HASH_TAB_LO).
5. Program the following fields to set the appropriate filters for the incoming frames in register GMAC_MAC_FRM_FILT
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter
   d. Unicast, Multicast, broad cast and control frames filter settings etc.
6. Program the following fields for proper flow control in register GMAC_FLOW_CTRL.
   a. Pause time and other pause frame control bits

b. Receive and Transmit Flow control bits
c. Flow Control Busy/Backpressure Activate

7. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
8. Program the appropriate fields in register GMAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
9. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.

Normal Receive and Transmit Operation

For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (GMAC_TX_POLL_DEMAND and GMAC_RX_POLL_DEMAND).
- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (GMAC_CUR_HOST_TX_DESC and GMAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (GMAC_CUR_HOST_TX_Buf_ADDR and GMAC_CUR_HOST_RX_BUF_ADDR).

Stop and Start Operation

When the transmission is required to be paused for some time then the following steps can be followed.

1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register GMAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register GMAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register GMAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

22.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.

The mux select rmii_speed is GRF_SOC_CON1[11].

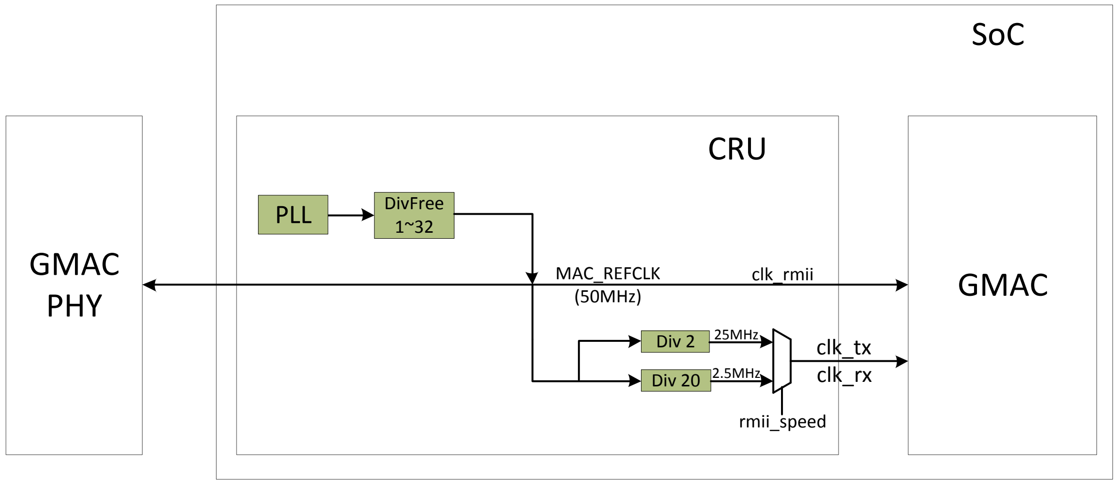


Fig. 22-12 RMII clock architecture when clock source from CRU

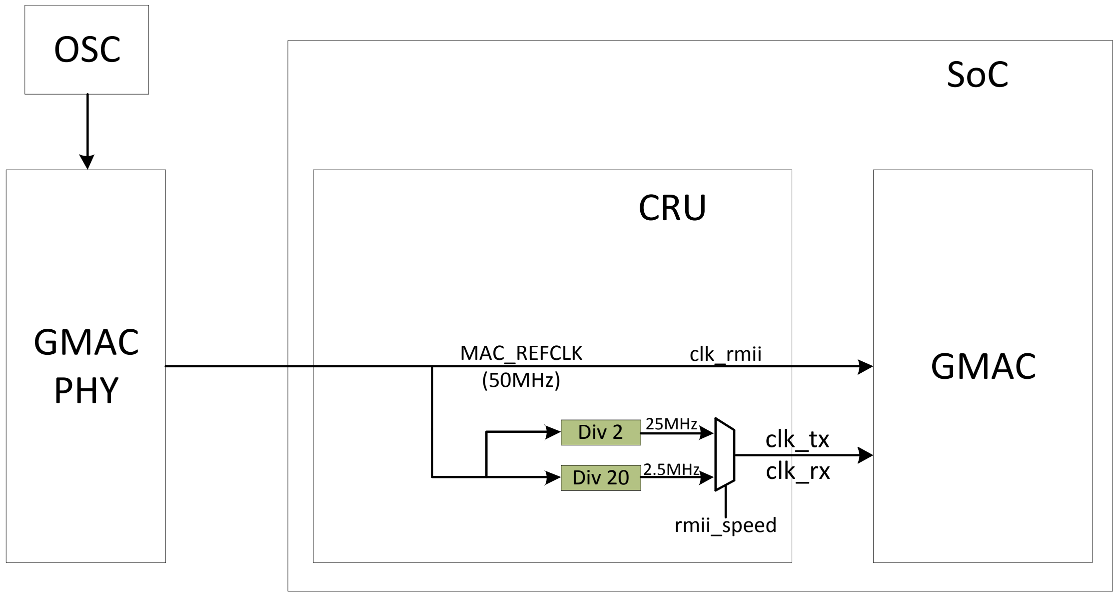


Fig. 22-13 RMII clock architecture when clock source from external OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.

In order to dynamically adjust the timing between TX/RX clocks with data, deleyline is integrated in TX and RX clock path. Register GRF_SOC_CON3[15:14] can enable the deleylines, and GRF_SOC_CON3[13:0] is used to determine the delay length. There are 100 deley elements in each delayline.

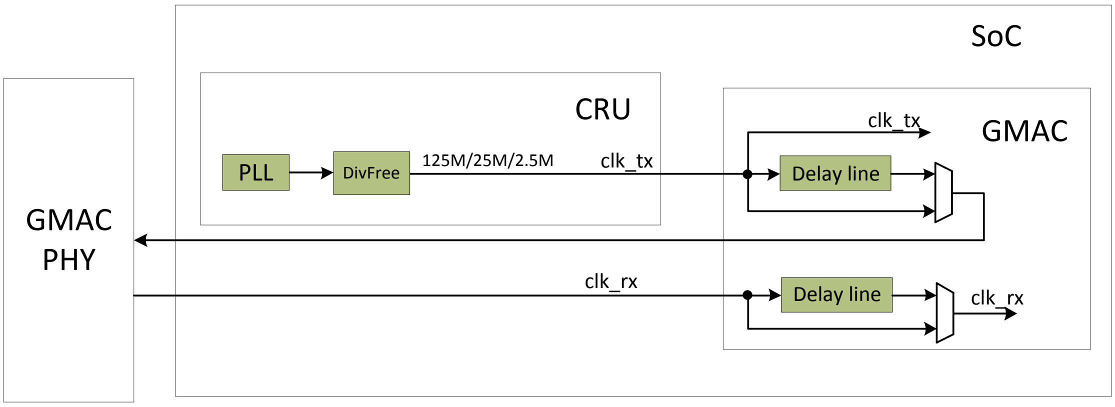


Fig. 22-14 RGMII clock architecture when clock source from CRU

22.6.6 Remote Wake-Up Frame Filter Register

The register wkupfmfilter_reg, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (wkupfmfilter_reg) must be written. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values in address (028) for wkupfmfilter_reg0, wkupfmfilter_reg1, ..., wkupfmfilter_reg7, respectively. Wkupfmfilter_reg is read in the same way.
The internal counter to access the appropriate wkupfmfilter_reg is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.

<table>
<tr><td>wkupfmfilter_reg0</td><td colspan="8">Filter 0 Byte Mask</td></tr>
<tr><td>wkupfmfilter_reg1</td><td colspan="8">Filter 1 Byte Mask</td></tr>
<tr><td>wkupfmfilter_reg2</td><td colspan="8">Filter 2 Byte Mask</td></tr>
<tr><td>wkupfmfilter_reg3</td><td colspan="8">Filter 3 Byte Mask</td></tr>
<tr><td>wkupfmfilter_reg4</td><td>RSVD</td><td>Filter 3 Command</td><td>RSVD</td><td>Filter 2 Command</td><td>RSVD</td><td>Filter 1 Command</td><td>RSVD</td><td>Filter 0 Command</td></tr>
<tr><td>wkupfmfilter_reg5</td><td colspan="2">Filter 3 Offset</td><td colspan="2">Filter 2 Offset</td><td colspan="2">Filter 1 Offset</td><td colspan="2">Filter 0 Offset</td></tr>
<tr><td>wkupfmfilter_reg6</td><td colspan="4">Filter 1 CRC - 16</td><td colspan="4">Filter 0 CRC - 16</td></tr>
<tr><td>wkupfmfilter_reg7</td><td colspan="4">Filter 3 CRC - 16</td><td colspan="4">Filter 2 CRC - 16</td></tr>
</table>

Fig. 22-15 Wake-Up Frame Filter Register

Filter i Byte Mask

This register defines which bytes of the frame are examined by filter i (0, 1, 2, and 3) in order to determine whether or not the frame is a wake-up frame. The MSB (thirty-first bit) must be zero. Bit j [30:0] is the Byte Mask. If bit j (byte number) of the Byte Mask is set, then Filter i Offset + j of the incoming frame is processed by the CRC block; otherwise Filter i Offset + j is ignored.

Filter i Command

This 4-bit command controls the filter i operation. Bit 3 specifies the address type, defining the pattern's destination address type. When the bit is set, the pattern applies to only multicast frames; when the bit is reset, the pattern applies only to unicast frame. Bit 2 and Bit 1 are reserved. Bit 0 is the enable for filter i; if Bit 0 is not set, filter i is disabled.

Filter i Offset

This register defines the offset (within the frame) from which the frames are examined by filter i. This 8-bit pattern-offset is the offset for the filter i first byte to examined. The minimum allowed is 12, which refers to the 13th byte of the frame (offset value 0 refers to the first byte of the frame).

Filter i CRC-16

This register contains the CRC_16 value calculated from the pattern, as well as the byte mask programmed to the wake-up filter register block.

22.6.7 System Consideration During Power-Down

GMAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.

The PMT interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.

The recommended power-down and wake-up sequence is as follows.

1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register GMAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.
6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the GMAC asserts the PMT interrupt signal and exits Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to the core.
9. Read the register GMAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

22.6.8 GRF Register Summary

GMAC2IO	
GRF Register	**Register Description**
GRF_MAC_CON0[6:0]	RGMII TX clock delayline value
GRF_MAC_CON0[13:7]	RGMII RX clock delayline value
GRF_MAC_CON1[0]	RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
GRF_MAC_CON1[1]	RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

GRF_MAC_CON1[2]	GMACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON1[3]	GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON1[6:4]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
GRF_MAC_CON1[7]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_MAC_CON1[9]	RMII mode selection 1'b1: RMII mode 1’b0: Reserved
GRF_MAC_CON1[10]	GMAC clock source selection 1’b1:clock from external OSC 1’b0:clock from CRU
GRF_MAC_CON1[12:11]	RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
GRF_CON_IOMUX[2]	GMAC IO selection 1’b1:select M1 1’b0:select M0
GRF_CON_IOMUX[10]	GMAC M1 channel select 1’b1:M1’s outputs come from M0’s pad when set GRF_CON_IOMUX[2] high 1’b0:GMAC controller connect M1 directly when set GRF_CON_IOMUX[2] high
GMAC2PHY	
GRF Register	**Register Description**
GRF_MAC_CON2[2]	GMACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON2[3]	GMAC transmit flow control When set high, instructs the GMAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the GMAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON2[6:4]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved

GRF_MAC_CON2[7]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_MAC_CON2[9]	RMII mode selection 1'b1: RMII mode 1’b0: Reserved
GRF_MAC_CON2[10]	GMAC clock source selection 1’b1:clock from external OSC 1’b0:clock from CRU

22.6.9 GMAC2IO Channel Description

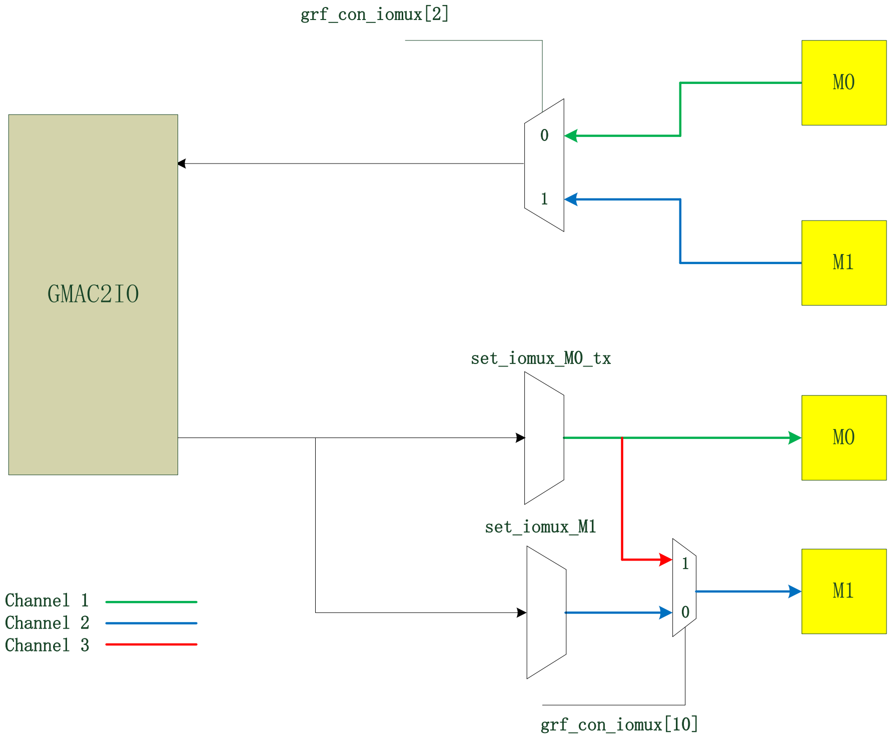


Fig. 22-16 gmac2io channel architecture

There are 3 different channels between GMAC controller and IO. The set_iomux_M0 and set_iomux_M1 in the upper figure means a series of IOMUX settings in table 1-1,1-2,1-3 and 1-4.

1. setting GRF_CON_IOMUX[2] low , GRF_CON_IOMUX[10] low and set_iomux_M0;
2. setting GRF_CON_IOMUX[2] high, GRF_CON_IOMUX[10] low and set_iomux_M1;
3. setting GRF_CON_IOMUX[2] high, GRF_CON_IOMUX[10] high , set_iomux_M1 and set_iomux_M0_tx;

Chapter 23 Pulse Density Modulation Interface Controller

23.1 Overview

The Pulse Density Modulation Interface Controller (PDMC) is a PDM interface controller and decoder that support PDM format. It integrates a clock generator driving the PDM microphone and embeds filters which decimate the incoming bit stream to obtain most common audio rates.

PDMC supports the following features:

- Support one internal 32-bit wide and 128-location deep FIFOs for receiving audio data
- Support receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of receive FIFO full interrupt
- Support combined interrupt output
- Support AHB bus slave interface
- Support DMA handshaking interface and configurable DMA water level
- Support PDM master receive mode
- Support 4 paths. Each path is composed of two digital microphone channels, the PDMC can be used with four stereo or eight mono microphones. Each path is enabled or disabled independently
- Support 16 ~24 bit sample resolution
- Support sample rate:

  8khz,16khz,32kHz,64kHz,128khz,11.025khz,22.05khz,44.1khz,88.2khz,176.4khz,12khz,24khz,48khz,96khz,192khz

- Support two 16-bit audio data store together in one 32-bit wide location
- Support programmable data sampling sensibility (rising or falling edge)

23.2 Block Diagram

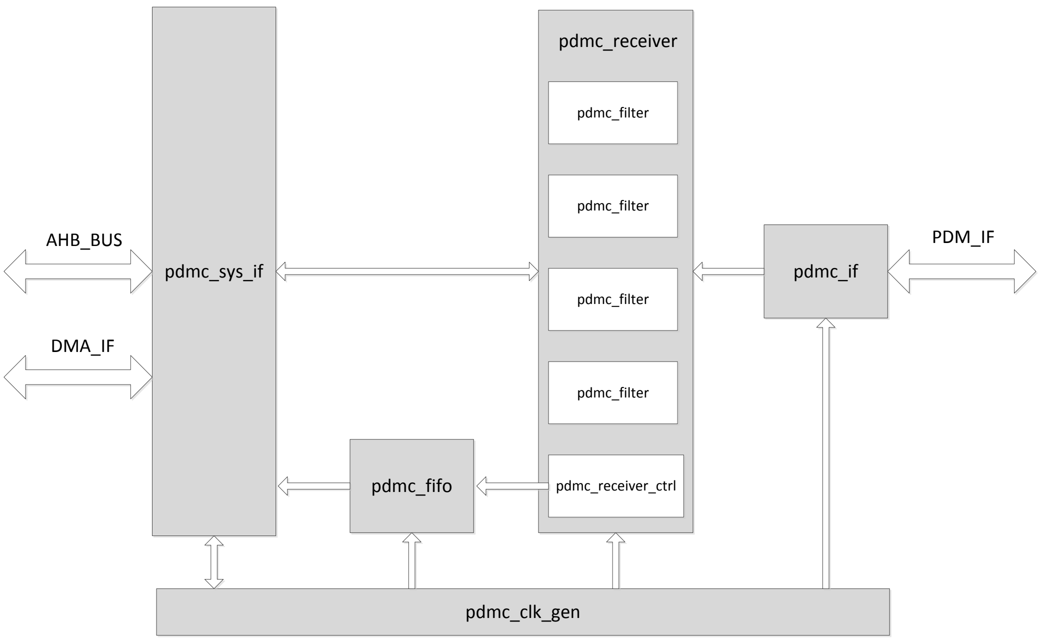


Fig.23-1 PDMC Block Diagram

System Interface

The system interface implements the APB slave operation. It contains not only control registers of receiver inside but also interrupt and DMA handshaking interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK, and by the divider of the module, the clock generator generates CLK_PDM to receiver.

Receiver

The receiver can act as a decimation filter of PDM. And export PCM format data.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 128.

PDM interface

The PDM interface implements PDM bit streams receive operation.

23.3 Function Description

23.3.1 AHB Interface

There is an AHB slave interface in PDMC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in 29.4.1.

23.3.2 PDM Interface

The PDM interface is a 5-wire interface. The PDMC module can support up to four external stereo and eight digital microphones.

Fig.1-2 and Fig.1-3 show two cases of use of the PDMC, but all configurations are possible with stereo and mono digital microphones.

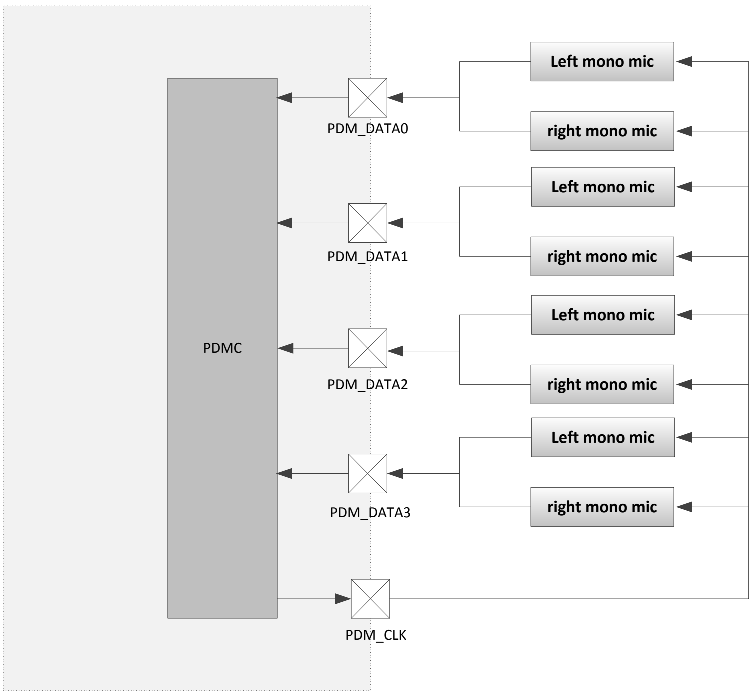


Fig.23-2 PDMC with Eight Mono MIC

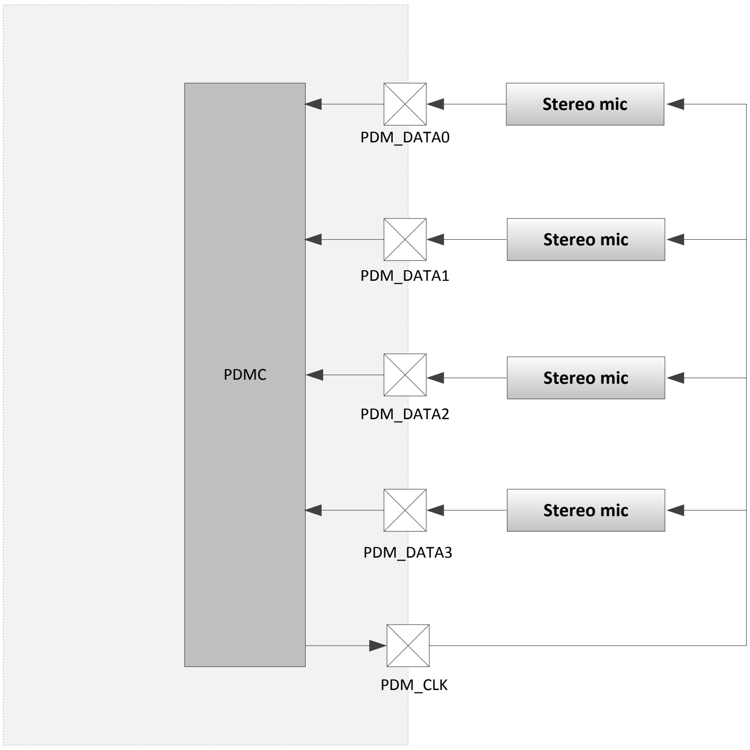


Fig.23-3 PDMC with Four Stereo MIC

The PDM interface consists of a serial-data shift clock output (PDM_CLK) and a serial data input (PDM_DATA). The clock is fanned out to both digital mics, and both digital mics' data (left channel and right channel) outputs share a single signal line. To share a single line, the digital mics tristate their output during one phase of the clock(high or low part of cycle, depending on how they are configured via their L/R input).

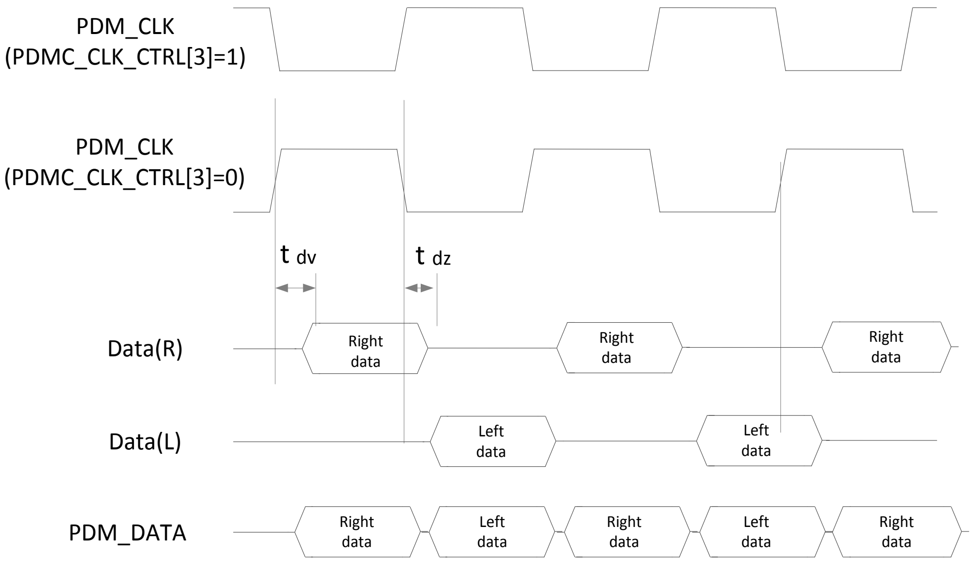


Fig.23-4 PDMC interface diagram with external MIC

23.3.3 Digital Filter

The external PDMIC generates a PDM stream of bits and transfers it in one period or one half-period of the clock provided by the PDMC. The aim of the PDMC is to process data from the PDM interface, decimate and filter the data, and store the processed data in the FIFO.

The four paths are identical. Each path is composed of a left and a right channel. The PDM interface delivers eight parallel data of 1bit. Each bit goes to a filter. The aim of the filter is to limit the noise and export PCM format audio data.

23.3.4 Clock Configuration

MCLK is the source clock signal. PDM_CLK is the output clocks generated in the PDMC and is fed to the external microphones. They are also the internal clock of the external microphones. User must take care about the value of PDM_CLK when selecting the source clock (MCLK).

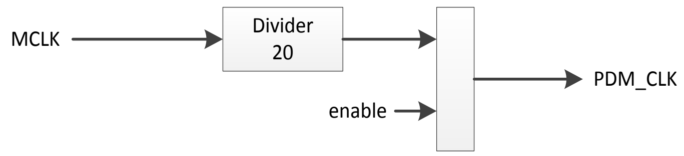


Fig.23-5 PDMC Clock Structure

Table 23-1 Relation between MCLK, ASP_CLK and sample rate

MCLK	PDM_CLK	Sample rate
61.44Mhz	3.072Mhz	12khz,24khz,48khz,96khz,192khz
56.448Mhz	2.8224Mhz	11.025khz,22.05khz,44.1khz,88.2khz,176.4khz
40.96Mhz	2.048Mhz	8khz,16khz,32kHz,64kHz,128khz

23.4 Register Description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PDMC_SYSCONFIG	0x00000	W	0x00000000	PDMC system config register
PDMC_CTRL0	0x00004	W	0x780003f7	PDMC control register 0
PDMC_CTRL1	0x00008	W	0x000000ff	PDMC control register 1
PDMC_CLK_CTRL	0x0000c	W	0x00000000	PDMC clock control register
PDMC_HPF_CTRL	0x00010	W	0x00000000	PDMC high pass filter control register
PDMC_FIFO_CTRL	0x00014	W	0x00000000	PDMC FIFO control register
PDMC_DMA_CTRL	0x00018	W	0x0000001f	PDMC DMA control register
PDMC_INT_EN	0x0001c	W	0x00000000	PDMC interrupt enable register

Name	Offset	Size	Reset Value	Description
PDMC_INT_CLR	0x00020	W	0x00000000	PDMC interrupt clear register
PDMC_INT_ST	0x00024	W	0x00000000	PDMC interrupt status register
PDMC_RXFIFO_DATA_REG	0x00030	W	0x00000000	PDMC receive FIFO data register
PDMC_DATA0R_REG	0x00034	W	0x00000000	PDMC path0 right channel data register
PDMC_DATA0L_REG	0x00038	W	0x00000000	PDMC path0 left channel data register
PDMC_DATA1R_REG	0x0003c	W	0x00000000	PDMC path1 right channel data register
PDMC_DATA1L_REG	0x00040	W	0x00000000	PDMC path1 left channel data register
PDMC_DATA2R_REG	0x00044	W	0x00000000	PDMC path2 right channel data register
PDMC_DATA2L_REG	0x00048	W	0x00000000	PDMC path2 left channel data register
PDMC_DATA3R_REG	0x0004c	W	0x00000000	PDMC path3 right channel data register
PDMC_DATA3L_REG	0x00050	W	0x00000000	PDMC path3 left channel data register
PDMC_DATA_VALID	0x00054	W	0x00000000	path data valid register
PDMC_VERSION	0x00058	W	0x59313030	PDMC version register
PDMC_RXDR	0x400~0x7fc	W	0x00000000	Receive FIFO data register

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

23.4.2 Detail Register Description

PDMC_SYSCONFIG

Address: Operational Base + offset (0x00000)

PDMC system config register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	rx_start RX transfer start bit RX Transfer start bit 0:stop RX transfer. 1:start RX transfer

Bit	Attr	Reset Value	Description
1	RO	0x0	reserved
0	RW	0x0	rx_clr PDMC RX logic clear PDMC RX logic clear; This is a self cleard bit. High active. Write 0x1: clear RX logic Write 0x0: no action Read 0x1: clear ongoing Read 0x0: clear done

PDMC_CTRL0

Address: Operational Base + offset (0x00004)

PDMC control register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	mode_sel Working mode selection: 0: PDM mode; 1: reserved;
30	RW	0x1	path3_en Path 3 enable; 1'b1: enable 1'b0: disable
29	RW	0x1	path2_en Path 2 enable; 1'b1: enable 1'b0: disable
28	RW	0x1	path1_en Path 1 enable; 1'b1: enable 1'b0: disable
27	RW	0x1	path0_en Path 0 enable; 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
26	RW	0x0	hwt_en HWT Halfword word transform Only valid when VDW select 16bit data. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
25	RW	0x0	Reserved
24	RW	0x0	Reserved
23	RW	0x0	Reserved
22	RW	0x0	Reserved
21:19	RW	0x0	Reserved
18	RW	0x0	Reserved
17	RW	0x0	Reserved
16	RW	0x0	Reserved
15:13	RO	0x0	reserved
12:10	RW	0x0	Reserved
9:5	RW	0x1f	Reserved
4:0	RW	0x17	data_vld_width (Can be written only when SYSCONFIG[2] is 0.) Valid Data width 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 23:24bit

PDMC_CTRL1

Address: Operational Base + offset (0x00008)

PDMC control register 1

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x0ff	Reserved

PDMC_CLK_CTRL

Address: Operational Base + offset (0x0000c)

PDMC clock control register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	Reserved
5	RW	0x0	pdm_clk_en Pdm clk enable.working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 0:pdm clk disable 1:pdm clk enable
4	RO	0x0	reserved
3	RW	0x0	clk_polar PDM_CLK polarity selection (Can be written only when SYSCONFIG[2] is 0.) 0: no inverted 1: inverted
2:0	RW	0x0	pdm_ds_ratio DS_RATIO,working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 3'b000: sample rate 192k/176.5k/128k 3'b001: sample rate 96kk/88.2k/64k 3'b010: sample rate 48kk/44.1k/32k 3'b011: sample rate 24kk/22.05k/16k 3'b100: sample rate 12kk/11.025k/8k

PDMC_HPF_CTRL

Address: Operational Base + offset (0x00010)

PDMC high pass filter control register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	hpfle HPFLE high pass filter enable for left channel 1'b0: high pass filter for right channel is diabled. 1'b1: high pass filter for right channel is enabled.
2	RW	0x0	hpfre HPFRE high pass filter enable for right channel 1'b0: high pass filter for right channel is diabled. 1'b1: high pass filter for right channel is enabled.
1:0	RW	0x0	hpf_cf HPF_CF high pass filter configure register high pass filter configure register 2'b00: 3.79Hz 2'b01: 60Hz 2'b10: 243Hz 2'b11: 493Hz

PDMC_FIFO_CTRL

Address: Operational Base + offset (0x00014)

PDMC fifo control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	rft Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO threshold interrupt is triggered.
7:0	RO	0x00	rfl RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.

PDMC_DMA_CTRL

Address: Operational Base + offset (0x00018)

PDMC dma control register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	rde Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
7	RO	0x0	reserved
6:0	RW	0x1f	rdl Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.

PDMC_INT_EN

Address: Operational Base + offset (0x0001c)

PDMC interrupt enable register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	rxoie RX overflow interrupt enable 0:disable 1:enable
0	RW	0x0	rxtie RX threshold interrupt enable 0:disable 1:enable

PDMC_INT_CLR

Address: Operational Base + offset (0x00020)

PDMC interrupt clear register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	rxoic RX overflow interrupt clear, high active, auto clear.
0	RO	0x0	reserved

PDMC_INT_ST

Address: Operational Base + offset (0x00024)

PDMC interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	rxoi RX overflow interrupt 0:inactive 1:active
0	RO	0x0	rxfi RX full interrupt 0:inactive 1:active

PDMC_RXFIFO_DATA_REG

Address: Operational Base + offset (0x00030)

PDMC receive fifo data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdr Receive FIFO shadow Register When the register is read, data in the receive FIFO is accessed.

PDMC_DATA0R_REG

Address: Operational Base + offset (0x00034)

PDMC path0 right channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0r Data of the path 0 right channel

PDMC_DATA0L_REG

Address: Operational Base + offset (0x00038)

PDMC path0 leght channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0l Data of the path 0 left channel

PDMC_DATA1R_REG

Address: Operational Base + offset (0x0003c)

PDMC path1 right channel data register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	data1r Data of the path 1 right channel

PDMC_DATA1L_REG

Address: Operational Base + offset (0x00040)

PDMC path1 left channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data1l Data of the path 1 left channel

PDMC_DATA2R_REG

Address: Operational Base + offset (0x00044)

PDMC path2 right channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2r Data of the path 2 right channel

PDMC_DATA2L_REG

Address: Operational Base + offset (0x00048)

PDMC path2 left channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2l Data of the path 2 left channel

PDMC_DATA3R_REG

Address: Operational Base + offset (0x0004c)

PDMC path3 right channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3r Data of the path 3 right channel

PDMC_DATA3L_REG

Address: Operational Base + offset (0x00050)

PDMC path3 left channel data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3l Data of the path 3 left channel

PDMC_DATA_VALID

Address: Operational Base + offset (0x00054)

path data valid register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RC	0x0	path0_vld 0: DATA0R_REG, DATA0L_REG value is invalid; 1: DATA0R_REG, DATA0L_REG value is valid;
2	RC	0x0	path1_vld 0: DATA1R_REG, DATA1L_REG value is invalid; 1: DAT1R_REG, DATA1L_REG value is valid;
1	RC	0x0	path2_vld 0: DATA2R_REG, DATA2L_REG value is invalid; 1: DATA2R_REG, DATA2L_REG value is valid;
0	RC	0x0	path3_vld 0: DATA3R_REG, DATA3L_REG value is invalid; 1: DATA3R_REG, DATA3L_REG value is valid;

PDMC_VERSION

Address: Operational Base + offset (0x00058)

PDMC version register

Bit	Attr	Reset Value	Description
31:0	RO	0x59313030	version PDMC version

23.5 Interface Description

Table 23-2 PDMC Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
O_pdm_clk	O	IO_I2S1sclk_PDMclkm0_TSPd7m1_CIFdata7m1_GPIO2C2vccio5/IO_I2Ssclkm0_GMACrxdvm1_PDMclkm1_GPIO1C6vccio4	PDMclkm0: GPIO2CL_IO[7:6]=2 PDMclkm1: GPIO1C_IO[13:12]=3
O_pdm_fsync	O	IO_I2S1sdo_PDMfsyncm0_GPIO2C7vccio5/IO_SDMMC1detn_GMACmdiom1_PDMfsyncm1_GPIO1C3vccio4	PDMfsyncm0: GPIO2CH_IO[15:14]=2 PDMfsyncm1: GPIO1C_IO[7:6]=3

I_pdm_data0	I	IO_I2S1sdi_PDMsdi0m0_CARDclkm1_GPIO2C3vccio5 /IO_I2S2lrcktxm0_GMACmdcm1_PDMsdi0m1_GPIO1C7vccio4	PDMsdi0m0: GPIO2CL_IO[10:9]=2 PDMsdi0m1: GPIO1C_IO[1:0]=3
I_pdm_data1	I	IO_I2S1sdio1_PDMsdi1m0_CARDrstm1_GPIO2C4vccio5 /IO_I2S2sdim0_GMACrxerm1_PDMsdi1m1_GPIO1D0vccio4	PDMsdi1m0: GPIO2CL_IO[13:12]=2 PDMsdi1m1: GPIO1D_IO[1:0]=3
I_pdm_data2	I	IO_I2S1sdio2_PDMsdi2m0_CARDdetm1_GPIO2C5vccio5 /IO_I2S2sdom0_GMACtxenm1_PDMsdi2m1_GPIO1D1vccio4	PDMsdi2m0: GPIO2CH_IO[1:0]=2 PDMsdi2m1: GPIO1D_IO[3:2]=3
I_pdm_data3	I	IO_I2S1sdio3_PDMsdi3m0_CARDiom1_GPIO2C6vccio5 /IO_I2S2lrckrxm0_CLKout_gmacm2_PDMsdi3m1_GPIO1D2vccio4	PDMsdi3m0: GPIO2CH_IO[4:3]=2 PDMsdi3m1: GPIO1D_IO[5:4]=3

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.

23.6 Application Notes

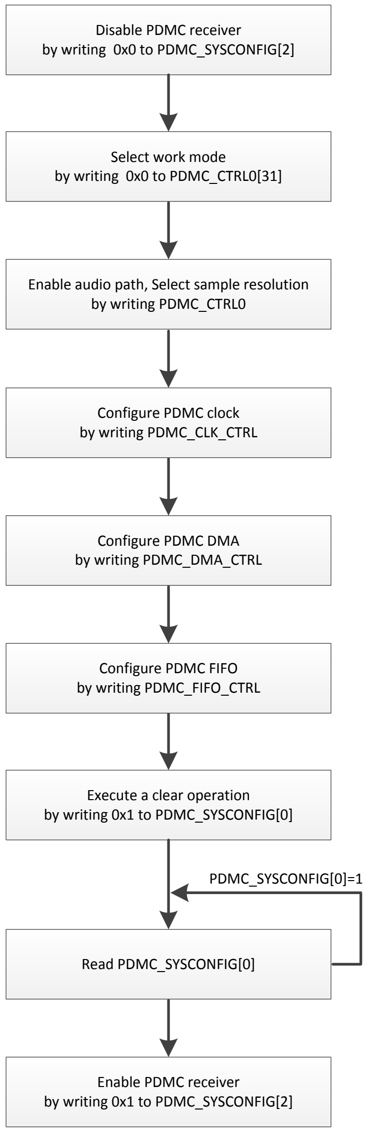


Table 23-3 PDMC operation flow

Chapter 24 Smart Card Reader (SCR)

24.1 Overview

The Smart Card Reader (SCR) is a communication controller that transmits data between the superior system and the Smart Card. The controller can perform a complete smart card session, including card activation, card deactivation, cold/warm reset, Answer to Reset (ATR) response reception, data transfers, etc.

SCR supports the following features:

- Supports the ISO/IEC 7816-3:1997(E) and EMV2000 (4.0) specifications
- Performs functions needed for complete smart card sessions, including:
  - Card activation and deactivation
  - Cold/warm reset
  - Answer to Reset (ATR) response reception
  - Data transfers to and from the card
- Extensive interrupt support system
- Adjustable clock rate and bit (baud) rate
- Configurable automatic byte repetition
- Handles commonly used communication protocols:
  - T=0 for asynchronous half-duplex character transmission
  - T=1 for asynchronous half-duplex block transmission
- Automatic convention detection
- Configurable timing functions:
  - Smart card activation time
  - Smart card reset time
  - Guard time
  - Timeout timers
- Automatic operating voltage class selection
- Supports synchronous and any other non-ISO 7816 and non-EMV cards
- Advanced Peripheral Bus (APB) slave interface for easy integration with AMBA-based host systems Block Diagram

24.2 Block Diagram

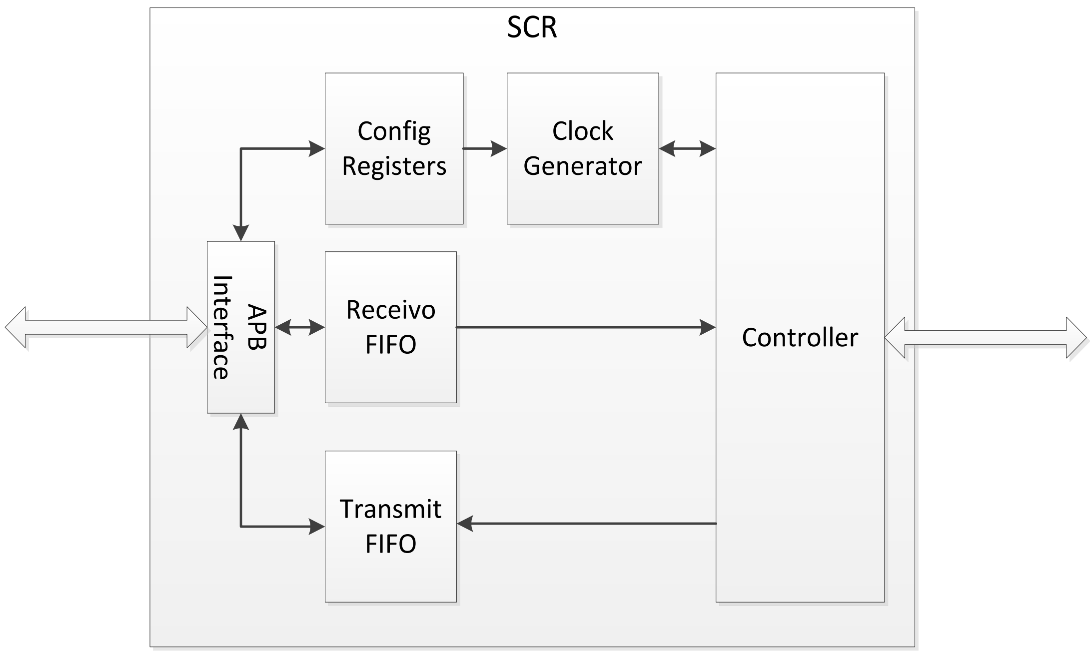


Fig. 24-1 SCR Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface.

24.2.1 APB Interface

The host processor gets access to PWM Register Block through the APB slave interface.

24.2.2 Configuration Registers

The Configuration Registers block provides control over all functions of the Smart Card Reader

24.2.3 Controller

The Controller is the main block in the SCR core. This block controls receiving characters transmitted by the Smart Card, storing them in the RX FIFO, and transmitting them to the Smart Card. This block also performs card activation, deactivation, and cold and warm reset. After the card is reset, the Answer To Reset (ATR) sequence is received by the controller and stored in RX FIFO.
The parallel to serial conversion needed to transmit data from a Smart Card Reader to a Smart Card and the serial to parallel conversion needed to transmit data in the opposite direction is performed by the UART. The UART also performs the guard time, parity checking and character repeating functions.

24.2.4 Receive FIFO

The Receive FIFO is used to store the data received from the Smart Card until the data is read out by the superior system.

24.2.5 Transmit FIFO

The Transmit FIFO is used to store the data to be transmitted to the Smart Card.

24.2.6 Clock Generator

The Clock Generator generates the Smart Card Clock signal and the Baud Clock Impulse signal, used in timing the Smart Card Reader.)

24.3 Function Description

A Smart Card session consists of following stages:

1. Smart Card insertion
2. Activation of contacts and cold reset sequence
3. Answer To Reset sequence (ATR)
4. Execution of transaction
5. Deactivation of contacts
6. Smart Card removal

24.3.1 Smart Card Insertion

A Smart Card session starts with the insertion of the Smart Card. This event is signaled to the SCR using the SCDETECT input. The SCPRESENT bit is set and also the SCINS interrupt is asserted (if enabled).
When the external card detect switch is not used, the input pin SCDETECT must be tied to inactive state.

24.3.2 Automatic operating voltage class selection

There are three operating classes (1.8V - class C, 3V - class B and 5V - class A) defined in ISO/IEC 7816-3(2006) specification. Only 1.8V and 3.3V are supported by the SCR.

Before the activation of contacts, operating classes have to be enabled via bits VCC18, VCC33 in CTRL2 register. In case that no operating class is enabled, the controller performs activation for all two voltage classes (1.8V, 3V) in sequence.
When Smart Card Reader performs activation of contacts the lowest enabled voltage class is automatically applied first. When the first character start bit of ATR sequence is received, the selected voltage class is correct (even if the ATR is then received with errors). When the ATR sequence reception does not start, ATRFAIL interrupt is not activated, deactivation is performed and next higher enabled voltage class is applied. If the ATR sequence reception does not start and no other higher class is enabled was already applied the ATRFAIL interrupt is activated and the last applied voltage class remains active.
After the automatic voltage class selection is finished the selected class can be read from bits VCC18, VCC33 in CTRL2 register. If the automatic voltage class selection fails, these bits remain untouched.
There is a delay applied between deactivation of contacts with lower voltage class and activation of contacts with higher voltage class. This delay should be at least 10 ms according to the ISO/IEC 7816-3 specification.

24.3.3 Activation of Contacts and Cold Reset Sequence

When the Smart Card is properly inserted and the ACT bit in CTRL2 register is asserted, the activation of contacts can be started. The duration of each part of the activation is the time Ta, which is equal to the ADEATIME register value. If no Vpp is necessary, the activation and deactivation part of Vpp can be omitted by clearing the AUTOADEAVPP bit in SCPADS register.
The Cold Reset sequence follows immediately after the activation. Time (Tc) is the duration of the Reset. The EMV specification recommends that this value should be between 40000 and 45000. The activation of contacts and cold reset sequence is shown inFig. 24-2.

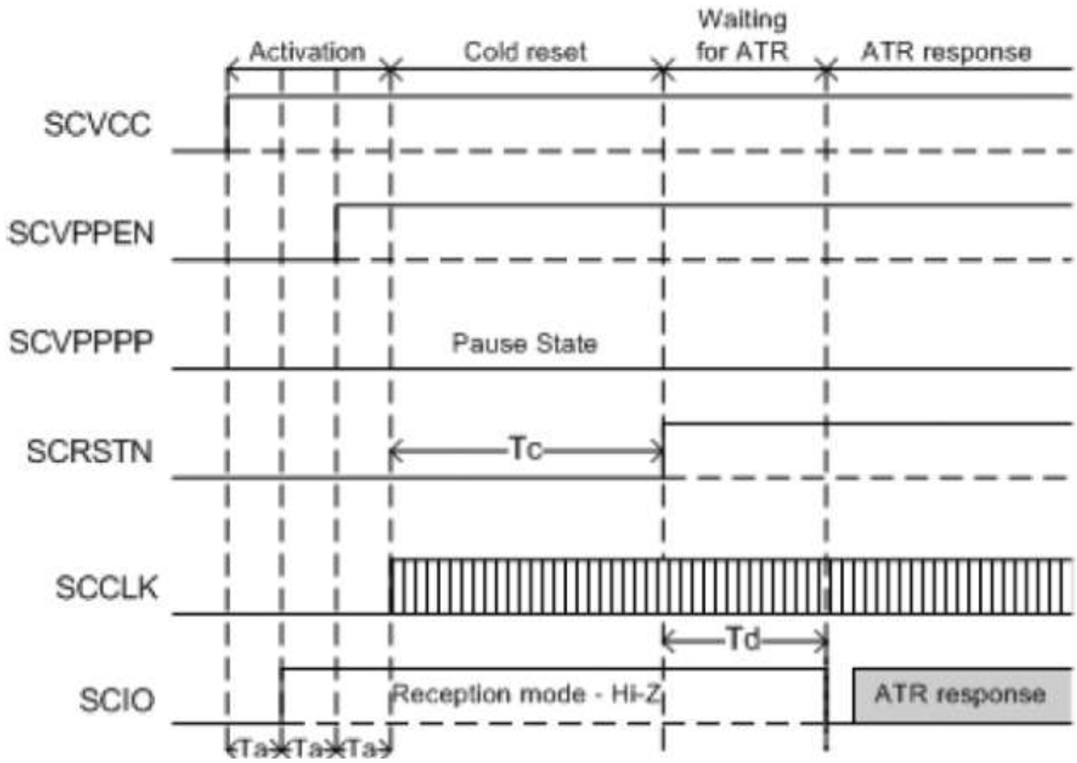


Fig. 24-2 Activation, Cold Reset and ATR

24.3.4 Execution of Transaction

All transfers between the Smart Card Reader and a Smart Card are under the control of the superior system. It controls the number of characters sent to the Smart Card and it knows the number of characters expected to be returned from the Smart Card.

24.3.5 Warm Reset

The Warm Reset sequence is initialized by setting the WRST bit in the CTRL2 register to '1'.Smart Card Reader drives the SCRSTN signal to '0' to perform the Warm Reset as shown in Fig. 24-3. After the SCRSTN assertion, the Warm Reset sequence then continues the same way as the Cold Reset sequence.

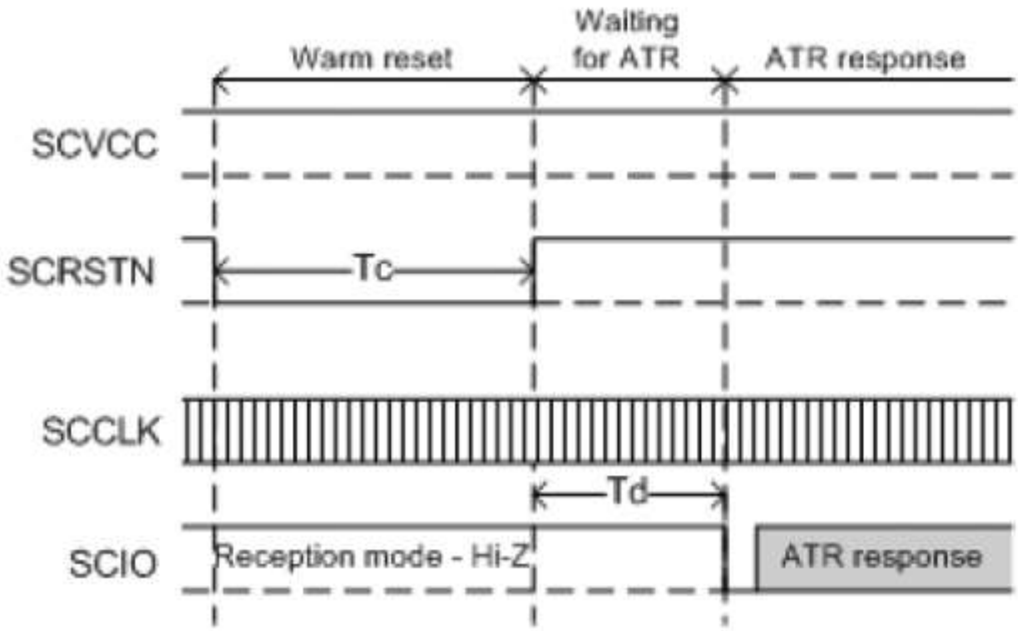


Fig. 24-3 Warm Reset and ATR

24.3.6 Deactivation of Contacts

After the smart card reader detects the removal of the smart card (SCREM interrupt) or the superior system initiates deactivation by setting the DEACT bit in the CTRL2 register to '1', the deactivation is performed immediately as shown in . The duration time (Ta), of each part of the deactivation sequence time is defined in the ADEATIME register.

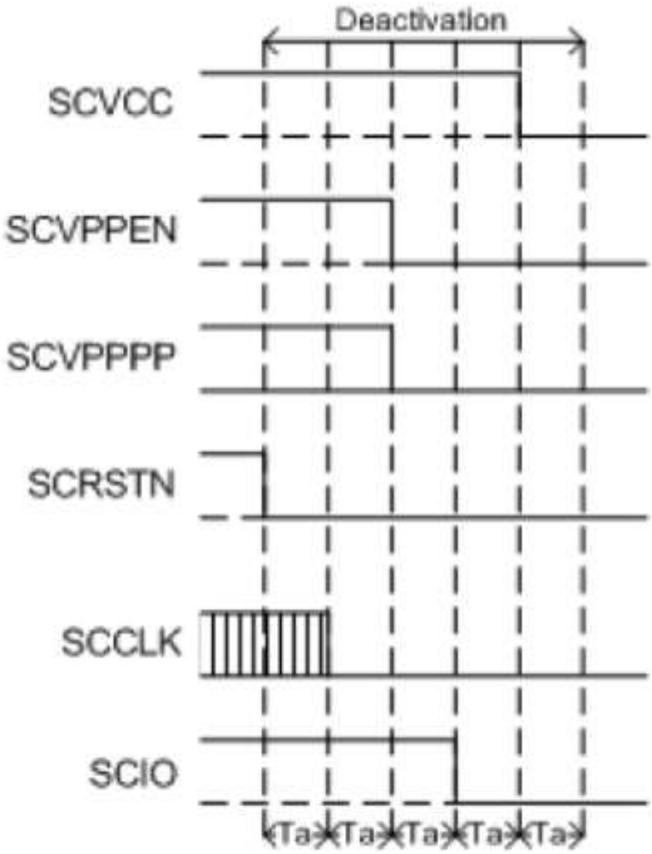


Fig. 24-4 Deactivation Sequence

24.4 Register Description

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SCR_CTRL1	0x0000	HW	0x0000	Control Register 1
SCR_CTRL2	0x0004	HW	0x0000	Control Register 2
SCR_SCPADS	0x0008	HW	0x0000	Smart Card Pads Register
SCR_INTEN1	0x000c	HW	0x0000	Interrupt Enable Register 1
SCR_INTSTAT1	0x0010	HW	0x0000	Interrupt Status Register 1
SCR_FIFOCTRL	0x0014	HW	0x0000	FIFO Control Register
SCR_LEGTXFICNT	0x0018	B	0x00	Legacy TX FIFO Counter
SCR_LEGRXFICNT	0x0019	B	0x00	Legacy RX FIFO Counter
SCR_RXFITH	0x001c	HW	0x0000	RX FIFO Threshold
SCR_REP	0x0020	B	0x00	Repeat
SCR_SCCDDIV	0x0024	HW	0x0000	Smart Card Clock Divisor
SCR_BAUDDIV	0x0028	HW	0x0000	Baud Clock Divisor
SCR_SCGUTIME	0x002c	B	0x00	Smart Card Guard-time
SCR_ADEATIME	0x0030	HW	0x0000	Activation / Deactivation Time
SCR_LOWRSTTIME	0x0034	HW	0x0000	Reset Duration
SCR_ATRSTARTLIMIT	0x0038	HW	0x0000	ATR Start Limit
SCR_C2CLIM	0x003c	HW	0x0000	Two Characters Delay Limit
SCR_INTEN2	0x0040	HW	0x0000	Interrupt Enable Register 2

Name	Offset	Size	Reset Value	Description
SCR_INTSTAT2	0x0044	HW	0x0000	Interrupt Status Register 2
SCR_TXFITH	0x0048	HW	0x0000	TX FIFO Threshold
SCR_TXFIFOCNT	0x004c	HW	0x0000	TX FIFO Counter
SCR_RXFIFOCNT	0x0050	HW	0x0000	RX FIFO Counter
SCR_BAUDTUNE	0x0054	B	0x00	Baud Tune Register
SCR_FIFO	0x0200	B	0x00	FIFO

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

24.4.2 Detail Register Description

SCR_CTRL1

Address: Operational Base + offset (0x0000)

Control Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	GINTEN Global Interrupt Enable When high, INTERRUPT output assertion is enabled.
14	RO	0x0	reserved
13	RW	0x0	TCKEN TCK enable When enabled all ATR bytes beginning from T0 are being XOR-ed. The result must be equal to TCK byte (when present). If the TCK byte does not match the computed value the ATR is considered to be malformed.
12	RW	0x0	ATRSTFLUSH ATR Start Flush FIFO When enabled, both FIFOs are flushed before the ATR is started.
11	RW	0x0	T0T1 T0/T1 Protocol Controls the using of T=0 or T=1 protocol. No character repeating is used when T=1 protocol is selected. The Character Guard-time (minimum delay between the leading edges of two consecutive characters) is reduced to 11 ETU when T=1 protocol is used and Guard-time value N = 255. The delay between the leading edge of the last received character and the leading edge of the first character transmitted is 16 ETU when T=0 protocol is used and 22 ETU when T=1 protocol is used.
10	RW	0x0	TS2FIFO TS to FIFO Enables to store the first ATR character TS in RX FIFO. During ideal card session there is no necessity to store TS character, so it can be disabled

Bit	Attr	Reset Value	Description
9	RW	0x0	RXEN Receiving enable When enabled the characters sent by the Smart Card are received by the UART and stored in RX FIFO. Receiving is internally disabled while a transmission is in progress.
8	RW	0x0	TXEN Transmission enable When enabled the characters are read from TX FIFO and transmitted through UART to the Smart Card
7	RW	0x0	CLKSTOPVAL Clock Stop Value The value of the scclk output during the clock stop state.
6	RW	0x0	CLKSTOP Clock Stop Clock Stop. When this bit is asserted and the smart card I/O line is in 'Z' state, the SCR core stops driving of the smart card clock signal after the CLKSTOPDELAY time expires. The smart card clock is restarted immediately after the CLKSTOP signal is de-asserted. New character transmission can be started by superior system after the CLKSTARTDELAY time expires. The expiration of both times is signaled by the CLKSTOPRUN bit in the Interrupt registers. Reading '1' from this bit signals that the clock is stopped or CLKSTARTDELAY time not expired yet. Reading '0' from this bit signals that the clock is not stopped.
5:3	RO	0x0	reserved
2	RW	0x0	PECH2FIFO Character With Wrong Parity to FIFO Enables storage of the characters received with wrong parity in RX FIFO.
1	RW	0x0	INVORD Inverse Bit Ordering When High, inverse bit ordering convention(MSB-LSB) is used.
0	RW	0x0	INVLEV Inverse Bit Level When high, inverse level convention is used(A= '1', Z='0');

SCR_CTRL2

Address: Operational Base + offset (0x0004)

Control Register 2

Bit	Attr	Reset Value	Description
15:8	RO	0x00	Reserved3 Reserved Reserved bits are hard-wired to zero

Bit	Attr	Reset Value	Description
7	RW	0x0	VCC50 Control 5V Smart Card Vcc Control 5V Smart Card Vcc. Setting of this bit allows selection of 5V Vcc for Smart Card session (Class A). After the selection of operating class is completed, this bit is in '1' if this class was selected. Default value after reset is '0'.
6	RW	0x0	VCC33 Control 3V Smart Card Vcc Setting of this bit allows selection of 3V Vcc for Smart Card session (Class B). After the selection of operating class is completed, this bit is in '1' if this class was selected. Default value after reset is '0'.
5	RW	0x0	VCC18 Control 1.8V Smart Card Vcc Control 1.8V Smart Card Vcc. Setting of this bit allows selection of 1.8V Vcc for Smart Card session (Class C). After the selection of operating class is completed, this bit is in '1' if this class was selected. Default value after reset is '0'.
4	RW	0x0	DEACT Deactivation Setting of this bit initializes the deactivation sequence. When the deactivation is finished, the DEACT bit is automatically cleared.
3	RW	0x0	ACT Activation Setting of this bit initializes the activation sequence. When the activation is finished, the ACT bit is automatically cleared.
2	WO	0x0	WARMRST Warm Reset Command Writing '1' to this bit initializes Warm Reset of the Smart Card. This bit is always read as '0'.
1:0	RO	0x0	reserved

SCR_SCPADS

Address: Operational Base + offset (0x0008)

Smart Card Pads Register

Bit	Attr	Reset Value	Description
15:10	RO	0x0	reserved
9	RO	0x0	SCPRESENT Smart Card presented This bit is set to '1' when the SCDETECT input is active at least for SCDETECTTIME
8	RW	0x0	DSCFCB Direct Smart Card Function Code Bit It provides direct access to SCFCB output

Bit	Attr	Reset Value	Description
7	RW	0x0	DSCVPPPP Direct Smart Card Vpp Pause/Prog It provides direct access to SCVPPPP output
6	RW	0x0	DSCVPPEN Direct Smart Card Vpp Enable It provides direct access to SCVPPEN output
5	RW	0x0	AUTOADEAVPP Automatic Vpp Handling. When high, it enables automatic handling of DSCVPPEN and DSCVPPPP signals during activation and deactivation sequence.
4	RW	0x0	DSCVCC Direct Smart Card Vcc Direct Smart Card Vcc. When DIRACCPADS = '1', the DSCVCC bit provides direct access to SCVCCx outputs. The appropriate SCVCC18, SCVCC33 and SCVCC50 outputs are driven according to state of bits VCC18, VCC33 and VCC50 in CTRL2 register.
3	RW	0x0	DSCRST Direct Smart Card Reset When DIRACCPADS = '1', the DSCRST bit provides direct access to SCRST output
2	RW	0x0	DSCCLK Direct Smart Card Clock When DIRACCPADS = '1', the DSCCLK bit provides direct access to SCCLK output
1	RW	0x0	DSCIO Direct Smart Card Input/Output When DIRACCPADS = '1', the DSCIO bit provides direct access to SCIO pad.
0	RW	0x0	DIRACCPADS Direct Access To Smart Card Pads When high, it disables a serial interface functionality and enables direct control of the smart card pads using following 4 bits.

SCR_INTEN1

Address: Operational Base + offset (0x000c)

Interrupt Enable Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	SCDEACT Smart Card Deactivation Interrupt When enabled, this interrupt is asserted after the Smart Card deactivation sequence is complete.

Bit	Attr	Reset Value	Description
14	RW	0x0	SCACT Smart Card Activation Interrupt. When enabled, this interrupt is asserted after the Smart Card activation sequence is complete.
13	RW	0x0	SCINS Smart Card Inserted Interrupt When enabled, this interrupt is asserted after the smart card insertion
12	RW	0x0	SCREM Smart Card Removed Interrupt. When enabled, this interrupt is asserted after the smart card removal.
11	RW	0x0	ATRDONE ATR Done Interrupt When enabled, this interrupt is asserted after the ATR sequence is successfully completed.
10	RW	0x0	ATRFAIL ATR Fail Interrupt When enabled, this interrupt is asserted if the ATR sequence fails.
9	RW	0x0	RXTHRESHOLD RX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in RX FIFO is equal or exceeds the RX FIFO threshold.
8	RW	0x0	C2CFULL Two Consecutive Characters Limit Interrupt When enabled, this interrupt is asserted if the time between two consecutive characters, transmitted between the Smart Card and the Reader in both directions, is equal the Two Characters Delay Limit described below. The C2CFULL interrupt is internally enabled from the ATR start to the deactivation or ATR restart initialization. It is recommended to use this counter to detect unresponsive Smart Cards.
7	RW	0x0	RXPERR Reception Parity Error Interrupt When enabled, this interrupt is asserted after the character with wrong parity was received when the number of repeated receptions exceeds RXREPEAT value or T=1 protocol is used
6	RW	0x0	TXPERR Transmission Parity Error Interrupt. When enabled, this interrupt is asserted if the Smart Card signals wrong character parity during the guard-time after the character transmission was repeated TXREPEAT-times

Bit	Attr	Reset Value	Description
5	RW	0x0	RXDONE Reception Done Interrupt When enabled, this interrupt is asserted after a character was received from the Smart Card.
4	RW	0x0	TXDONE Transmission Done Interrupt When enabled, this interrupt is asserted after one character was transmitted to the Smart Card.
3	RW	0x0	CLKSTOPRUN Smart Card Clock Stop Interrupt When enabled, this interrupt is asserted in two cases: 1. When the smart card clock is stopped (after CLOCKSTOP assertion). 2. When the new character transfer can be started (the smart card clock is fully running after CLOCKSTOP de-assertion).
2	RW	0x0	RXFIFULL RX FIFO Full Interrupt When enabled, this interrupt is asserted if the RX FIFO is filled up.
1	RW	0x0	TXFIEMPTY TX FIFO Empty Interrupt. When enabled, this interrupt is asserted if the TX FIFO is emptied out.
0	RW	0x0	TXFIDONE TX FIFO Done Interrupt When enabled, this interrupt is asserted after all bytes from TX FIFO were transferred to the Smart Card

SCR_INTSTAT1

Address: Operational Base + offset (0x0010)

Interrupt Status Register 1

Bit	Attr	Reset Value	Description
15	RW	0x0	SCDEACT Smart Card Deactivation Interrupt When enabled, this interrupt is asserted after the Smart Card deactivation sequence is complete.
14	RW	0x0	SCACT Smart Card Activation Interrupt. When enabled, this interrupt is asserted after the Smart Card activation sequence is complete.
13	RW	0x0	SCINS Smart Card Inserted Interrupt When enabled, this interrupt is asserted after the smart card insertion

Bit	Attr	Reset Value	Description
12	RW	0x0	SCREM Smart Card Removed Interrupt. When enabled, this interrupt is asserted after the smart card removal.
11	RW	0x0	ATRDONE ATR Done Interrupt When enabled, this interrupt is asserted after the ATR sequence is successfully completed.
10	RW	0x0	ATRFAIL ATR Fail Interrupt When enabled, this interrupt is asserted if the ATR sequence fails.
9	RW	0x0	RXTHRESHOLD RX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in RX FIFO is equal or exceeds the RX FIFO threshold.
8	RW	0x0	C2CFULL Two Consecutive Characters Limit Interrupt When enabled, this interrupt is asserted if the time between two consecutive characters, transmitted between the Smart Card and the Reader in both directions, is equal the Two Characters Delay Limit described below. The C2CFULL interrupt is internally enabled from the ATR start to the deactivation or ATR restart initialization. It is recommended to use this counter to detect unresponsive Smart Cards.
7	RW	0x0	RXPERR Reception Parity Error Interrupt When enabled, this interrupt is asserted after the character with wrong parity was received when the number of repeated receptions exceeds RXREPEAT value or T=1 protocol is used
6	RW	0x0	TXPERR Transmission Parity Error Interrupt. When enabled, this interrupt is asserted if the Smart Card signals wrong character parity during the guard-time after the character transmission was repeated TXREPEAT-times
5	RW	0x0	RXDONE Reception Done Interrupt When enabled, this interrupt is asserted after a character was received from the Smart Card.
4	RW	0x0	TXDONE Transmission Done Interrupt When enabled, this interrupt is asserted after one character was transmitted to the Smart Card.

Bit	Attr	Reset Value	Description
3	RW	0x0	CLKSTOPRUN Smart Card Clock Stop Interrupt When enabled, this interrupt is asserted in two cases: 1. When the smart card clock is stopped (after CLOCKSTOP assertion). 2. When the new character transfer can be started (the smart card clock is fully running after CLOCKSTOP de-assertion).
2	RW	0x0	RXFIFULL RX FIFO Full Interrupt When enabled, this interrupt is asserted if the RX FIFO is filled up.
1	RW	0x0	TXFIEMPTY TX FIFO Empty Interrupt. When enabled, this interrupt is asserted if the TX FIFO is emptied out.
0	RW	0x0	TXFIDONE TX FIFO Done Interrupt When enabled, this interrupt is asserted after all bytes from TX FIFO were transferred to the Smart Card

SCR_FIFOCTRL

Address: Operational Base + offset (0x0014)

FIFO Control Register

Bit	Attr	Reset Value	Description
15:11	RO	0x0	reserved
10	WO	0x0	RXFIFLUSH Flush RX FIFO RX FIFO is flushed, when '1' is written to this bit.
9	RO	0x0	RXFIFULL RX FIFO Full RX FIFO Full
8	RO	0x0	RXFIEMPTY RX FIFO Empty RX FIFO Empty
7:3	RO	0x0	reserved
2	WO	0x0	TXFIFLUSH Flush TX FIFO. TX FIFO is flushed, when '1' is written to this bit.
1	RO	0x0	TXFIFULL TX FIFO Full TX FIFO Full
0	RO	0x0	TXFIEMPTY TX FIFO Empty. TX FIFO Empty.

SCR_LEGTXFICNT

Address: Operational Base + offset (0x0018)

Legacy TX FIFO Counter

Bit	Attr	Reset Value	Description
7:0	RO	0x00	LEGTXFICNT Legacy TX FIFO Counter It is equal to TX FIFO Counter up to value 255. All values above 255 are read as 255. It is recommended to use the 16-bit TX FIFO Counter instead of this register.

SCR_LEGRXFICNT

Address: Operational Base + offset (0x0019)

Legacy RX FIFO Counter

Bit	Attr	Reset Value	Description
7:0	RO	0x00	LEGRXFICNT Legacy RX FIFO Counter It is equal to RX FIFO Counter up to value 255. All values above 255 are read as 255. It is recommended to use the 16-bit RX FIFO Counter instead of this register.

SCR_RXFITH

Address: Operational Base + offset (0x001c)

RX FIFO Threshold

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	RXFITH RX FIFO Threshold The interrupt is asserted when the number of bytes it receives is equal to, or exceeds the threshold

SCR_REP

Address: Operational Base + offset (0x0020)

Repeat

Bit	Attr	Reset Value	Description
7:4	RW	0x0	RXREP RX Repeat This is a 4-bit, read/write register that specifies the number of attempts to request character re-transmission after wrong parity was detected. The re-transmission of the character is requested using the 1 ETU long error signal during the guard-time

Bit	Attr	Reset Value	Description
3:0	RW	0x0	TXREP TX Repeat This is a 4-bit, read/write register that specifies the number of attempts to re-transmit the character after the Smart Card signals the wrong parity during the guard-time.

SCR_SCCDDIV

Address: Operational Base + offset (0x0024)

Smart Card Clock Divisor

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	SCCDDIV Smart Card Clock Divisor This is a 16-bit, read/write register that defines the divisor value used to generate the Smart Card Clock from the system clock.

SCR_BAUDDIV

Address: Operational Base + offset (0x0028)

Baud Clock Divisor

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	BAUDDIV Baud Clock Divisor This is a 16-bit, read/write register that defines a divisor value used to generate the Baud Clock impulses from the system clock

SCR_SCGUTIME

Address: Operational Base + offset (0x002c)

Smart Card Guard-time

Bit	Attr	Reset Value	Description
7:0	RW	0x00	SCGUTI Smart Card Guard-time This is an 8-bit, read/write register that sets a delay at the end of each character transmitted from the Smart Card Reader to the Smart Card. The value is in Elementary Time Units (ETU). The parity error is besides signaled during the guardtime

SCR_ADEATIME

Address: Operational Base + offset (0x0030)

Activation / Deactivation Time

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ADEATIME Activation / Deactivation Time Sets the duration of each part of the activation and deactivation sequence. The value is in Smart Card Clock Cycles.
7:0	RW	0x00	Reserved Reserved Reserved bits are hard-wired to zero.

SCR_LOWRSTTIME

Address: Operational Base + offset (0x0034)

Reset Duration

Bit	Attr	Reset Value	Description
15:8	RW	0x00	LOWRSTTIME Reset Duration Sets the duration of the smart card reset sequence. This value is same for the cold and warm reset. The value is in terms of smart card clock cycles.
7:0	RW	0x00	Reserved Reserved Bits (7:0) of this register are hard-wired to zero.

SCR_ATRSTARTLIMIT

Address: Operational Base + offset (0x0038)

ATR Start Limit

Bit	Attr	Reset Value	Description
15:8	RW	0x00	ATRSTARTLIMIT ATR Start Limit Defines the maximum time between the rising edge of the SCRSTN signal and the start of ATR response. The value is in terms of smart card clock cycles
7:0	RW	0x00	Reserved Reserved Bits (7:0) of this register are hard-wired to zero

SCR_C2CLIM

Address: Operational Base + offset (0x003c)

Two Characters Delay Limit

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	C2CLIM Two Characters Delay Limit This is a 16-bit, read/write register that sets the maximum time between the leading edges of two, consecutive characters. The value is in ETUs.

SCR_INTEN2
Address: Operational Base + offset (0x0040)
Interrupt Enable Register 2

Bit	Attr	Reset Value	Description
15:2	RO	0x0	reserved
1	RW	0x0	TCKERR TCK Error Interrupt. When enabled, this interrupt is asserted if the TCK byte does not match computed value.
0	RW	0x0	TXTHRESHOLD TX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in TX FIFO is equal or less than the TX FIFO threshold.

SCR_INTSTAT2
Address: Operational Base + offset (0x0044)
Interrupt Status Register 2

Bit	Attr	Reset Value	Description
15:2	RO	0x0	reserved
1	RW	0x0	TCKERR TCK Error Interrupt When enabled, this interrupt is asserted if the TCK byte does not match computed value.
0	RW	0x0	TXTHRESHOLD TX FIFO Threshold Interrupt When enabled, this interrupt is asserted if the number of bytes in TX FIFO is equal or less than the TX FIFO threshold.

SCR_TXFITH
Address: Operational Base + offset (0x0048)
TX FIFO Threshold

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	TXFITH TX FIFO Threshold The interrupt is asserted when the number of bytes in TX FIFO is equal or less than the threshold

SCR_TXFIFOCNT

Address: Operational Base + offset (0x004c)

TX FIFO Counter

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	TXFIFOCNT TX FIFO Counter This is a 16-bit, read-only register that provides the number of bytes stored in the RX FIFO

SCR_RXFIFOCNT

Address: Operational Base + offset (0x0050)

RX FIFO Counter

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	RXFIFOCNT RX FIFO Counter This is a 16-bit, read-only register that provides the number of bytes stored in the RX FIFO.

SCR_BAUDTUNE

Address: Operational Base + offset (0x0054)

Baud Tune Register

Bit	Attr	Reset Value	Description
7:4	RO	0x0	reserved
3:0	RW	0x0	BAUDTUNE Baud Tune Register This is a 3-bit, read/write register that defines an additional value used to increase the accuracy of the Baud Clock impulses

SCR_FIFO

Address: Operational Base + offset (0x0200)

FIFO

Bit	Attr	Reset Value	Description
7:0	RW	0x00	FIFO FIFO This is an 8-bit, read/write register that provides access to the receive and transmit FIFO buffers. The TX FIFO is accessed during the APB write transfer. The RX FIFO is accessed during the APB read transfer. All read/write accesses at address range 200h-3ffh are redirected to the FIFO.

24.5 Interface Description

Table 24-1 SCR Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
sc_clk	O	IO_CARDclkm0_GPIO3B4vccio6	GPIO3B_IOMUX[9:8]=01 GRF_CON_IOMUX[7]=0
		IO_I2S1sdi_PWMsdi0m0_CARDclkm1_GPIO2C3vccio5	GPIO2CL_IOMUX[11:9]=011 GRF_CON_IOMUX[7]=1
sc_rst	O	IO_CARDrstm0_GPIO3B5vccio6	GPIO3B_IOMUX[11:10]=01 GRF_CON_IOMUX[7]=0
		IO_I2S1sdio1_PDMsdi1m0_CARDrstm1_GPIO2C4vccio5	GPIO2CL_IOMUX[14:12]=011 GRF_CON_IOMUX[7]=1
sc_detect	I	IO_CARDdetm0_GPIO3B6vccio6	GPIO3B_IOMUX[13:12]=01 GRF_CON_IOMUX[7]=0
		IO_I2S1sdio2_PDMsdi2m0_CARDdetm1_GPIO2C5vccio5	GPIO2CH_IOMUX[2:0]=011 GRF_CON_IOMUX[7]=1
sc_io	I	IO_CARDiom0_GPIO3B7vccio6	GPIO3B_IOMUX[15:14]=01 GRF_CON_IOMUX[7]=0
		IO_I2S1sdio3_PDMsdi3m0_CARDiom1_GPIO2C6vccio5	GPIO2CH_IOMUX[5:3]=011 GRF_CON_IOMUX[7]=1

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

24.6 Application Notes

24.6.1 BCHST/BCHLOC/BCHDE/SPARE Application

The Smart Card Clock signal is used as the main clock for the smart card. Its frequency canbeadjusted using the Smart Card Clock Divisor (SCCDIV). This value is used to divide the system clock.
The SCCLK frequency is given by the following equation:

$$\mathrm{SCCLK_{freq}} = \frac{\mathrm{CLK\_freq}}{2 * (\mathrm{SCCDIV} + 1)}, \qquad \mathrm{SCCDIV} \cong \frac{\mathrm{CLK_{freq}}}{2 * \mathrm{SCCLK_{freq}}} - 1$$

SCCLK_freq– Smart Card Clock Frequency
CLK_freq- System Clock Frequency
The Baud Clock Impulse signal is used to transmit and receive serial data between the Smart CardReader and the Smart Card. The baud rate can be modified using the Baud Clock Divisor (BAUDDIV)which is used to divide the system clock. The BAUDDIV value must be >= 4. The BAUD rate is given by the following equation:

$$\mathrm{BAUD_{rate}} = \frac{\mathrm{CLK\_freq}}{2 * (\mathrm{BAUDDIV} + 1)}$$

The duration of one bit, Elementary Time Unit (ETU) and parameters F and D are defined in the ISO/IEC7816-3 specification.

$$\frac{1}{\mathrm{BAUD\_rate}} \cong \mathrm{ETU} = \frac{\mathrm{F}}{\mathrm{D}} * \frac{1}{\mathrm{SCCLK_{freq}}}, \frac{\mathrm{F}}{\mathrm{D}} \cong \frac{\mathrm{BAUDDIV} + 1}{\mathrm{SCCDIV} + 1}$$

BAUDDIV equation based on SCCDIV value and Smart Card parameters F and D is following:

$$\mathrm{BAUDDIV} \cong (\mathrm{SCCDIV} + 1) * \frac{\mathrm{F}}{\mathrm{D}} - 1$$

During the first answer to reset response after the cold reset, the initial ETU must be equal to 372 SmartCard Clock Cycles (given by parameters F=372 and D=1). In this case, the BAUDDIV should be:

$$\mathrm{BAUDDIV} \cong (\mathrm{SCCDIV} + 1) * \frac{372}{1} - 1$$

After the ATR is completed, the BAUDDIV register value can be changed according to Smart Cardparameters F and D.

Baud Tune Register (BAUDTUNE) 3-bit value that can be used to increase the accuracy of the BaudClock impulses timing by using the BAUDTUNE Increment from Table listed below in combination with BAUDDIVregister value.

Table 24-2 BAUDTUNE register

BAUDTUNE	000	001	010	011	100	101	110	111
$\mathrm{BAUDTUNE}_{\mathrm{INCR}}$	+0	+0.125	0.25	+0.375	+0.5	+0.625	+0.75	+0.875

$$\mathrm{BAUDDIV} + \mathrm{BAUDTUNE}_{\mathrm{INCR}} \cong (\mathrm{SCCDIV} + 1) * \frac{\mathrm{F}}{\mathrm{D}} - 1$$

The BAUDDIV register value (nearest integer) can be computed using following equation:

$$\mathrm{BAUDDIV} \cong (\mathrm{SCCDIV} + 1) * \frac{\mathrm{F}}{\mathrm{D}} - 1 - \mathrm{BAUDTUNE}_{\mathrm{INCR}}$$

24.6.2 Smart Card Detect Application

It is configurable for SCR’s detect pin when Smart Card is inserted.
When config GRF_SOC_CON7[0]=0, SCDETECT`s active state is 0.
When config GRF_SOC_CON7[0]=1, SCDETECT`s active state is 1.

Chapter 25 I2S/PCM Controller

25.1 Overview

The I2S/PCM controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

Not only I2S but also PCM mode surround audio output and stereo input are supported in I2S/PCM controller.

There are three I2S/PCM controllers embedded in the design, I2S0, I2S1 and I2S2. Different features between I2S/PCM controllers are as follows.

- Support four internal 32-bit wide and 32-location deep FIFOs for transmitting audio data for I2S0
- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and four for receiving audio data for I2S1
- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and one for receiving audio data for I2S2
- Support 8 channels audio data transmitting in I2S mode for I2S0, 8 channels audio data transmitting or 8 channels audio data receiving for I2S1, 2 channels audio data transmitting and 2 channels audio data receiving for I2S2.

Common features for I2S0, I2S1 and I2S2 are as follows.

- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2 channels audio receiving in PCM mode
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and one for transmitting audio data. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity

25.2 Block Diagram

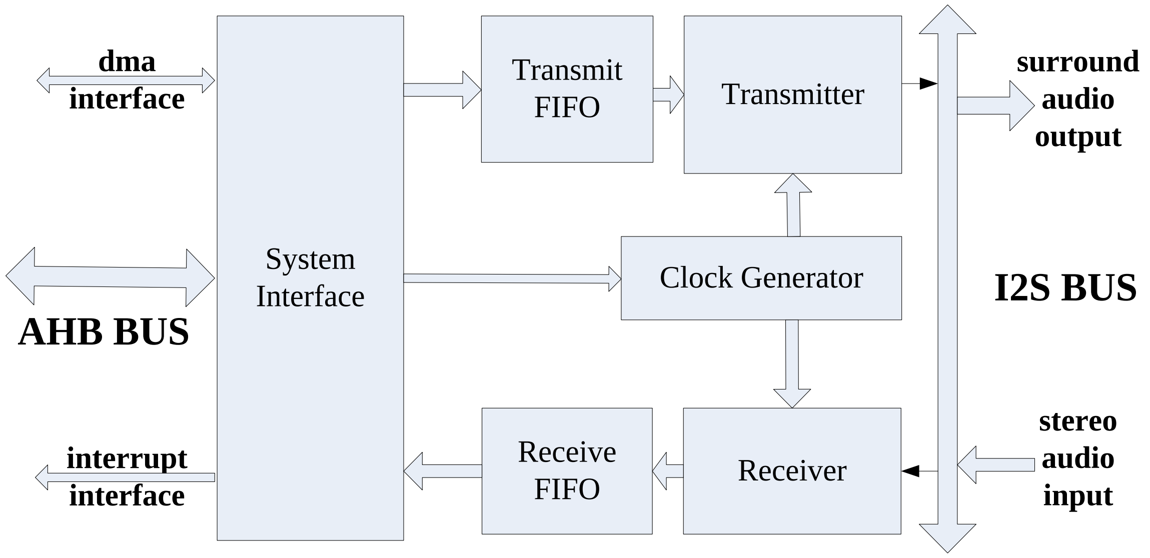


Fig. 25-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitter and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitter

The Transmitter implements transmission operation. The transmitter can act as either master or slave, with I2S or PCM mode surround serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

25.3 Function description

In the I2S/PCM controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

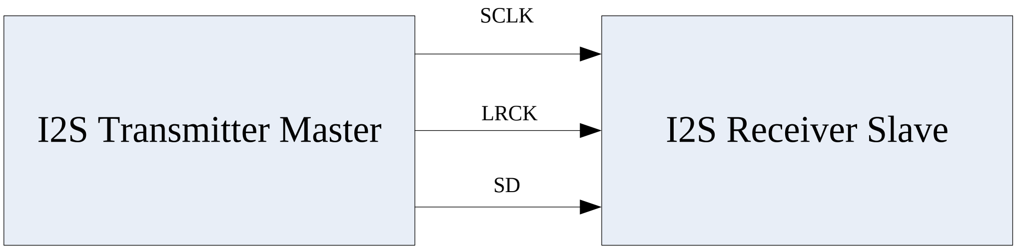


Fig. 25-2 I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

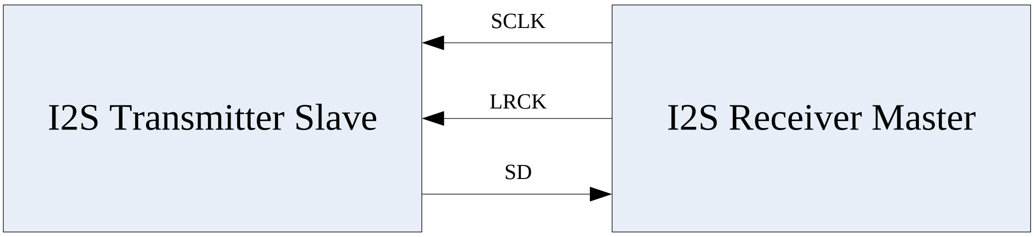


Fig. 25-3 I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then the receiver start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

25.3.1 i2s normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo,i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

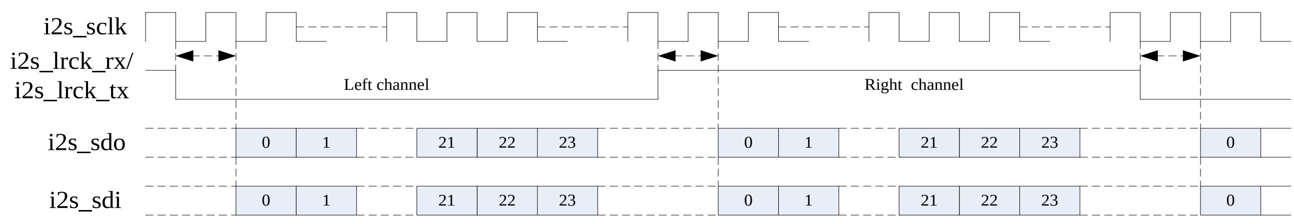


Fig. 25-4 I2S normal mode timing format

25.3.2 i2s left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

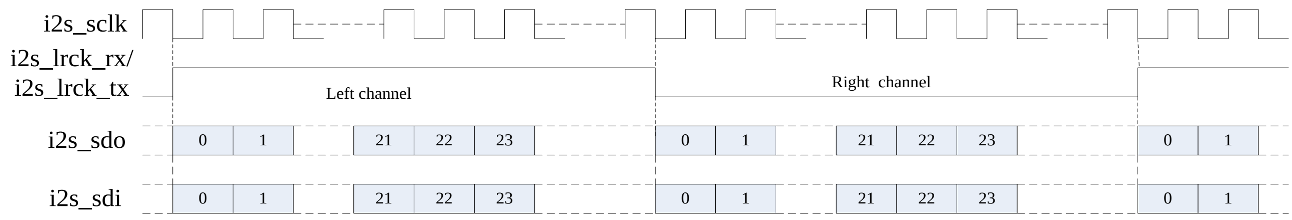


Fig. 25-5 I2S left justified mode timing format

25.3.3 i2s right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

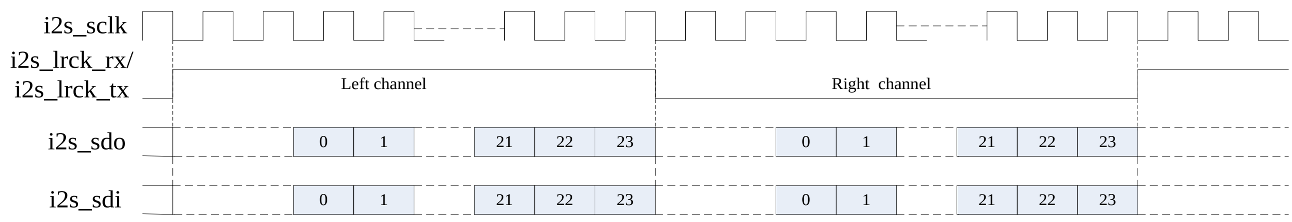


Fig. 25-6 I2S right justified mode timing format

25.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

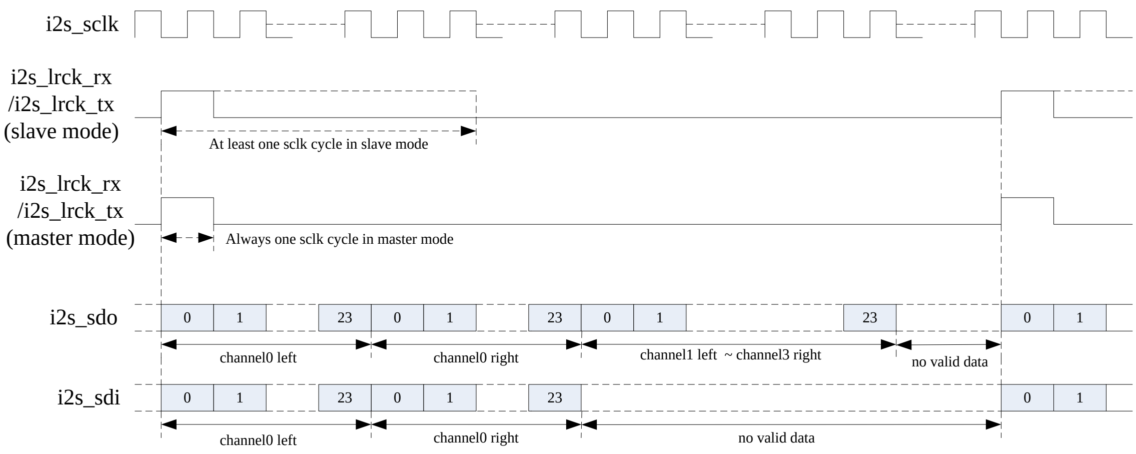


Fig. 25-7 PCM early mode timing format

25.3.5 PCM late1 mode

This is the waveform of PCM late1 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

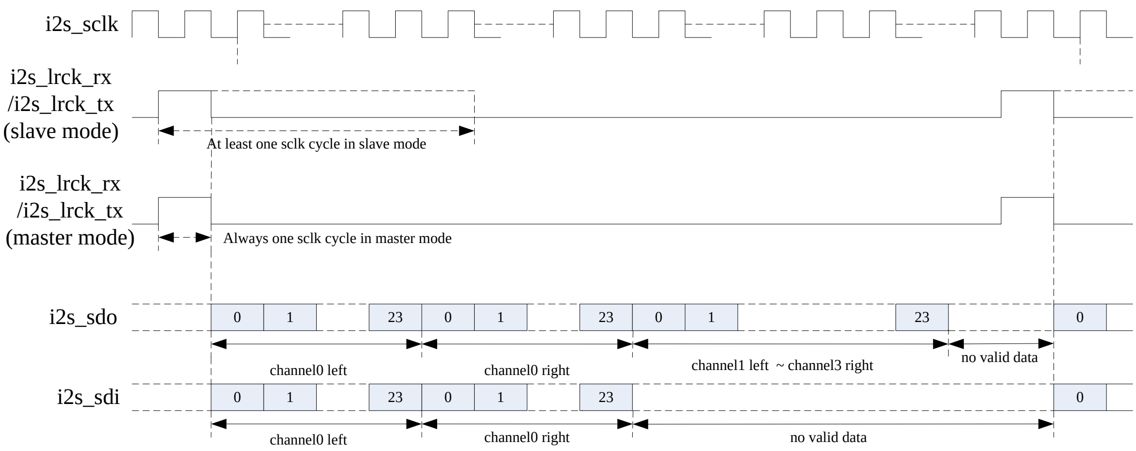


Fig. 25-8 PCM late1 mode timing format

25.3.6 PCM late2 mode

This is the waveform of PCM late2 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

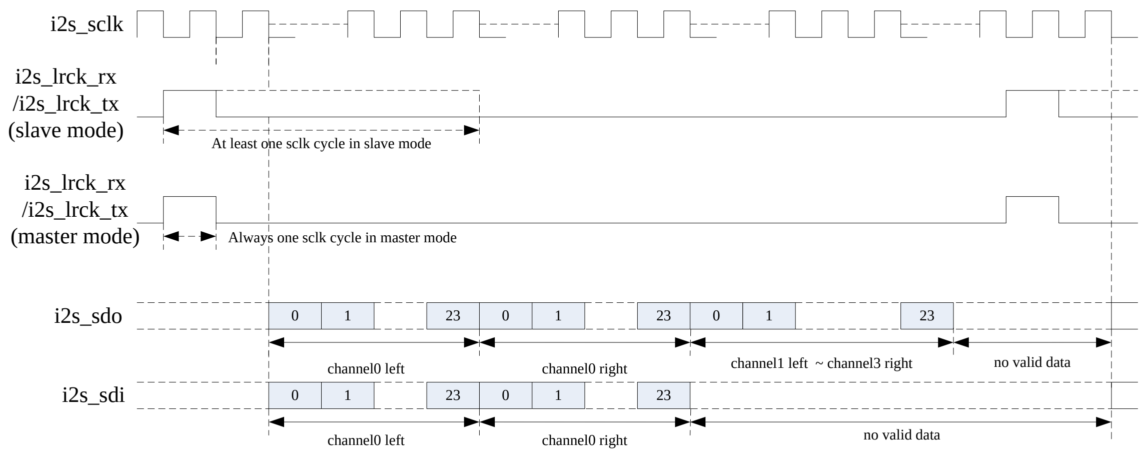


Fig. 25-9 PCM late2 mode timing format

25.3.7 PCM late3 mode

This is the waveform of PCM late3 mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

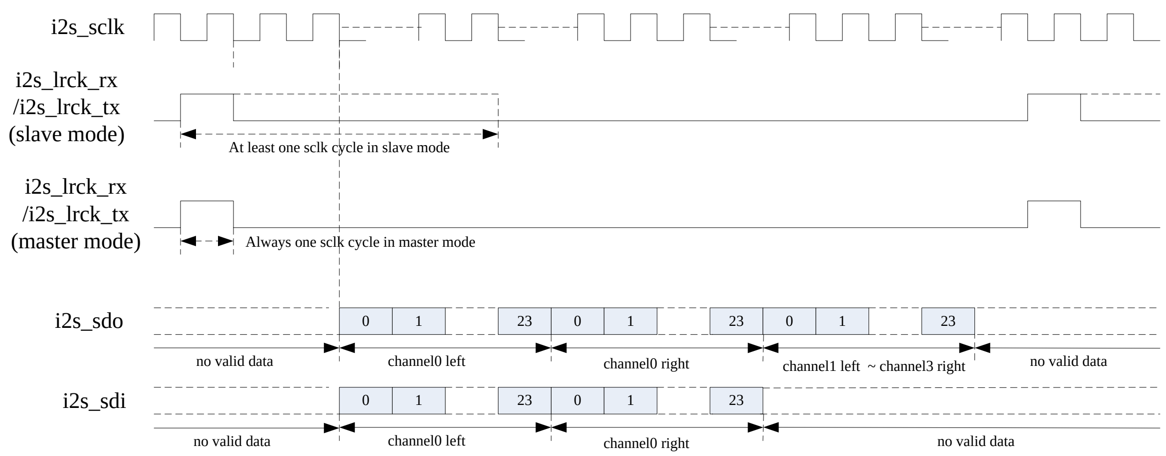


Fig. 25-10 PCM late3 mode timing format

25.4 Register Description

This section describes the control/status registers of the design.

25.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2S_RXCR	0x0004	W	0x0000000f	receive operation control register
I2S_CKR	0x0008	W	0x00071f1f	clock generation register
I2S_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x00000000	interrupt control register

Name	Offset	Size	Reset Value	Description
I2S_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register
I2S_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register
I2S_VERSION	0x0030	W	0x20150001	I2s version

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

25.4.2 Detail Register Description

I2S_TXCR

Address: Operational Base + offset (0x0000)

transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right justified counter (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	TCSR TX 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[0] bit is 0.) Only valid when VDW select 16bit data. 0:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	SJM Store justified mode SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. This bit is invalid if VDW select 16bit data and HWT select 0, Because every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)

receive operation control register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16:15	RW	0x0	RCSR RX 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. 0:32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1:low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_CKR

Address: Operational Base + offset (0x0008)

clock generation register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM 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.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk

Bit	Attr	Reset Value	Description
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
23:16	RW	0x07	MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclk frequecy / txsclk frequecy-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; …… 2n,2n+1:Fmclk=(2n+2)*Ftxsclk; …… 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; …… 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… n: (n+1)fs …… 253: 254fs 254: 255fs 255: 256fs

I2S_TXFIFOLR

Address: Operational Base + offset (0x000c)

TX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.

Bit	Attr	Reset Value	Description
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_DMACR

Address: Operational Base + offset (0x0010)

DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : 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 (RXFIFO0 if RCSR=00;RXFIFO1 if RCSR=01,RXFIFO2 if RCSR=10,RXFIFO3 if RCSR=11)is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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 TXFIFO(TXFIFO0 if TCSR=00;TXFIFO1 if TCSR=01,TXFIFO2 if TCSR=10,TXFIFO3 if TCSR=11)is equal to or below this field value.

I2S_INTCR

Address: Operational Base + offset (0x0014)

interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries (RXFIFO0 if RCSR=00; RXFIFO1 if RCSR=01, RXFIFO2 if RCSR=10, RXFIFO3 if RCSR=11) 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 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if TCSR=00; TXFIFO1 if TCSR=01, TXFIFO2 if TCSR=10, TXFIFO3 if TCSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2S_INTSR

Address: Operational Base + offset (0x0018)

interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2S_XFER

Address: Operational Base + offset (0x001c)

Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2S_CLR

Address: Operational Base + offset (0x0020)

SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	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_TXDR

Address: Operational Base + offset (0x0024)

Transmit FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR

Address: Operational Base + offset (0x0028)

Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

I2S_RXFIFOLR

Address: Operational Base + offset (0x002c)

RX FIFO level register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RO	0x00	RFL3 Receive FIFO3 Level Contains the number of valid data entries in the receive FIFO3.
17:12	RO	0x00	RFL2 Receive FIFO2 Level Contains the number of valid data entries in the receive FIFO2.
11:6	RU	0x00	RFL1 Receive FIFO1 Level Contains the number of valid data entries in the receive FIFO1.
5:0	RO	0x00	RFL0 Receive FIFO0 Level Contains the number of valid data entries in the receive FIFO0.

I2S_VERSION

Address: Operational Base + offset (0x0030)

I2S version

Bit	Attr	Reset Value	Description
31:0	RO	0x20150001	I2S version

25.5 16.5 Interface description

Table 25-1 I2S Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
Interface for i2s1			

Module Pin	Direction	Pad Name	IOMUX Setting
i2s1_mclk	I/O	IO_I2S1mclk_Nouse0_TSPd0m1_CIFdata7m1_GPIO2C2vccio5	GRF_GPIO2BH_IOMUX[8:6]=3’b001
i2s1_sclk	I/O	IO_I2S1sclk_PDMclkm0_TSPd7m1_CIFdata7m1_GPIO2C2vccio5	GRF_GPIO2CL_IOMUX[8:6]=3’b001
i2s1_lrck_rx	I/O	IO_I2S1lrckrx_NOuse1_TSPd5m1_CIFdata5m1_GPIO2C0vccio5	GRF_GPIO2CL_IOMUX[2:0]=3’b001
i2s1_lrck_tx	I/O	IO_I2S1lrcktx_SPDIFtxm1_TSPd6m1_CIFdata6m1_GPIO2C1vccio5	GRF_GPIO2CL_IOMUX[5:3]=3’b001
i2s1_sdo0	O	IO_I2S1sdo_PDMfsyncm0_GPIO2C7vccio5	GRF_GPIO2CH_IOMUX[7:6]=2’b01
i2s1_sdo1	O	IO_I2S1sdio1_PDMsdi1m0_CARDrstm1_GPIO2C4vccio5	GRF_GPIO2CL_IOMUX[14:12]=3’b001
i2s1_sdo2	O	IO_I2S1sdio2_PDMsdi2m0_CARDdetm1_GPIO2C5vccio5	GRF_GPIO2CH_IOMUX[2:0]=3’b001
i2s1_sdo3	O	IO_I2S1sdio3_PDMsdi3m0_CARDiom1_GPIO2C6vccio5	GRF_GPIO2CH_IOMUX[5:3]=3’b001
i2s1_sdi0	I	IO_I2S1sdi_PDMsdi0m0_CARDclkm1_GPIO2C3vccio5	GRF_GPIO2CL_IOMUX[11:9]=3’b001
i2s1_sdi1	I	IO_I2S1sdio1_PDMsdi1m0_CARDrstm1_GPIO2C4vccio5	GRF_GPIO2CL_IOMUX[14:12]=3’b001
i2s1_sdi2	I	IO_I2S1sdio2_PDMsdi2m0_CARDdetm1_GPIO2C5vccio5	GRF_GPIO2CH_IOMUX[2:0]=3’b001
i2s1_sdi3	I	IO_I2S1sdio3_PDMsdi3m0_CARDiom1_GPIO2C6vccio5	GRF_GPIO2CH_IOMUX[5:3]=3’b001
Interface for i2s2 M0 IO			
i2s2_mclk	I/O	IO_I2S2mclk_GMACclkm1_GPIO1C5vccio4	GRF_GPIO1C_IOMUX[11:10]=2’b01
i2s2_sclk	I/O	IO_I2S2sclkm0_GMACrxdvm1_PDMclkm1_GPIO1C6vccio4	GRF_GPIO1C_IOMUX[13:12]=2’b01

Module Pin	Direction	Pad Name	IOMUX Setting
i2s2_lrck_tx	I/O	IO_I2S2lrcktxm0_GMACmdcm1_PDMsdi0m1_GPIO1C7vccio4	GRF_GPIO1C_IOMUX[15:14]=2'b01
i2s2_lrck_rx	I/O	IO_I2S2lrckrxm0_CLKout_gmacm2_PDMsdi3m1_GPIO1D2vccio4	GRF_GPIO1D_IOMUX[5:4]=2'b01
i2s2_sdi	I	IO_I2S2sdim0_GMACrxerm1_PDMsdi1m1_GPIO1D0vccio4	GRF_GPIO1D_IOMUX[1:0]=2'b01
i2s2_sdo	O	IO_I2S2sdom0_GMACtxenm1_PDMsdi2m1_GPIO1D1vccio4	GRF_GPIO1D_IOMUX[3:2]=2'b01
Interface for i2s2 M1 IO			
i2s2_sclk	I/O	IO_TSPvalid_CIFvsync_SDMMC0EXTcmd_SPIclkm2_USB3PHYdebug1_I2S2sclkm1_GPIO3A0vccio6	GRF_GPIO3AL_IOMUX[2:0]=3'b110
i2s2_lrck_tx	I/O	IO_TSPd4_CIFdata4_SPIcsn0m2_I2S2lrcktxm1_USB3PHYdebug8_I2S2lrckrxm1_GPIO3B0vccio6	GRF_GPIO3BL_IOMUX[2:0]=3'b100
i2s2_lrck_rx	I/O	IO_TSPd4_CIFdata4_SPIcsn0m2_I2S2lrcktxm1_USB3PHYdebug8_I2S2lrckrxm1_GPIO3B0vccio6	GRF_GPIO3BL_IOMUX[2:0]=3'b110
i2s2_sdi	I	IO_TSPclk_CIFclkin_SDMMC0EXTclkout_SPIrxdm2_USB3PHYdebug3_I2S2sdim1_GPIO3A2vccio6	GRF_GPIO3AL_IOMUX[8:6]=3'b110
i2s2_sdo	O	IO_TSPfail_CIFhref_SDMMC0EXTdet_SPItxdm2_USB3PHYdebug2_I2S2sdom1_GPIO3A1vccio6	GRF_GPIO3AL_IOMUX[5:3]=3'b110

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

The i2s1_sdix(x=1,2,3) and i2s1_sdox(x=1,2,3) signals shares the same IO, the direction is configured by setting GRF_CON_CON10 [4:2]. Each bit controls the direction of IO_I2S1sdio1_PDMsdi1m0_CARDrstm1_GPIO2C4vccio5, IO_I2S1sdio2_PDMsdi2m0_CARDdetm1_GPIO2C5vccio5 and IO_I2S1sdio3_PDMsdi3m0_CARDiom1_GPIO2C6vccio5 respectively with high level meaning output.

When M0 IO is used, I2S2 can used as transmitter and receiver and the same time.

When M1 IO is used, IO_TSPd4_CIFdata4_SPIcsn0m2_I2S2lrcktxm1_USB3PHYdebug8_I2S2lrckrxm1_GPIO3B0vccio6 is connected to either of i2s2_lrck_rx and i2s2_lrck_tx at the same time, so I2S2 cannot be used as transmitter and receiver and the same time.

The I2S1 is also connected to the ACODEC which supports master and slave mode. When the ACODEC acts as a master, the signal i2s1_lrck_tx_in which connected to I2S1 can be selected from ACODEC or external IO by setting GRF_SOC_CON2[15].

Table 25-2 Interface Between I2S1 and ACODEC

Module Pin	Direction	Module Pin	Direction
i2s1_mclk	O	pin_mclk	I
i2s1_sclk_out	O	pin_sck_i	I
i2s1_sclk_in	I	pin_sck_o	O
i2s1_lrck_tx_out	O	pin_dac_ws_i	I
i2s1_lrck_tx_in	I	pin_dac_ws_o	O
i2s1_sdo0	O	pin_dac_sd_i	I

The I2S0 module is connected to the audio interface of HDMI, which supports 8 channels audio data transmitting.

Table 25-3 I2S Interface Between I2S2 and HDMI

Module Pin	Direction	Module Pin	Direction
i2s0_sclk_out	O	ii2sclk	I
i2s0_tx_lrck_out	O	ii2slrck	I
i2s0_sdo[3:0]	O	ii2sdata[3:0]	I

25.6 16.6 Application Notes

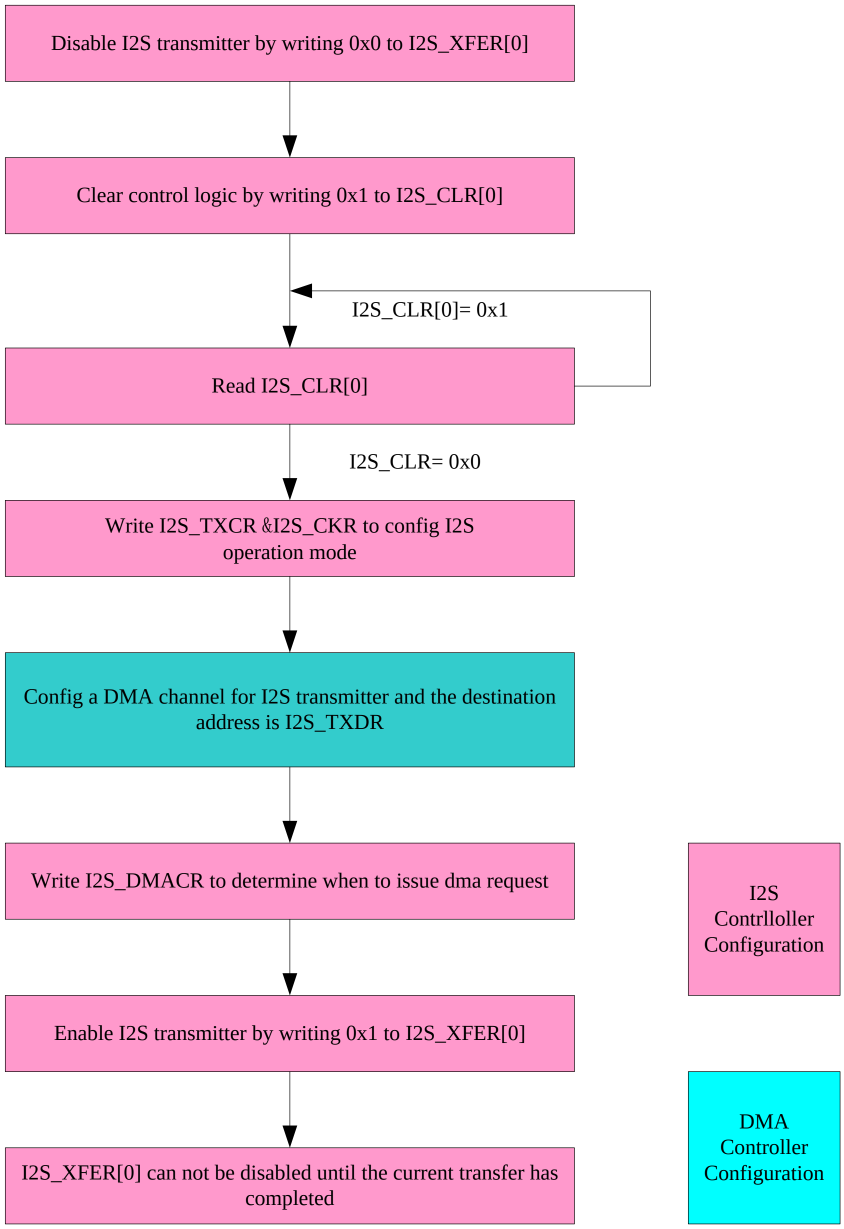


Fig. 25-11 I2S/PCM controller transmit operation flow chart

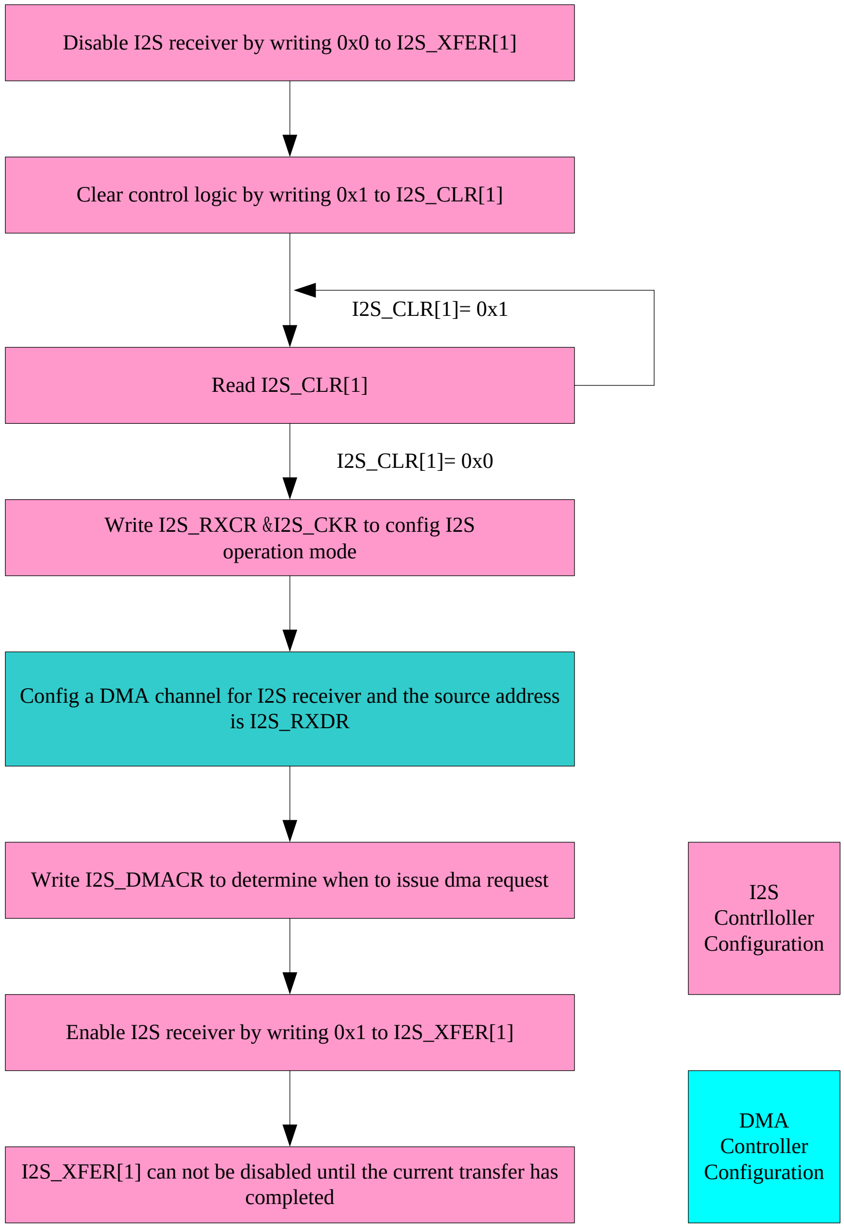


Fig. 25-12 I2S/PCM controller receive operation flow chart

Chapter 26 Graphics Process Unit (GPU)

26.1 Overview

The GPU is a hardware accelerator for 2D and 3D graphics systems. Its triangle rate can be 30 Mtris/s, pixel rate can be 300Mpix/s@300MHz.

The GPU supports the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1
- EGL 1.5

The GPU consists of:

- 2 Pixel Processors (PPs)
- 1 geometry Processor (GP)
- 2 Level2 Cache controller (L2)
- 1 Memory Management Unit (MMU) for each GP and PP included in the GPU

The GPU contains a 32-bit APB bus and 2 128-bit AXI bus. CPU configures GPU through APB bus, GPU read and write data through AXI bus.

26.2 Block Diagram

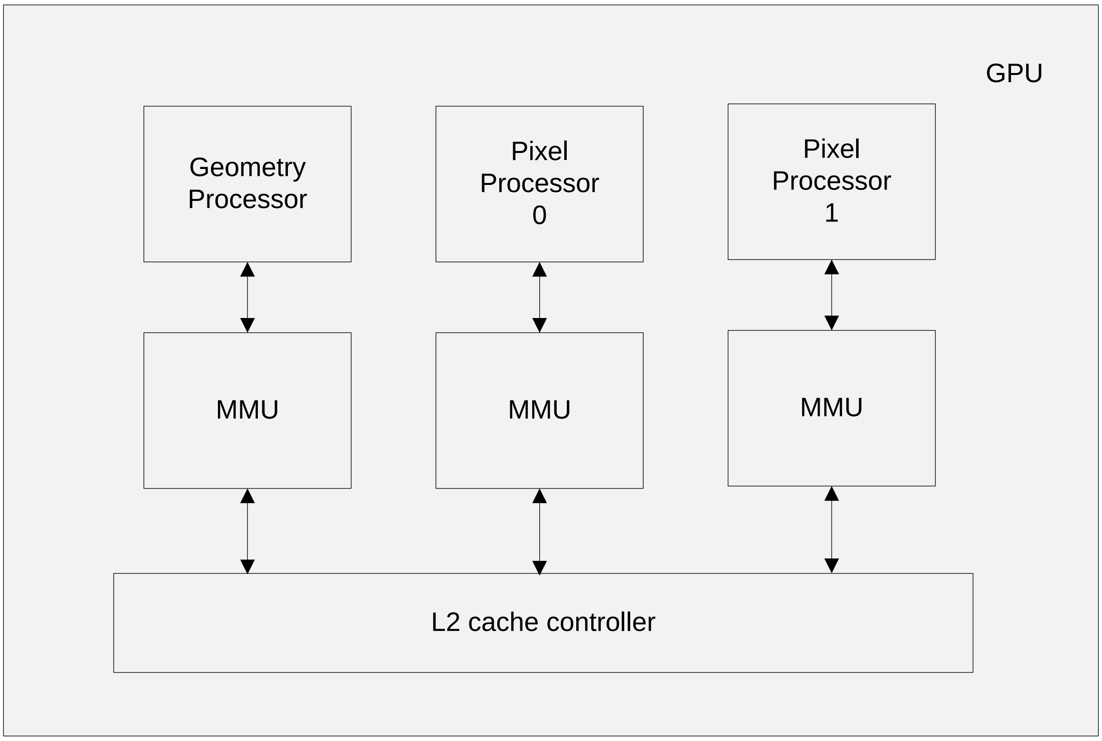


Fig. 26-1 GPU block diagram

The GPU contains 1 geometry processor, 2 pixel processors, 3 MMU and 2 L2 cache controller.

The pixel processor features are:

- each pixel processor used processes a different tile, enabling a faster turnaround
- programmable fragment shader
- alpha blending
- complete non-power-of-2 texture support
- cube mapping
- fast dynamic branching
- fast trigonometric functions, including arctangent
- framebuffer blend with destination Alpha

- indexable texture samplers
- line, quad, triangle and point sprites
- no limit on program length
- perspective correct texturing
- point sampling, bilinear, and trilinear filtering
- programmable mipmap level-of-detail biasing and replacement
- stencil buffering, 8-bit
- two-sided stencil
- unlimited dependent texture reads
- 4-level hierarchical Z and stencil operations
- 4-bit per texel compressed texture format
- Up to 512 times Full Scene Anti-Aliasing (FSAA). 4x multisampling by 128x supersampling.

The geometry processor features are:

- two programmable vertex shaders
- flexible input and output formats
- autonomous operation tile list generation
- indexed and non-indexed geometry input
- primitive constructions with points, lines, triangles and quads.

The L2 cache controller features are:

- 64KB
- 4-way set-associative
- supports up to 32 outstanding AXI transactions
- implements a standard pseudo-LRU algorithm
- cache line and line fill burst size is 64 bytes
- supports eight to 64bytes uncached read bursts and write bursts
- 128-bit interface to memory sub-system
- support for hit-under-miss and miss-under-miss with the only limitation of AXI ordering rules.

The MMU features are:

- accesses control registers through the bus infrastructure to configure the memory system.
- each processor has its own MMU to control and translate memory accesses that the GPU initiates.

APB broadcast features are:

- configuration of multiple PPs in parallel
- the ability to use a single read to poll multiple PP interrupts.

DMA features are:

- The register DMA reduces the number of required APB transactions by configuring the rest of the GPU using configuration data stored in main memory. The driver writes the configuration data for each frame to main memory while the previous frame is rendered. The register DMA unit performs the setup after the previous frame is completed. This reduces the system overhead between frames, and reduces the workload for the CPU. The DMA simplifies transfer of GPU commands and data from memory to the pixel processors. A counter in the DMA determines how many register write packages are processed.

Load balancing features are:

- The address of the tile lists and the number of tiles in the framebuffer is programmable. The dynamic load balancing unit assigns a new tile to the different pixel processors   ecause they complete the previous tile. The dynamic load balancing unit iterates over the frame in a Z-order pattern starting at the first tile for pixel processor 0-3 and the last tile for pixel processor 4-7. This ensures that the pixel processors connected to the same level 2 cache process nearby tiles. This improves cache efficiency. This also balances the workload for the different pixel processors regardless of the frame content.

26.3 Register Description

The GPU base addressis 0XFF30_0000.

26.4 Interface Description

Fig. 26-2 GPU interrupt connection

Pmu interrupt keeps 0 because GPU is not configured to support PMU function.

Chapter 27 Video Digital Analog Converter (VDAC)

27.1 Overview

Video DAC PHY is a small-sized, 27~300MHz, 1-channel, 10bit, high-speed D/A converter optimized for video or graphic applications. This IP designed to support Component(Pr,Y,Pb),Composite(CVBS), and S-Video(Y,C) signal standards for "consumer quality".

27.1.1 Features

- 10-bit resolution
- Single channel
- Up to 300Msps throughput rate
- Programmable current output: 14.7mA~ 34.8mA with 64 adjustable steps
- Current consumption: 1mA @Iout = 14.7mA, 39mA @Iout = 34.8mA
- 57dBc SFDR @Iout = 14.7, fclk = 300MHz and fout = 5MHz;45dBc SFDR @Iout = 34.8, fclk = 300MHz and fout = 5MHz;
- Clock frequency : 27MHz to 300MHz
- Cable connection detection
- Build-in bandgap reference
- 1.8V supply for analog and 1.0V supply for digital

27.2 Block Diagram

The architecture is shown in the following figure.

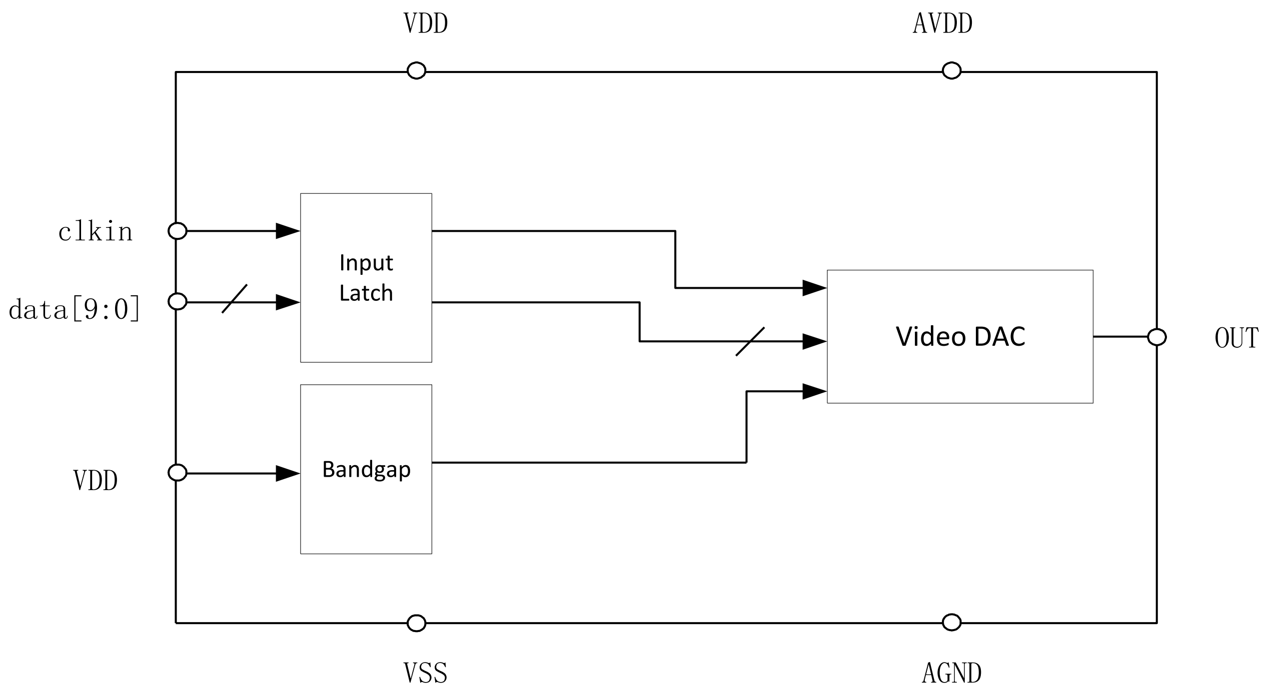


Fig. 27-1 VDAC Block Diagram

27.3 Function Description

27.3.1 System configure write timing for apb bus

The Write transfer starts with the address, write data, write signl all changing after the rising edge of the clock. The first clock cycle of the transfer is called the SETUP cycle. After the following clock edge the enable signal PENABLE is asserted and this indicates that

ENABLE cycle is taking place. The address, data and control signals all remain valid throughout the ENABLE cycle. The transfer completes at the end of this cycle.
The enable signal, PENABLE, will be de-asserted at the end of the transfer. The select signal will also go LOW, unless the transfer is to be immediately follower by another transfer to the sample peripheral.
In order to reduce power consumption the address signal and the write signal will not change after a transfer until the next access occurs.

27.3.2 System configure read timing for apb bus

The timing of the address, write, select and strobe signals are all the same as for the write transfer. In the case of a read, the slave must provide the data during then ENABLE cycle. The data is sampled on the rising edge of clock at the end of the ENABLE cycle.

27.4 Register Description

27.4.1 Internal Address Mapping

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

27.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
VDAC_VDAC0	0x0000	W	0x000000c0	VDAC0
VDAC_VDAC1	0x0280	W	0x00000070	VDAC1
VDAC_VDAC2	0x0284	W	0x00000020	VDAC2
VDAC_VDAC3	0x0288	W	0x00000030	VDAC3

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

1.4.3 Detail Register Description

VDAC_VDAC0
Address: Operational Base + offset (0x0000)
VDAC0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x1	RST_ANA soft analog reset_n, low reset soft analog reset_n, low reset
6	RW	0x1	RST_DIG soft digital reset_n, low reset soft digital reset_n, low reset
5:0	RO	0x0	reserved

VDAC_VDAC1
Address: Operational Base + offset (0x0280)
VDAC1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CUR_REF select typical current reference select typical current reference
3:2	RO	0x0	reserved
1	RW	0x0	DR_PWR_DOWN vdac driver power down vdac driver power down 1: power down 0: power on
0	RW	0x0	BG_PWR_DOWN vdac band gap power down vdac band gap power down 1: power down 0: power on

VDAC_VDAC2

Address: Operational Base + offset (0x0284)

VDAC2

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x20	CUR_CTR output current control for DAC output current control for DAC tvdac_sw[5:0]

VDAC_VDAC3

Address: Operational Base + offset (0x0288)

VDAC3

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x1	CAB_EN Enable cable connection detection for DAC Enable cable connection detection for DAC 1: enable 0: disable
4	RW	0x1	CAB_REF reference voltage for cable disconnection detection of DAC reference voltage for cable disconnection detection of DAC 0: select 500mV 1: select 800mV
3:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	CAB_FLAG status output for DAC cable connection detection (1 means cable disconnection)

27.5 Application Notes

27.5.1 CABLE DETECTION

The DAC channel contains a cable detection circuit to detect the cable plug condition. For typical application, cable with 75Ω characteristic impedance is used and DAC output is terminated by 75Ω double termination. In such case, a 75Ω source termination resistance is connected to ground at DAC output end. The 75Ω source termination resistance combined with 75Ω load termination resistance results in an equivalent load resistance of 37.5Ω.

Therefore, the equivalent load resistance for DAC output is 37.5Ω when cable is connected. It becomes 75Ω when cable is not connected. Compared to the case cable is connected, DAC output level will be twice in the case that cable is not connected with identical output current.

To start cable detection, controller should enable this function (controlled by register tvdac_dispdet_en) and set the 10-bit input data for a DAC channel to be middle level. Then controller should select a proper reference voltage(controlled by register tvdac_sw), which will be compared with DAC output level to judge whether cable is connected or not. The reference voltage selection is shown in following table.

Tvdac_sw	Tvdac_dispdet_sel	Reference voltage
6’b000000~6’b011111	1’b0	500mV
6’b100000~6’b111111	1’b1	800mV

If DAC output level is larger than the reference voltage, the cable detection flag signal(tvdac_dispdet) will be high and it means cable is disconnected. Otherwise, the cable detection flag signal will be low and it means cable is connected.

Tvdac_dispdet	1	Cable is connected
	0	Cable is disconnected

27.5.2 TYPICAL CONFIGURATION

The typical configuration is shown in following figure. DAC output is connected through 75Ω cable with 75Ω double termination.

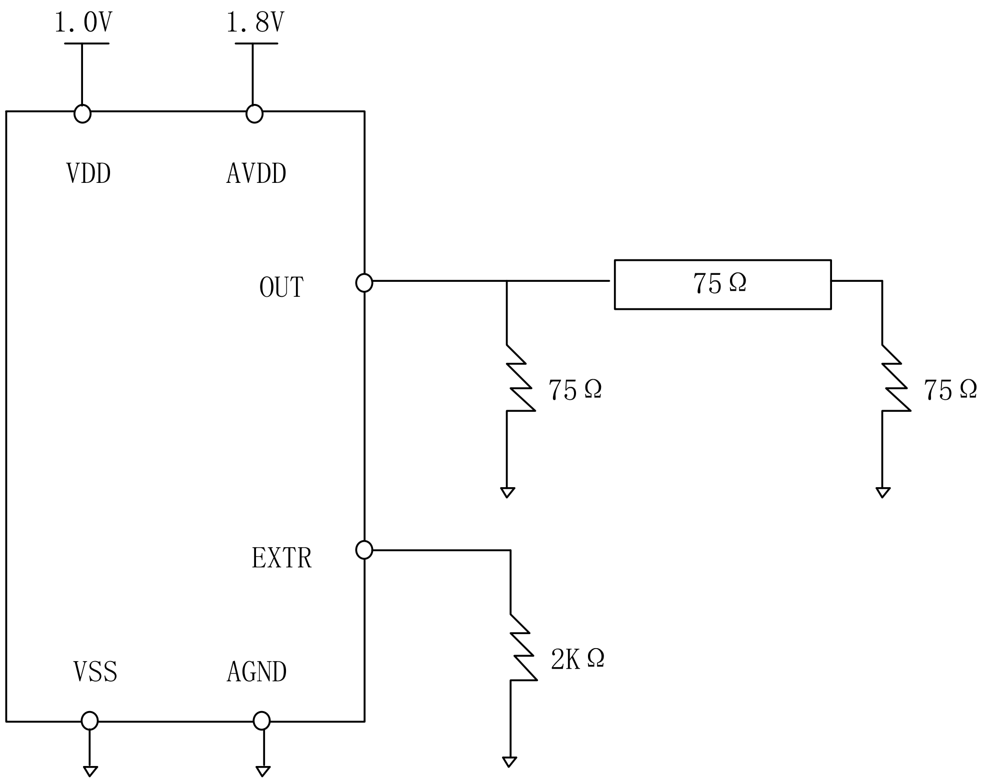


Fig. 27-2 VDAC Block Diagram

Analog supply AVDD should be connected to 1.8V power with decoupling. The digital supply VDD should be connected to digital core.

If external resistor is selected to produce reference current, EXTR should connect a 2KΩ resistor to ground.

Video DAC is suggested to placed close to the connector, in order to reduce signal noise and reflection due to impedance mismatch.

The DAC outputs are suggested to connect a 75Ω source termination resistance to ground. The termination resistors should be placed close to video DAC outputs to minimize reflection.

27.5.3 INSTRUCTION TO BRING UP VDAC

The following is a step by step instruction for bringing up the VDAC to your system, we use APB bus to configure VDAC.

Step1. Turn on entire system.

Step2. Configure 0xb3(data) to 0x280(address) to disable VDAC.

Step3. Configure 0x39(data) to 0x284(address) for current control.

Step4. Configure VOP.

Step5. Configure 0xb0(data) to 0x280(address) to enable VDAC and for typical current reference.

Step6. Now, TVDAC is ready to go. Start your test.