Rockchip
RK3288
Technical Reference Manual
Part1

Revision 1.2
Mar. 2017

Revision History

Date	Revision	Description
2017-3-21	1.2	Update
2015-8-20	1.1	Update
2015-6-26	1.0	Update-Part1 System and System control

Warranty Disclaimer

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

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

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

Copyright and Patent Right

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

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

Trademarks

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

Confidentiality

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

Reverse engineering or disassembly is prohibited.

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

Copyright © 2015 Rockchip Electronics Co., Ltd.

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

RK3288 support to boot from internal bootrom, which support remap function by software programming. Remap is controlled by SGRF_SOC_CON0 bit[11].

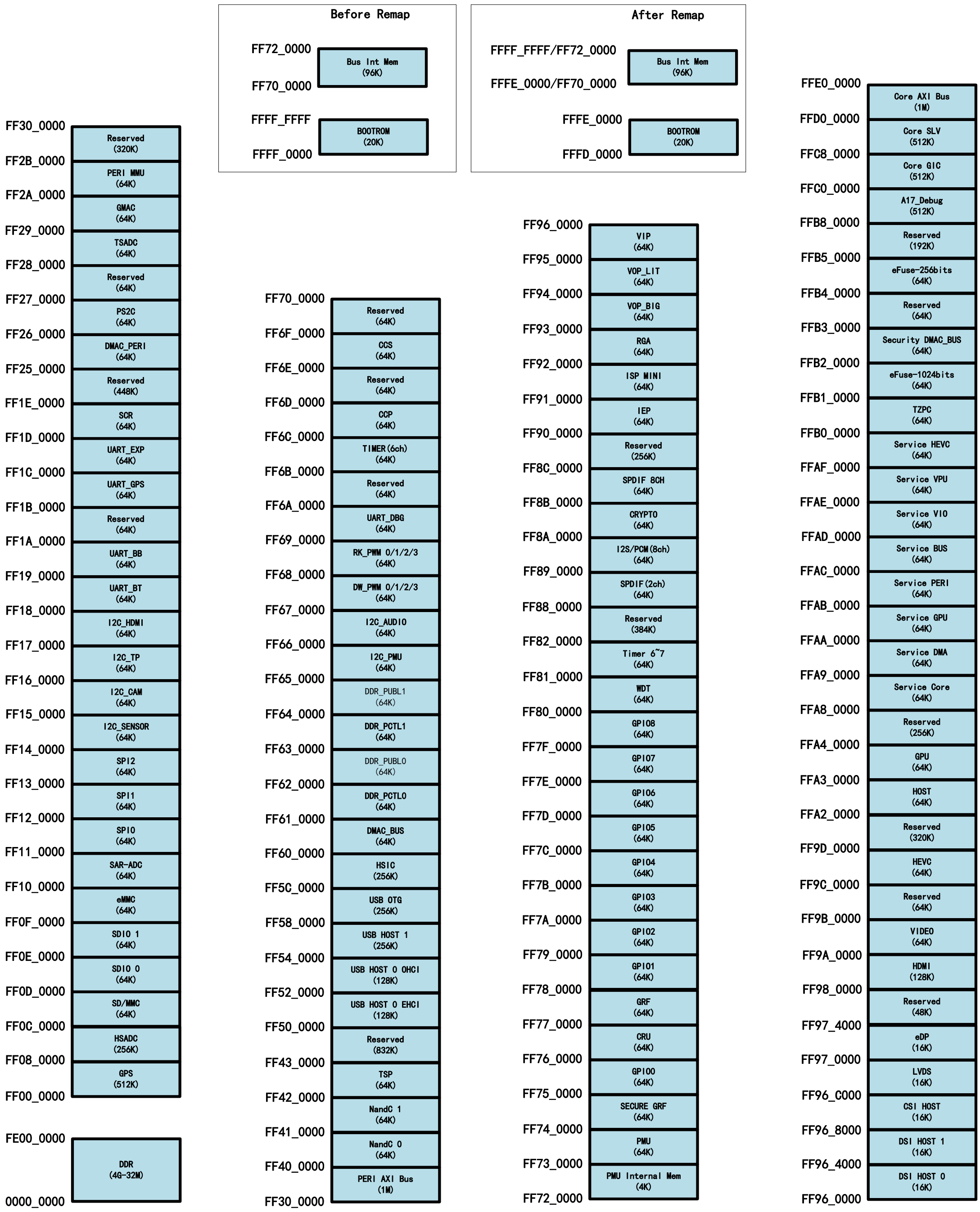


Fig. 1-1 RK3288 Address Mapping

1.2 System Boot

RK3288 provides system boot from off-chip devices such as SDMMC card, 8bits async nand flash or toggle nand flash, SPI nor or nand, and eMMC memory. 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 secure boot mode and non-secure boot mode
Support system boot from the following device:
8bits Async Nand Flash
8bits Toggle Nand Flash
SPI2_CS0 interface
eMMC interface
SDMMC Card
Support system code download by USB OTG

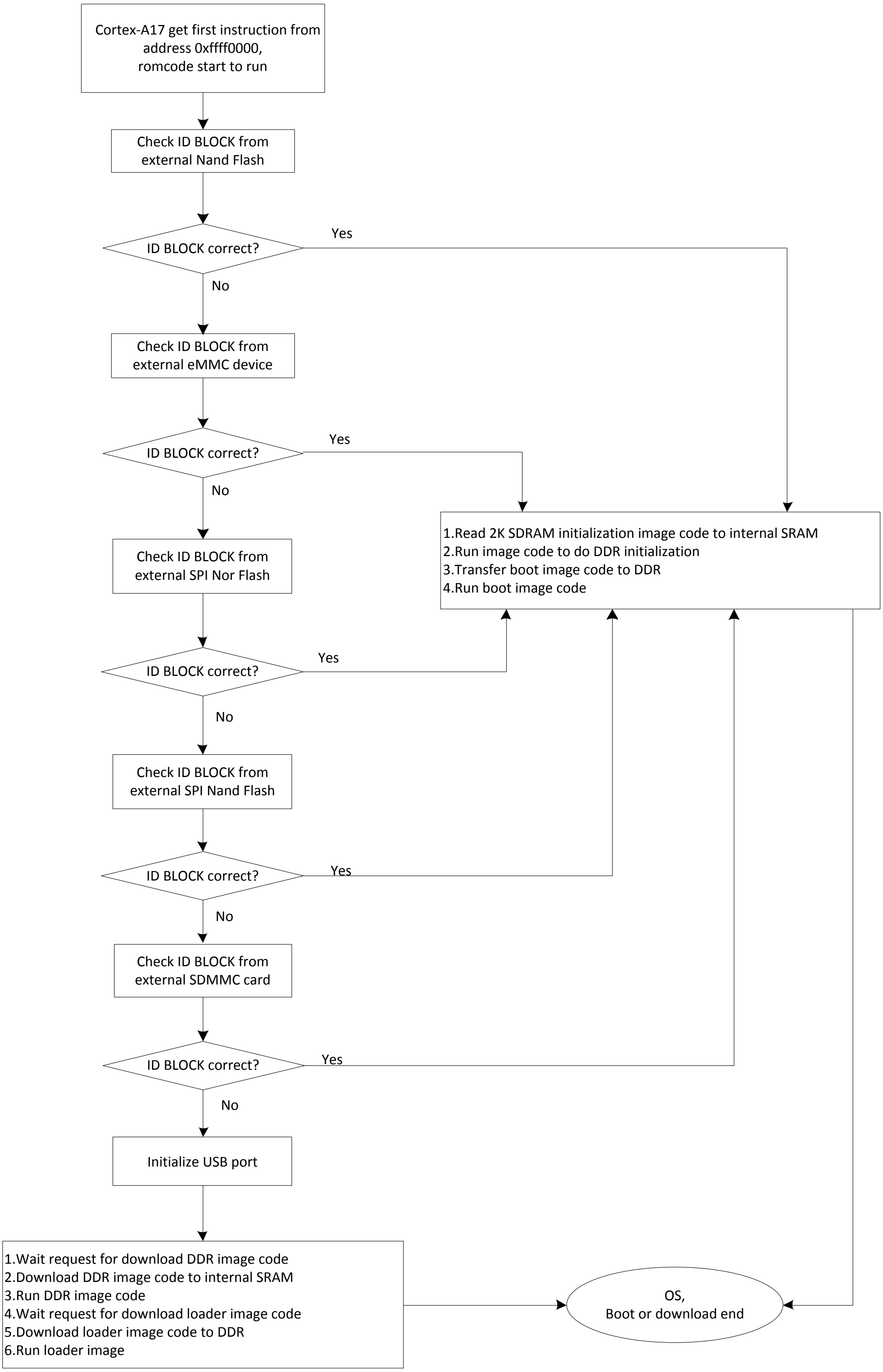

Cortex-A17 get first instruction from address 0xffff0000, romcode start to run
Check ID BLOCK from external Nand Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external eMMC device
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SPI Nor Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SPI Nand Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SDMMC card
ID BLOCK correct?
Yes
No
Initialize USB port
1.Read 2K SDRAM initialization image code to internal SRAM
2.Run image code to do DDR initialization
3.Transfer boot image code to DDR
4.Run boot image code
1.Wait request for download DDR image code
2.Download DDR image code to internal SRAM
3.Run DDR image code
4.Wait request for download loader image code
5.Download loader image code to DDR
6.Run loader image
OS,
Boot or download end

Fig. 1-2 RK3288 boot procedure flow

1.3 System Interrupt connection

RK3288 provides an general interrupt controller(GIC) for Cortex-A17 MPCore processor, which has 112 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 13.

Table 1-1 RK3288 Interrupt connection list

IRQ Type	IRQ ID	Source(spi)	Polarity
PPI	26	HYPERVISOR TIMER	High level
	27	VIRTUAL TIMER	High level
	29	SECURE PHYSICAL TIMER	High level
	30	NON-SECURE PHY TIMER	High level
SPI	32	DMAC_BUS (0)	High level
	33	DMAC_BUS (1)	High level
	34	DMAC_PERI (0)	High level
	35	DMAC_PERI (1)	High level
	36	UPCTL 0	High level
	37	UPCTL 1	High level
	38	GPU_IRQJOB	High level
	39	GPU_IRQMMU	High level
	40	GPU_IRQGPU	High level
	41	VIDEO ENCODER	High level
	42	VIDEO DECODER	High level
	43	VIDEO MMU	High level
	44	HEVC	High level
	45	VIP	High level
	46	ISP	High level
	47	VOP_BIG	High level
	48	VOP_LIT	High level
	49	IEP	High level
	50	RGA	High level
	51	DSI 0 HOST	High level
	52	DSI 1 HOST	High level
	53	CSI HOST 0	High level
	54	CSI HOST 1	High level
	55	USB OTG	High level
	56	USB HOST 0 EHCI	High level
	57	USB HOST 1	High level
	58	N/A	High level
	59	GMAC	High level
	60	GMAC PMT	High level
	61	GPS	High level
	62	GPS TIMER	High level
	63	HS-ADC/TSI	High level
	64	SD/MMC	High level
	65	SDIO 0	High level
	66	SDIO 1	High level
	67	eMMC	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	68	SARADC	High level
	69	TSADC	High level
	70	NANDC 0	High level
	71	PERI MMU	High level
	72	NANDC 1	High level
	73	USB HOST 0 OHCI	High level
	74	TPS	High level
	75	SCR	High level
	76	SPI0	High level
	77	SPI1	High level
	78	SPI2	High level
	79	PS2C	High level
	80	CRYPTO	High level
	81	HOST PULSE 0	High level
	82	HOST PULSE 1	High level
	83	HOST 0	High level
	84	HOST 1	High level
	85	I2S/PCM (8ch)	High level
	86	SPDIF(8ch)	High level
	87	UART_BT	High level
	88	UART_BB	High level
	89	UART_DBG	High level
	90	UART_GPS	High level
	91	UART_EXP	High level
	92	I2C_PMU	High level
	93	I2C_AUDIO	High level
	94	I2C_SENSOR	High level
	95	I2C_CAM	High level
	96	I2C_TP	High level
	97	I2C_HDMI	High level
	98	TIMER 6CH 0	High level
	99	TIMER 6CH 1	High level
	100	TIMER 6CH 2	High level
	101	TIMER 6CH 3	High level
	102	TIMER 6CH 4	High level
	103	TIMER 6CH 5	High level
	104	TIMER 2CH 0	High level
	105	TIMER 2CH 1	High level
	106	PWM0	High level
	107	PWM1	High level
	108	PWM2	High level
	109	PWM3	High level
	110	RK_PWM	High level
	111	WDT	High level

IRQ Type	IRQ ID	Source(spi)	Polarity
	112	PMU	High level
	113	GPIO0	High level
	114	GPIO1	High level
	115	GPIO2	High level
	116	GPIO3	High level
	117	GPIO4	High level
	118	GPIO5	High level
	119	GPIO6	High level
	120	GPIO7	High level
	121	GPIO8	High level
	122	AHB ARBITER0 (USB)	High level
	123	AHB ARBITER1 (EMEM)	High level
	124	AHB ARBITER2 (MMC)	High level
	125	USBOTG_ID	High level
	126	USBOTG_BVALID	High level
	127	USBOTG_LINESTATE	High level
	128	USBHOST0_LINESTATE	High level
	129	USBHOST1_LINESTATE	High level
	130	eDP DP	High level
	131	SDMMC_DETECT_N	High level
	132	SDIO0_DETECT_N	High level
	133	SDIO1_DETECT_N	High level
	134	HDMI WAKEUP	High level
	135	HDMI	High level
	136	CCP	High level
	137	CCS	High level
	138	SDMMC DETECT DUAL EDGE	High level
	139	GPIO7_B3 DUAL EDGE	High level
	140	GPIO7_C6_DUAL EDGE	High level
	141	GPIO8_A2_DUAL EDGE	High level
	142	eDP HDMI	High level
	143	HEVC MMU	High level
	183	PMUIRQ[0]	High level
	184	PMUIRQ[1]	High level
	185	PMUIRQ[2]	High level
	186	PMUIRQ[3]	High level

1.4 System DMA hardware request connection

RK3288 provides 2 DMA controllers: DMAC_BUS inside bus system and DMAC_PERI inside peripheral system. As for DMAC_BUS, there are 6 hardware request ports. Another, 15 hardware request ports are used in DMAC_PERI, the trigger type for each of them is high level, not programmable. For detailed descriptions of DMAC_BUS and DMAC_PERI, please refer to related section.

Table 1-2 RK3288 DMAC_BUS Hardware request connection list

Req Number	Source	Polarity
0	I2S/PCM(8CH) TX	High level
1	I2S/PCM(8CH) RX	High level
2	SPDIF(2CH) TX	High level
3	SPDIF(8CH) TX	High level
4	UART_DBG TX	High level
5	UART_DBG RX	High level

Table 1-3 RK3288 DMAC_PERI Hardware request connection list

Req Number	Source	Polarity
0	HS-ADC/TSI	High level
1	UART_BT TX	High level
2	UART_BT RX	High level
3	UART_BB TX	High level
4	UART_BB RX	High level
5	N/A	N/A
6	N/A	N/A
7	UART_GPS TX	High level
8	UART_GPS RX	High level
9	UART_EXP TX	High level
10	UART_EXP RX	High level
11	SPI0 TX	High level
12	SPI0 RX	High level
13	SPI1 TX	High level
14	SPI1 RX	High level
15	SPI2 TX	High level
16	SPI2 RX	High level

Chapter 2 Clock and Reset Unit (CRU)

2.1 Overview

The CRU is an APB slave module that is designed for generating all of the system clocks, resets of chip. CRU generates system clock 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 five 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

The CRU comprises with:

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

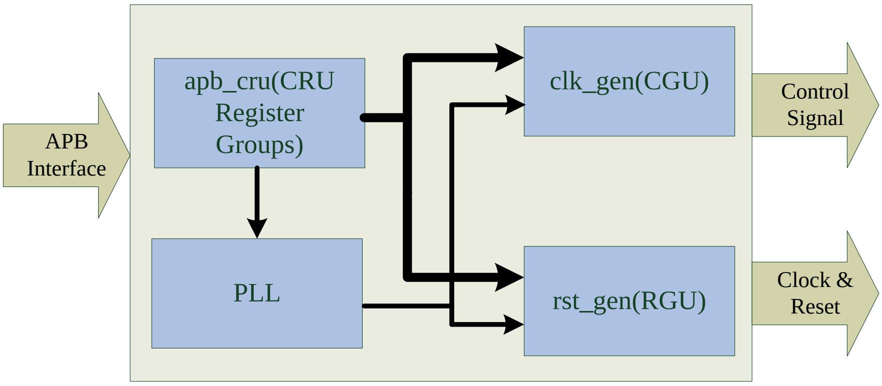


Fig. 2-1 CRU Architecture

2.3 System Clock Solution

2.3.1 CRU architecture

The following diagrams show CRU clock architecture (mux and divider information).

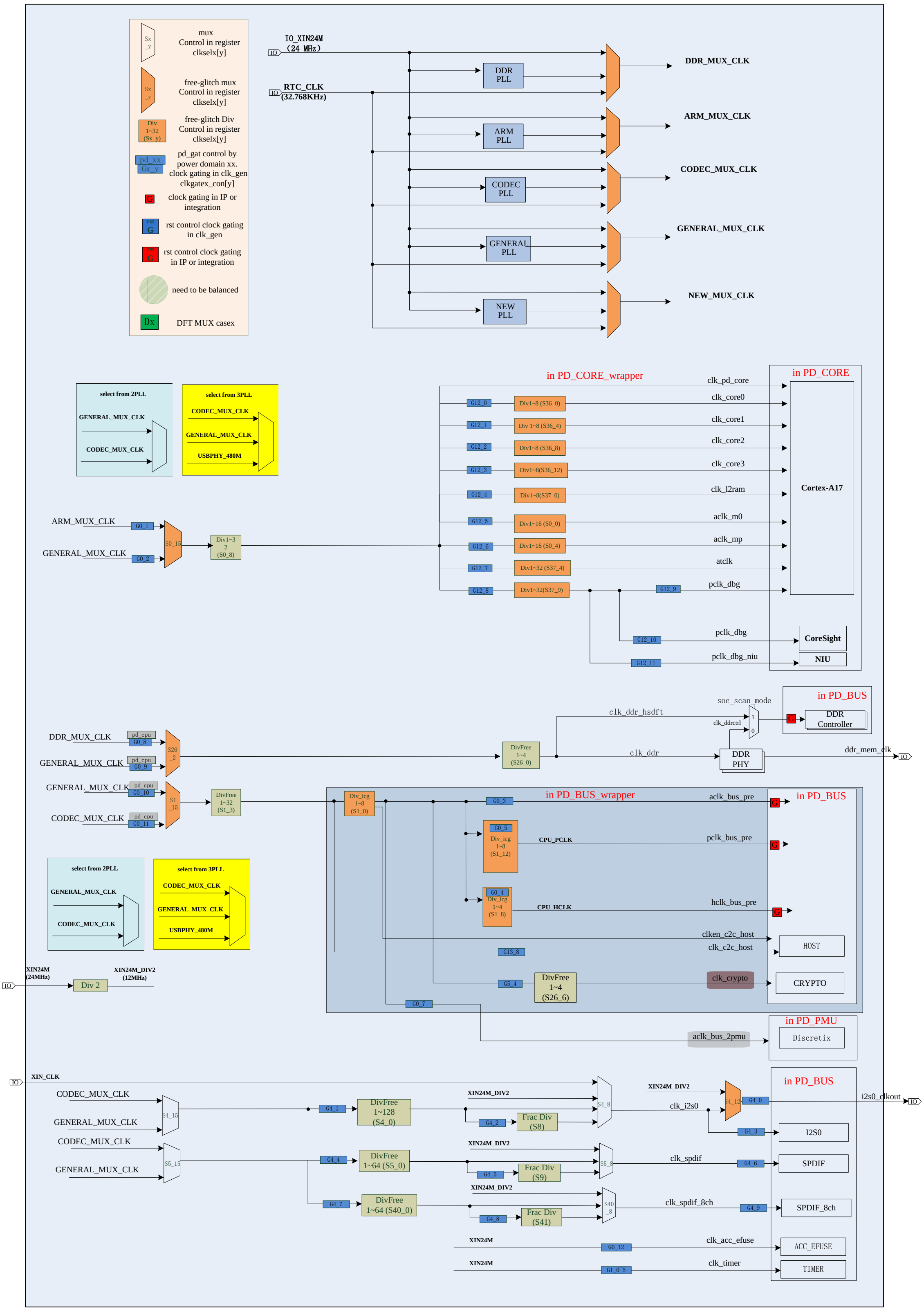


Fig. 2-2 CRU Clock Architecture Diagram 1

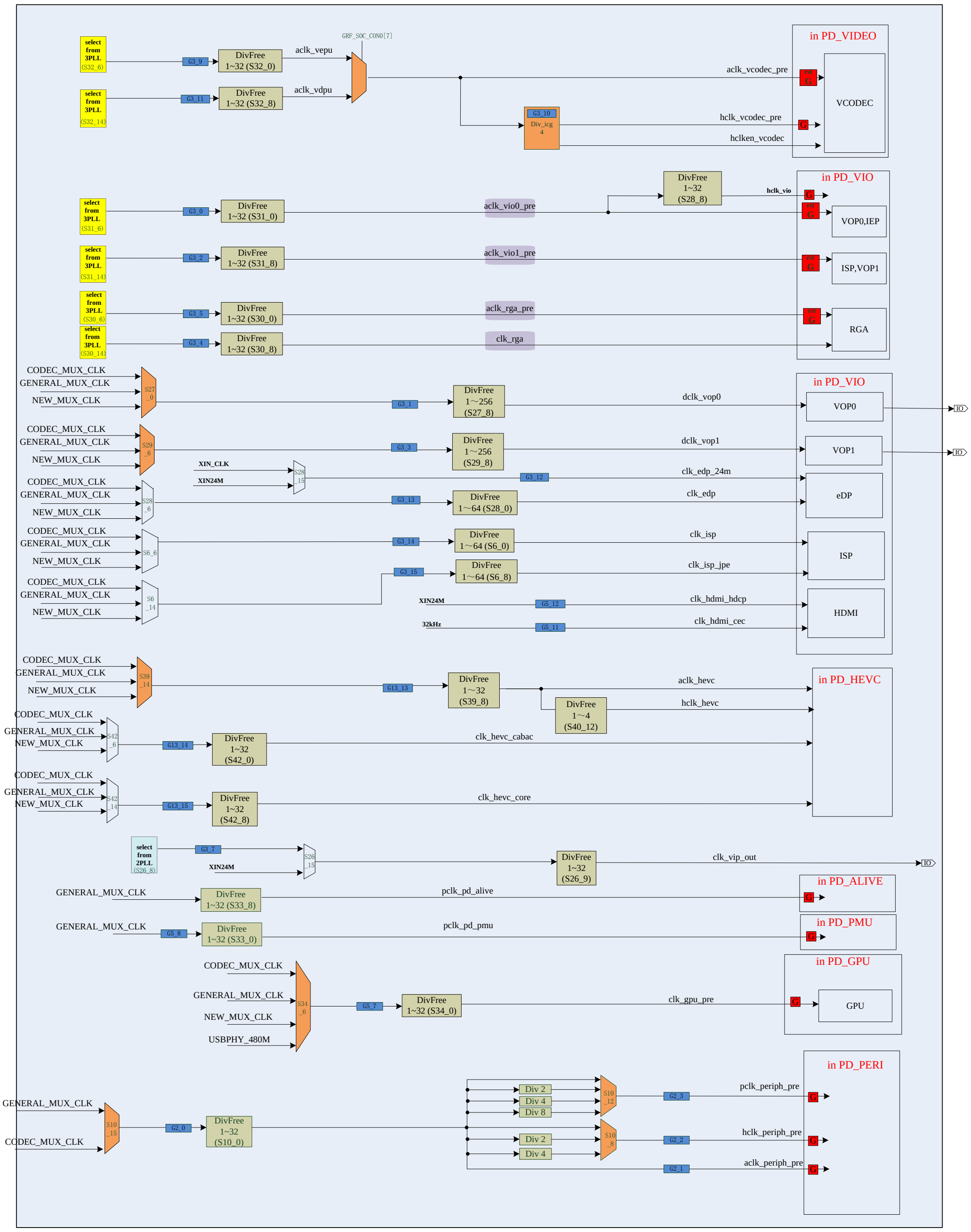


Fig. 2-3 CRU Clock Architecture Diagram 2

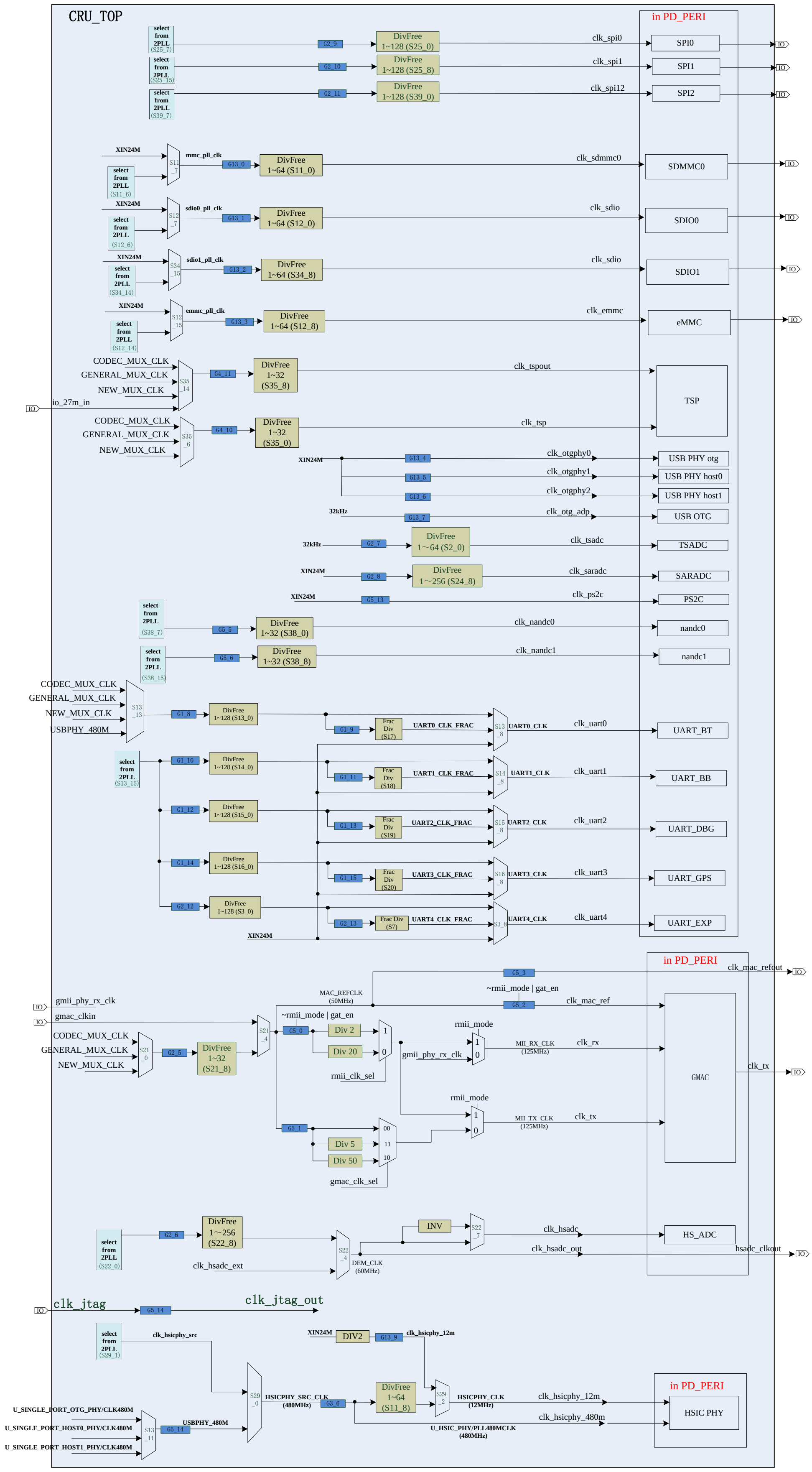


Fig. 2-4 CRU Clock Architecture Diagram 3

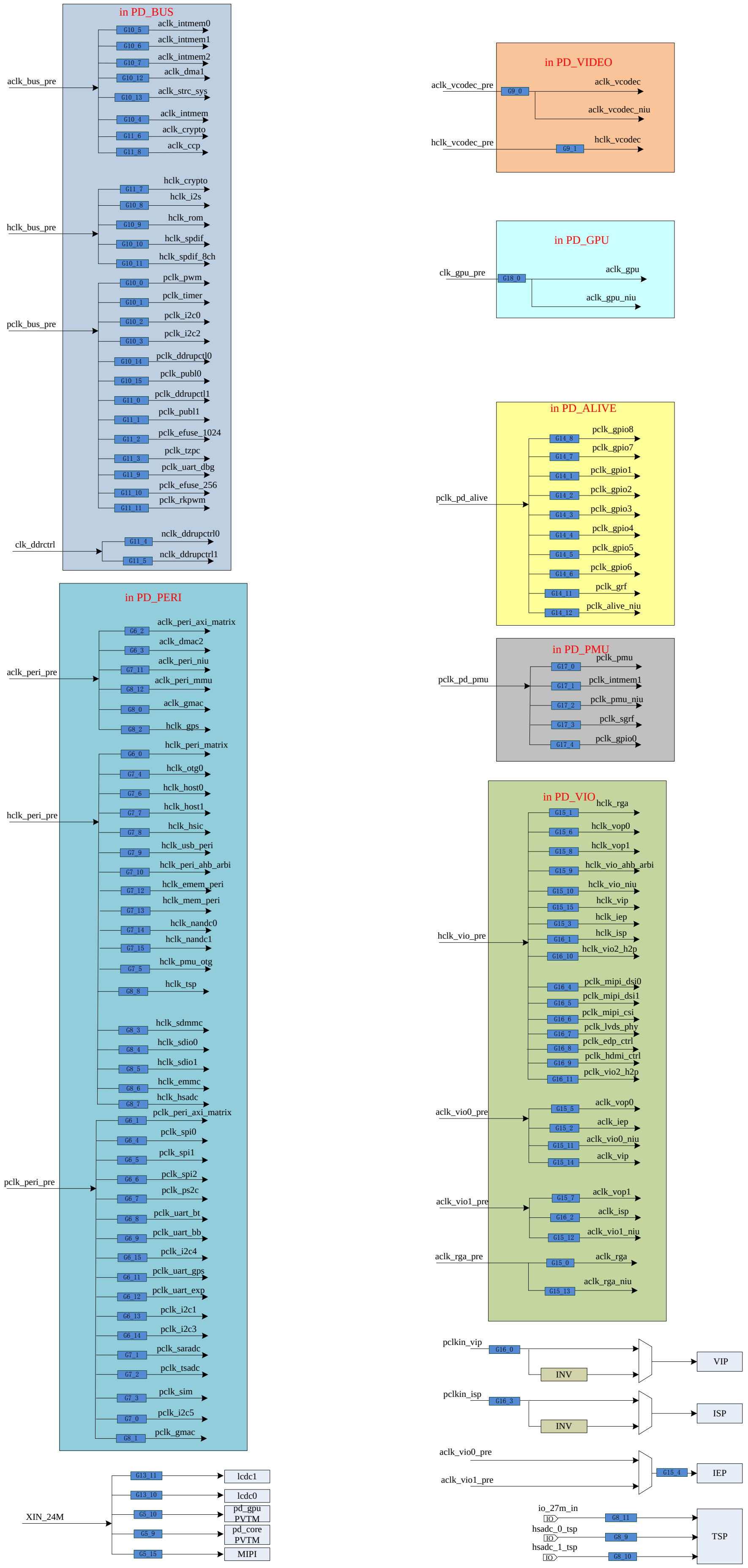


Fig. 2-5 CRU Clock Architecture Diagram 4

2.4 System Reset Solution

The following diagrams show reset architecture in this block.

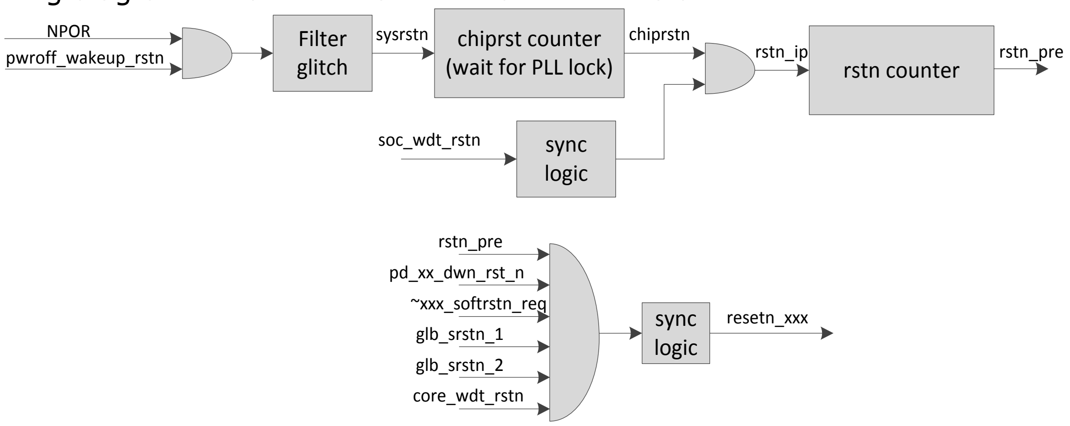


Fig. 2-6 Reset Architecture Diagram

Reset source of each reset signal includes hardware reset (NPOR), power-off mode wakeup reset (pwroff_wakeup_rstn), soc watch dog reset (soc_wdt_rstn), power domain power down reset (pd_xx_dwn_rst_n), software reset request (xxx_softrstn_req), global software reset1 (glb_srstn_1), global software reset2 (glb_srstn_2) and A9 core watch dog reset (core_wdt_rstn).

The 'xx' of pd_xx_dwn_rst_n represents core0, core1, core2, core3, cs, cpu, peri, vio, video or gpu. The 'xxx' of resetn_xxx and xxx_softrstn_req is the module name.

Pwroff_wakeup_rstn is the reset when wakeup from the power-off mode, it will reset the all SOC logic except internal PMU.

Soc_wdt_rstn is the reset from watch-dog IP in the SoC, but core_wdt_rstn is the reset from A9 core watch-dog block.

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-clear by hardware. Glb_srstn_1 will reset the all logic except PMU_SYS_REG0~3. And Glb_srstn_2 will reset the all logic except PMU_SYS_REG0~3, GRF and all GPIOs.

2.5 Function Description

There are five PLLs:

ARM PLL, DDR PLL, CODEC PLL, GENERAL PLL and NEW PLL in CRU.

PLLs all can be set to slow mode or deep slow mode, directly output selectable 24MHz or 32.768kHz. 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 three PLLs (CODEC PLL, GENERAL PLL and NEW PLL).

To provide some specific frequency, another solution is integrated: fractional divider. In order to be sure 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 reset can be software generated.

2.6 PLL Introduction

2.6.1 Overview

This chip uses 5GHz PLL for all four PLLs. The 5GHz PLL is a general purpose, high-performance PLL-based clock generator. The VCO operates from 440 MHz to 5000MHz, but the frequency range of 440 MHz to 2200MHz is recommended. It has a programmable output frequency, which ranges from 27.5 MHz to 5000 MHz configured through a 6-bit input divider, a 13-bit feedback divider and a 4-bit output divider. Around 50% duty cycle of output clocks can be achieved by enabling the output divider. It can also be used as a clock buffer through a bypass

mode that bypasses and powers down the PLL. A full power-down mode is also available.
2.2GHz PLL supports the following features:

- Fully integrated, including loop filter
- Power supply: 1.0V single power supply
- VCO operating range: 440MHz – 5000MHz(440MHz – 2200MHz is recommended)
- Output frequency range: 27.5MHz – 5000MHz (27.5MHz – 2200MHz is recommended)
- Input frequency range: 269kHz – 5000MHz (269kHz – 2200MHz is recommended)
- PFD comparison frequency range: 269kHz – 5000MHz(269kHz – 2200MHz is recommended)
- Low power consumption: 3mA @ 1100MHz during normal operation
- Contains 6-bit input, 13-bit feedback and 4-bit output dividers
- Input divider value range: 1–64
- Feedback divider value range: 1–4096
- Output divider value range: 1, 2-16 (even only)
- Bandwidth adjustment of div. reference: 1–4096
- Output duty cycle: +/-5% (/1), +/-2% (/N)
- Period jitter (P-P) (max): +/-2.5% output cycle
- Reset pulse width (min): 5us
- Lock time (min allowed): 500 div. reference cycles
- Freq. overshot (full-~/half-~) (max): 40%/50%
- Ref. input jitter (long-term, P-P) (max): 2% div. reference cycle
- Reference H/L pulse width (min) : 230ps
- Bypass and Power-down mode
- Lock detector

2.6.2 Block diagram

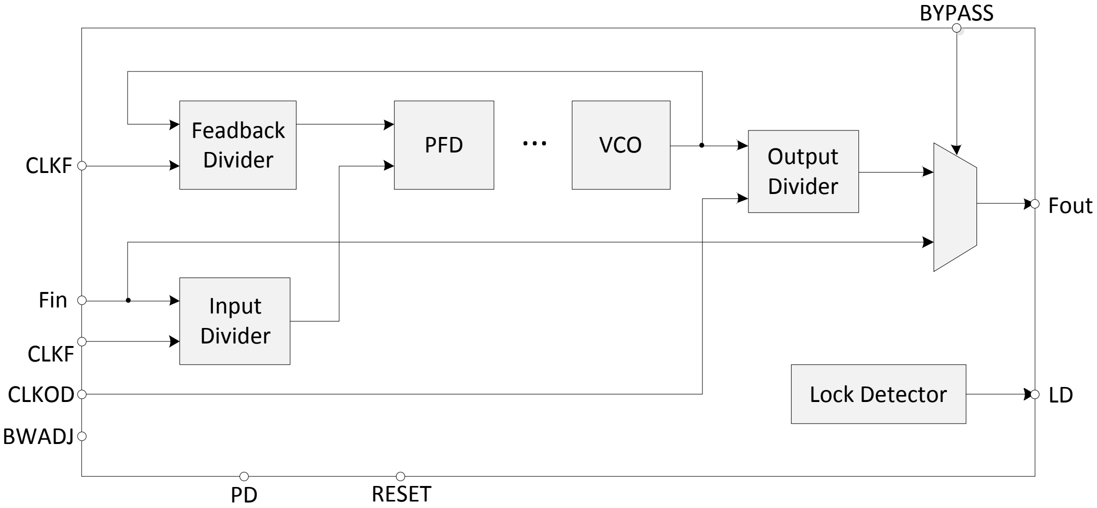


Fig. 2-7 PLL Block Diagram

2.6.3 Operation mode

- Locked

The positive edges of the PLL feedback and reference signals are phase aligned in normal operation. Because the feedback signal is internal, NO phase relationship is guaranteed between RCLK and CLKOUT. The output clock frequency is programmable through the divider setting of CLKR[5:0], CLKF[12:0] and CLKOD[3:0].

- Reset (RESET=1)

The PLL outputs a fixed free-running frequency in the range of 20MHz to 200MHz for a divide by 1 output depending on the specific PLL type.

- Power-down (PWRDN=1)

All analog circuitry in the PLL is turned off so as to only dissipate leakage current. The digital dividers are not affected.

Bypass (BYPASS=1)

The reference input is bypassed directly to the outputs.

Test (TEST=1)

The reference input drives all dividers cascaded one after the other for production testing.

2.6.4 PLL Bandwidth Adjustment

The loop bandwidth (BW) of the PLL can be adjusted using BWADJ[11:0]. The bandwidth is given by: BW = nom_BW*sqrt(NF / 2 / NB), where nom_BW is approximately given by: nom_BW = Fref / (NR*20), and Fref is the reference clock frequency. The damping factor (D) is approximately given by: D = nom_D*sqrt(NF / 2 / NB), where nom_D is approximately 1. Because the damping factor changes with bandwidth settings, the bandwidth is practically limited to: nom_BW/sqrt(2) < BW < nom_BW*sqrt(2), in order to limit the damping factor range to 0.7 - 1.4. The -3dB bandwidth (Fbw_3dB) is approximately given by: Fbw_3dB = 2.4 * nom_BW * (NF / 2 / NB). The recommended setting for NB is NF / 2, which will yield the nominal bandwidth. Note that nom_BW and nom_D are chosen to result in optimal PLL loop dynamics.

2.7 Register Description

This section describes the control/status registers of the design.

2.7.1 CRU Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x00000b01	ARM PLL configuration register0
CRU_APLL_CON1	0x0004	W	0x000003e7	ARM PLL configuration register1
CRU_APLL_CON2	0x0008	W	0x000001f3	ARM PLL configuration register2
CRU_APLL_CON3	0x000c	W	0x00000008	ARM PLL configuration register3
CRU_DPLL_CON0	0x0010	W	0x00000b03	DDR PLL configuration register0
CRU_DPLL_CON1	0x0014	W	0x0000031f	DDR PLL configuration register1
CRU_DPLL_CON2	0x0018	W	0x0000018f	DDR PLL configuration register2
CRU_DPLL_CON3	0x001c	W	0x00000008	DDR PLL configuration register3
CRU_CPLL_CON0	0x0020	W	0x00000b03	CODEC PLL configuration register0
CRU_CPLL_CON1	0x0024	W	0x000002ff	CODEC PLL configuration register1
CRU_CPLL_CON2	0x0028	W	0x0000017f	CODEC PLL configuration register2
CRU_CPLL_CON3	0x002c	W	0x00000008	CODEC PLL configuration register3
CRU_GPLL_CON0	0x0030	W	0x00000b01	GENERAL PLL configuration register0
CRU_GPLL_CON1	0x0034	W	0x00000251	GENERAL PLL configuration register1
CRU_GPLL_CON2	0x0038	W	0x00000128	GENERAL PLL configuration register2
CRU_GPLL_CON3	0x003c	W	0x00000008	GENERAL PLL configuration register3
CRU_NPLL_CON0	0x0040	W	0x00000b03	NEW PLL configuration register0
CRU_NPLL_CON1	0x0044	W	0x000003e7	NEW PLL configuration register1
CRU_NPLL_CON2	0x0048	W	0x000001f3	NEW PLL configuration register2
CRU_NPLL_CON3	0x004c	W	0x00000008	NEW PLL configuration register3
CRU_MODE_CON	0x0050	W	0x00000000	System work mode control register
CRU_CLKSEL0_CON	0x0060	W	0x00000031	Internal clock select and divide register0

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL1_CON	0x0064	W	0x0000b109	Internal clock select and divide register1
CRU_CLKSEL2_CON	0x0068	W	0x00000020	Internal clock select and divide register2
CRU_CLKSEL3_CON	0x006c	W	0x00000200	Internal clock select and divide register3
CRU_CLKSEL4_CON	0x0070	W	0x00000300	Internal clock select and divide register4
CRU_CLKSEL5_CON	0x0074	W	0x00000200	Internal clock select and divide register5
CRU_CLKSEL6_CON	0x0078	W	0x00000101	Internal clock select and divide register6
CRU_CLKSEL7_CON	0x007c	W	0x0bb8ea60	Internal clock select and divide register7
CRU_CLKSEL8_CON	0x0080	W	0x0bb8ea60	Internal clock select and divide register8
CRU_CLKSEL9_CON	0x0084	W	0x0bb8ea60	Internal clock select and divide register9
CRU_CLKSEL10_CON	0x0088	W	0x0000a101	Internal clock select and divide register10
CRU_CLKSEL11_CON	0x008c	W	0x00002780	Internal clock select and divide register11
CRU_CLKSEL12_CON	0x0090	W	0x00008080	Internal clock select and divide register12
CRU_CLKSEL13_CON	0x0094	W	0x00000200	Internal clock select and divide register13
CRU_CLKSEL14_CON	0x0098	W	0x00000200	Internal clock select and divide register14
CRU_CLKSEL15_CON	0x009c	W	0x00000200	Internal clock select and divide register15
CRU_CLKSEL16_CON	0x00a0	W	0x00000200	Internal clock select and divide register16
CRU_CLKSEL17_CON	0x00a4	W	0x0bb8ea60	Internal clock select and divide register17
CRU_CLKSEL18_CON	0x00a8	W	0x0bb8ea60	Internal clock select and divide register18
CRU_CLKSEL19_CON	0x00ac	W	0x0bb8ea60	Internal clock select and divide register19
CRU_CLKSEL20_CON	0x00b0	W	0x0bb8ea60	Internal clock select and divide register20
CRU_CLKSEL21_CON	0x00b4	W	0x00000b00	Internal clock select and divide register21
CRU_CLKSEL22_CON	0x00b8	W	0x00000900	Internal clock select and divide register22

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL23_CON	0x00bc	W	0x001f05dc	Internal clock select and divide register23
CRU_CLKSEL24_CON	0x00c0	W	0x00001700	Internal clock select and divide register24
CRU_CLKSEL25_CON	0x00c4	W	0x00000707	Internal clock select and divide register25
CRU_CLKSEL26_CON	0x00c8	W	0x00000ec0	Internal clock select and divide register26
CRU_CLKSEL27_CON	0x00cc	W	0x00000700	Internal clock select and divide register27
CRU_CLKSEL28_CON	0x00d0	W	0x00000f03	Internal clock select and divide register28
CRU_CLKSEL29_CON	0x00d4	W	0x00000742	Internal clock select and divide register29
CRU_CLKSEL30_CON	0x00d8	W	0x00000000	Internal clock select and divide register30
CRU_CLKSEL31_CON	0x00dc	W	0x00000000	Internal clock select and divide register31
CRU_CLKSEL32_CON	0x00e0	W	0x00000101	Internal clock select and divide register32
CRU_CLKSEL33_CON	0x00e4	W	0x00000303	Internal clock select and divide register33
CRU_CLKSEL34_CON	0x00e8	W	0x00008000	Internal clock select and divide register34
CRU_CLKSEL35_CON	0x00ec	W	0x00000303	Internal clock select and divide register35
CRU_CLKSEL36_CON	0x00f0	W	0x00000000	Internal clock select and divide register36
CRU_CLKSEL37_CON	0x00f4	W	0x00001ef3	Internal clock select and divide register37
CRU_CLKSEL38_CON	0x00f8	W	0x00000303	Internal clock select and divide register38
CRU_CLKSEL39_CON	0x00fc	W	0x00000007	Internal clock select and divide register39
CRU_CLKSEL40_CON	0x0100	W	0x00000200	Internal clock select and divide register40
CRU_CLKSEL41_CON	0x0104	W	0x0bb8ea60	Internal clock select and divide register41
CRU_CLKSEL42_CON	0x0108	W	0x00000000	Internal clock select and divide register42
CRU_CLKGATE0_CON	0x0160	W	0x00000000	Internal clock gating control register0
CRU_CLKGATE1_CON	0x0164	W	0x00000000	Internal clock gating control register1

Name	Offset	Size	Reset Value	Description
CRU_CLKGATE2_CON	0x0168	W	0x00000000	Internal clock gating control register2
CRU_CLKGATE3_CON	0x016c	W	0x00000000	Internal clock gating control register3
CRU_CLKGATE4_CON	0x0170	W	0x00000000	Internal clock gating control register4
CRU_CLKGATE5_CON	0x0174	W	0x00000000	Internal clock gating control register5
CRU_CLKGATE6_CON	0x0178	W	0x00000000	Internal clock gating control register6
CRU_CLKGATE7_CON	0x017c	W	0x00000000	Internal clock gating control register7
CRU_CLKGATE8_CON	0x0180	W	0x00000000	Internal clock gating control register8
CRU_CLKGATE9_CON	0x0184	W	0x00000000	Internal clock gating control register9
CRU_CLKGATE10_CON	0x0188	W	0x00000000	Internal clock gating control register10
CRU_CLKGATE11_CON	0x018c	W	0x00000000	Internal clock gating control register11
CRU_CLKGATE12_CON	0x0190	W	0x00000000	Internal clock gating control register12
CRU_CLKGATE13_CON	0x0194	W	0x00000000	Internal clock gating control register13
CRU_CLKGATE14_CON	0x0198	W	0x00000000	Internal clock gating control register14
CRU_CLKGATE15_CON	0x019c	W	0x00000000	Internal clock gating control register15
CRU_CLKGATE16_CON	0x01a0	W	0x00000000	Internal clock gating control register16
CRU_CLKGATE17_CON	0x01a4	W	0x00000000	Internal clock gating control register17
CRU_CLKGATE18_CON	0x01a8	W	0x00000000	Internal clock gating control register18
CRU_GLB_SRST_FST_VALUE	0x01b0	W	0x00000000	The first global software reset config value
CRU_GLB_SRST_SND_VALUE	0x01b4	W	0x00000000	The second global software reset config value
CRU_SOFTRST0_CON	0x01b8	W	0x00000000	Internal software reset control register0
CRU_SOFTRST1_CON	0x01bc	W	0x00000000	Internal software reset control register1
CRU_SOFTRST2_CON	0x01c0	W	0x00000000	Internal software reset control register2

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST3_CON	0x01c4	W	0x00000000	Internal software reset control register3
CRU_SOFTRST4_CON	0x01c8	W	0x00000000	Internal software reset control register4
CRU_SOFTRST5_CON	0x01cc	W	0x00000000	Internal software reset control register5
CRU_SOFTRST6_CON	0x01d0	W	0x00000000	Internal software reset control register6
CRU_SOFTRST7_CON	0x01d4	W	0x00000000	Internal software reset control register7
CRU_SOFTRST8_CON	0x01d8	W	0x00000000	Internal software reset control register8
CRU_SOFTRST9_CON	0x01dc	W	0x00000000	Internal software reset control register9
CRU_SOFTRST10_CON	0x01e0	W	0x00000000	Internal software reset control register10
CRU_SOFTRST11_CON	0x01e4	W	0x00000000	Internal software reset control register11
CRU_MISC_CON	0x01e8	W	0x00000000	SCU control register
CRU_GLB_CNT_TH	0x01ec	W	0x00000064	global reset wait counter threshold
CRU_GLB_RST_CON	0x01f0	W	0x00000000	global reset trigger select
CRU_GLB_RST_ST	0x01f8	W	0x00000000	global reset status
CRU_SDMMC_CON0	0x0200	W	0x00000002	sdmmc control0
CRU_SDMMC_CON1	0x0204	W	0x00000000	sdmmc control1
CRU_SDIO0_CON0	0x0208	W	0x00000002	sdio0 control0
CRU_SDIO0_CON1	0x020c	W	0x00000000	sdio0 control1
CRU_SDIO1_CON0	0x0210	W	0x00000002	sdio1 control0
CRU_SDIO1_CON1	0x0214	W	0x00000000	sdio1 control1
CRU_EMMC_CON0	0x0218	W	0x00000002	emmc control0
CRU_EMMC_CON1	0x021c	W	0x00000000	emmc control1

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

2.7.2 Detail Register Description

CRU_APLL_CON0
Address: Operational Base + offset (0x0000)
ARM PLL configuration register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	WO	0x00	clkr_mask CLKR value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
23:20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:16	WO	0x0	clkod_mask Clock OD value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x0b	clkr PLL CLKR factor control NR = CLKR + 1 NR: 1-64
7:4	RO	0x0	reserved
3:0	RW	0x1	clkod PLL CLKOD factor control NO = CLKOD + 1 NO: 1, 2-16 (even only)

CRU_APLL_CON1
Address: Operational Base + offset (0x0004)
ARM PLL configuration register1

Bit	Attr	Reset Value	Description
31	RW	0x0	lock PLL lock status 1'b0: unlock 1'b1: lock
30:13	RO	0x0	reserved
12:0	RW	0x03e7	clkf PLL CLKF factor control NF = CLKF + 1 NF: 1-4096

CRU_APLL_CON2
Address: Operational Base + offset (0x0008)
ARM PLL configuration register2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x1f3	bwadj PLL loop bandwidth adjust NB = BWADJ + 1

CRU_APLL_CON3
Address: Operational Base + offset (0x000c)
ARM PLL configuration register3

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21	WO	0x0	reset_mask Reset configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
20	WO	0x0	test_mask Test configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
19	WO	0x0	ensat_mask Ensat configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
18	WO	0x0	fasten_mask Fasten configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
17	WO	0x0	power_down_mask Power down configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
16	WO	0x0	bypass_mask Bypass configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reset PLL reset control 1'b0: normal 1'b1: reset
4	RW	0x0	test PLL test control 1'b0: normal 1'b1: test mode
3	RW	0x1	ensat PLL saturation behavior enable 1'b0: disable 1'b1: enable
2	RW	0x0	fasten PLL enable fast locking circuit 1'b0: disable 1'b1: enable
1	RW	0x0	power_down PLL power down control 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
0	RW	0x0	bypass PLL bypass mode control 1'b0: no bypass 1'b1: bypass

CRU_DPLL_CON0
Address: Operational Base + offset (0x0010)
DDR PLL configuration register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	WO	0x00	clkr_mask CLKR value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
23:20	RO	0x0	reserved
19:16	WO	0x0	clkod_mask Clock OD value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x0b	clkr PLL CLKR factor control NR = CLKR + 1 NR: 1-64
7:4	RO	0x0	reserved
3:0	RW	0x3	clkod PLL CLKOD factor control NO = CLKOD + 1 NO: 1, 2-16 (even only)

CRU_DPLL_CON1
Address: Operational Base + offset (0x0014)
DDR PLL configuration register1

Bit	Attr	Reset Value	Description
31	RW	0x0	lock PLL lock status 1'b0: unlock 1'b1: lock
30:13	RO	0x0	reserved
12:0	RW	0x031f	clkf PLL CLKF factor control NF = CLKF + 1 NF: 1-4096

CRU_DPLL_CON2
Address: Operational Base + offset (0x0018)
DDR PLL configuration register2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x18f	bwadj PLL loop bandwidth adjust NB = BWADJ + 1

CRU_DPLL_CON3
Address: Operational Base + offset (0x001c)
DDR PLL configuration register3

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	WO	0x0	reset_mask Reset configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
20	WO	0x0	test_mask Test configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
19	WO	0x0	ensat_mask Ensat configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
18	WO	0x0	fasten_mask Fasten configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
17	WO	0x0	power_down_mask Power down configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
16	WO	0x0	bypass_mask Bypass configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reset PLL reset control 1'b0: normal 1'b1: reset

Bit	Attr	Reset Value	Description
4	RW	0x0	test PLL test control 1'b0: normal 1'b1: test mode
3	RW	0x1	ensat PLL saturation behavior enable 1'b0: disable 1'b1: enable
2	RW	0x0	fasten PLL enable fast locking circuit 1'b0: disable 1'b1: enable
1	RW	0x0	power_down PLL power down control 1'b0: no power down 1'b1: power down
0	RW	0x0	bypass PLL bypass mode control 1'b0: no bypass 1'b1: bypass

CRU_CPLL_CON0
Address: Operational Base + offset (0x0020)
CODEC PLL configuration register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	WO	0x00	clkr_mask CLKR value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
23:20	RO	0x0	reserved
19:16	WO	0x0	clkod_mask Clock OD value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x0b	clkr PLL CLKR factor control NR = CLKR + 1 NR: 1-64
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x3	clkod PLL CLKOD factor control NO = CLKOD + 1 NO: 1, 2-16 (even only)

CRU_CPLL_CON1
Address: Operational Base + offset (0x0024)
CODEC PLL configuration register1

Bit	Attr	Reset Value	Description
31	RW	0x0	lock PLL lock status 1'b0: unlock 1'b1: lock
30:13	RO	0x0	reserved
12:0	RW	0x02ff	clkf PLL CLKF factor control NF = CLKF + 1 NF: 1-4096

CRU_CPLL_CON2
Address: Operational Base + offset (0x0028)
CODEC PLL configuration register2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x17f	bwadj PLL loop bandwidth adjust NB = BWADJ + 1

CRU_CPLL_CON3
Address: Operational Base + offset (0x002c)
CODEC PLL configuration register3

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	WO	0x0	reset_mask Reset configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
20	WO	0x0	test_mask Test configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
19	WO	0x0	ensat_mask Ensat configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
18	WO	0x0	fasten_mask Fasten configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
17	WO	0x0	power_down_mask Power down configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
16	WO	0x0	bypass_mask Bypass configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reset PLL reset control 1'b0: normal 1'b1: reset
4	RW	0x0	test PLL test control 1'b0: normal 1'b1: test mode
3	RW	0x1	ensat PLL saturation behavior enable 1'b0: disable 1'b1: enable
2	RW	0x0	fasten PLL enable fast locking circuit 1'b0: disable 1'b1: enable
1	RW	0x0	power_down PLL power down control 1'b0: no power down 1'b1: power down

Bit	Attr	Reset Value	Description
0	RW	0x0	bypass PLL bypass mode control 1'b0: no bypass 1'b1: bypass

CRU_GPLL_CON0
Address: Operational Base + offset (0x0030)
GENERAL PLL configuration register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	WO	0x00	clkr_mask CLKR value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
23:20	RO	0x0	reserved
19:16	WO	0x0	clkod_mask Clock OD value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x0b	clkr PLL CLKR factor control NR = CLKR + 1 NR: 1-64
7:4	RO	0x0	reserved
3:0	RW	0x1	clkod PLL CLKOD factor control NO = CLKOD + 1 NO: 1, 2-16 (even only)

CRU_GPLL_CON1
Address: Operational Base + offset (0x0034)
GENERAL PLL configuration register1

Bit	Attr	Reset Value	Description
31	RW	0x0	lock PLL lock status 1'b0: unlock 1'b1: lock
30:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:0	RW	0x0251	clkf PLL CLKF factor control NF = CLKF + 1 NF: 1-4096

CRU_GPLL_CON2
Address: Operational Base + offset (0x0038)
GENERAL PLL configuration register2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x128	bwadj PLL loop bandwidth adjust NB = BWADJ + 1

CRU_GPLL_CON3
Address: Operational Base + offset (0x003c)
GENERAL PLL configuration register3

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	WO	0x0	reset_mask Reset configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
20	WO	0x0	test_mask Test configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
19	WO	0x0	ensat_mask Ensat configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
18	WO	0x0	fasten_mask Fasten configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
17	WO	0x0	power_down_mask Power down configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
16	WO	0x0	bypass_mask Bypass configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reset PLL reset control 1'b0: normal 1'b1: reset
4	RW	0x0	test PLL test control 1'b0: normal 1'b1: test mode
3	RW	0x1	ensat PLL saturation behavior enable 1'b0: disable 1'b1: enable
2	RW	0x0	fasten PLL enable fast locking circuit 1'b0: disable 1'b1: enable
1	RW	0x0	power_down PLL power down control 1'b0: no power down 1'b1: power down
0	RW	0x0	bypass PLL bypass mode control 1'b0: no bypass 1'b1: bypass

CRU_NPLL_CON0
Address: Operational Base + offset (0x0040)
NEW PLL configuration register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	WO	0x00	clkr_mask CLKR value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
23:20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:16	WO	0x0	clkod_mask Clock OD value write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x0b	clkr PLL CLKR factor control NR = CLKR + 1 NR: 1-64
7:4	RO	0x0	reserved
3:0	RW	0x3	clkod PLL CLKOD factor control NO = CLKOD + 1 NO: 1, 2-16 (even only)

CRU_NPLL_CON1
Address: Operational Base + offset (0x0044)
NEW PLL configuration register1

Bit	Attr	Reset Value	Description
31	RW	0x0	lock PLL lock status 1'b0: unlock 1'b1: lock
30:13	RO	0x0	reserved
12:0	RW	0x03e7	clkf PLL CLKF factor control NF = CLKF + 1 NF: 1-4096

CRU_NPLL_CON2
Address: Operational Base + offset (0x0048)
NEW PLL configuration register2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x1f3	bwadj PLL loop bandwidth adjust NB = BWADJ + 1

CRU_NPLL_CON3
Address: Operational Base + offset (0x004c)
NEW PLL configuration register3

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	WO	0x0	reset_mask Reset configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
20	WO	0x0	test_mask Test configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
19	WO	0x0	ensat_mask Ensat configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
18	WO	0x0	fasten_mask Fasten configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
17	WO	0x0	power_down_mask Power down configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
16	WO	0x0	bypass_mask Bypass configuration write mask. When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5	RW	0x0	reset PLL reset control 1'b0: normal 1'b1: reset
4	RW	0x0	test PLL test control 1'b0: normal 1'b1: test mode
3	RW	0x1	ensat PLL saturation behavior enable 1'b0: disable 1'b1: enable
2	RW	0x0	fasten PLL enable fast locking circuit 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
1	RW	0x0	power_down PLL power down control 1'b0: no power down 1'b1: power down
0	RW	0x0	bypass PLL bypass mode control 1'b0: no bypass 1'b1: bypass

CRU_MODE_CON
Address: Operational Base + offset (0x0050)
System work mode control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	npll_work_mode NEW PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
13:12	RW	0x0	gpll_work_mode GENERAL PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
11:10	RO	0x0	reserved
9:8	RW	0x0	cpll_work_mode CODEC PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
7:6	RO	0x0	reserved
5:4	RW	0x0	dpll_work_mode DDR PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	apll_work_mode ARM PLL work mode select 2'b00: Slow mode, clock from external 24MHz OSC (default) 2'b01: Normal mode, clock from PLL output 2'b10: Deep slow mode, clock from external 32.768kHz

CRU_CLKSEL0_CON
Address: Operational Base + offset (0x0060)
Internal clock select and divide register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	core_clk_pll_sel CORE clock pll source selection 1'b0: select ARM PLL 1'b1: select GENERAL PLL
14:13	RO	0x0	reserved
12:8	RW	0x00	a17_core_div_con Control A17 core clock divider frequency clk_core=clk_src/(div_con+1)
7:4	RW	0x3	aclk_core_mp_div_con Control core MP AXI clock divider frequency clk=clk_src/(div_con+1)
3:0	RW	0x1	aclk_core_m0_div_con Control core M0 AXI clock divider frequency clk=clk_src/(div_con+1)

CRU_CLKSEL1_CON
Address: Operational Base + offset (0x0064)
Internal clock select and divide register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	bus_aclk_pll_sel pd_bus axi clock pll source selection 1'b0: select CODEC PLL 1'b1: select GENERAL PLL

Bit	Attr	Reset Value	Description
14:12	RW	0x3	pd_bus_pclk_div_con Control pd_bus APB clock divider frequency clk=clk_src/(div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x1	pd_bus_hclk_div_con Control pd_bus AHB clock divider frequency 2'b00: aclk_bus:hclk_bus = 1:1 2'b01: aclk_bus:hclk_bus = 2:1 2'b11: aclk_bus:hclk_bus = 4:1
7:3	RW	0x01	pd_bus_aclk_div_con Control pd_bus aclk divider frequency clk=clk_src/(div_con+1)
2:0	RW	0x1	pd_bus_clk_div_con1 Control pd_bus AXI clock divider1 frequency clk=clk_src/(div_con+1)

CRU_CLKSEL2_CON
Address: Operational Base + offset (0x0068)
Internal clock select and divide register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	testout_div_con test out clk divider frequency clk_testout=testout_clk_src/(testout_div_con+1)
7:6	RO	0x0	reserved
5:0	RW	0x20	tsadc_div_con Control tsadc divider frequency clk_tsadc=tsadc_clk_src/(tsadc_div_con+1)

CRU_CLKSEL3_CON
Address: Operational Base + offset (0x006c)
Internal clock select and divide register3

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

Bit	Attr	Reset Value	Description
9:8	RW	0x2	uart4_clk_sel Control UART4 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	uart4_div_con Control UART4 divider frequency clk_uart0=uart_clk_src/(uart0_div_con+1)

CRU_CLKSEL4_CON
Address: Operational Base + offset (0x0070)
Internal clock select and divide register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2s_pll_sel Control I2S PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14:13	RO	0x0	reserved
12	RW	0x0	i2s0_outclk_sel Control I2S clock work frequency selection 1'b0: select clk_i2s 1'b1: select 12MHz
11:10	RO	0x0	reserved
9:8	RW	0x3	i2s0_clk_sel Control I2S clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select clock from IO input 2'b11: select 12MHz from osc input
7	RO	0x0	reserved
6:0	RW	0x00	i2s0_pll_div_con Control I2S PLL output divider freuency i2s1_div_clk=i2s1_div_src/(i2s1_pll_div_con+1)

CRU_CLKSEL5_CON
Address: Operational Base + offset (0x0074)
Internal clock select and divide register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	spdif_pll_sel Control SPDIF PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14:10	RO	0x0	reserved
9:8	RW	0x2	spdif_clk_sel Control SPDIF clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 12MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	spdif_pll_div_con Control SPDIF PLL output divider freuency spdif_div_clk=spdif_div_src/(spdif_pll_div_con+1)

CRU_CLKSEL6_CON
Address: Operational Base + offset (0x0078)
Internal clock select and divide register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	isp_jpeg_pll_sel Control ISP jpeg PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
13:8	RW	0x01	isp_jpeg_div_con Control isp jpeg divider freuency jpeg_div_clk=jpeg_div_src/(isp_jpeg_div_con+1)
7:6	RW	0x0	isp_pll_sel Control ISP PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
5:0	RW	0x01	isp_div_con Control isp divider freuency isp_div_clk=isp_div_src/(isp_pll_div_con+1)

CRU_CLKSEL7_CON
Address: Operational Base + offset (0x007c)
Internal clock select and divide register7

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart4_frac_factor Control uart4 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL8_CON
Address: Operational Base + offset (0x0080)
Internal clock select and divide register8

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	i2s0_frac_factor Control I2S fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL9_CON
Address: Operational Base + offset (0x0084)
Internal clock select and divide register9

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	spdif_frac_factor Control SPDIF fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL10_CON
Address: Operational Base + offset (0x0088)
Internal clock select and divide register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	peri_pll_sel Control peripheral clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x2	peri_pclk_div_con Control the divider ratio between aclk_periph and pclk_periph 2'b00: aclk_periph:pclk_periph = 1:1 2'b01: aclk_periph:pclk_periph = 2:1 2'b10: aclk_periph:pclk_periph = 4:1 2'b11: aclk_periph:pclk_periph = 8:1
11:10	RO	0x0	reserved
9:8	RW	0x1	peri_hclk_div_con Control the divider ratio between aclk_periph and hclk_periph 2'b00: aclk_periph:hclk_periph = 1:1 2'b01: aclk_periph:hclk_periph = 2:1 2'b10: aclk_periph:hclk_periph = 4:1
7:5	RO	0x0	reserved
4:0	RW	0x01	peri_aclk_div_con Control periphral clock divider frequency aclk_periph=periph_clk_src/(peri_aclk_div_con+1)

CRU_CLKSEL11_CON
Address: Operational Base + offset (0x008c)
Internal clock select and divide register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:8	RW	0x27	reserved
7:6	RW	0x2	mmc0_pll_sel Control mmc0 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select 24MHz
5:0	RW	0x00	mmc0_div_con Control SDMMC0 divider frequency clk_sdmmc0=general_pll_clk/(mmc0_div_con+1)

CRU_CLKSEL12_CON
Address: Operational Base + offset (0x0090)
Internal clock select and divide register12

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	emmc_pll_sel Control emmc clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select 24MHz
13:8	RW	0x00	emmc_div_con Control EMMC divider frequency clk_emmc=general_pll_clk/(emmc_div_con+1)
7:6	RW	0x2	sdio0_pll_sel Control sdio0 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select 24MHz
5:0	RW	0x00	sdio0_div_con Control SDIO0 divider frequency clk_sdio=general_pll_clk/(sdio_div_con+1)

CRU_CLKSEL13_CON
Address: Operational Base + offset (0x0094)
Internal clock select and divide register13

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	uart_pll_sel Control UART1~4 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14:13	RW	0x0	uart0_src_sel UART0 clock source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select 480M USBPHY clock 2'b11: select new pll clock

Bit	Attr	Reset Value	Description
12:11	RW	0x0	usbphy_480m_sel USBPHY 480M clock source selection 2'b00: select HOST0 USB pll clock 2'b01: select HOST1 USB pll clock 2'b10: select OTG USB pll clock
10	RO	0x0	reserved
9:8	RW	0x2	uart0_clk_sel Control UART0 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	uart0_div_con Control UART0 divider frequency clk_uart0=uart_clk_src/(uart0_div_con+1)

CRU_CLKSEL14_CON
Address: Operational Base + offset (0x0098)
Internal clock select and divide register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	uart1_clk_sel Control UART1 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	uart1_div_con Control UART1 divider frequency clk_uart1=uart_clk_src/(uart1_div_con+1)

CRU_CLKSEL15_CON
Address: Operational Base + offset (0x009c)
Internal clock select and divide register15

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	uart2_clk_sel Control UART2 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	uart2_div_con Control UART2 divider frequency clk_uart2=uart_clk_src/(uart2_div_con+1)

CRU_CLKSEL16_CON
Address: Operational Base + offset (0x00a0)
Internal clock select and divide register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9:8	RW	0x2	uart3_clk_sel Control UART3 clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 24MHz from osc inpu
7	RO	0x0	reserved
6:0	RW	0x00	uart3_div_con Control UART3 divider frequency clk_uart3=uart_clk_src/(uart3_div_con+1)

CRU_CLKSEL17_CON
Address: Operational Base + offset (0x00a4)
Internal clock select and divide register17

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart0_frac_factor Control UART0 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL18_CON
Address: Operational Base + offset (0x00a8)
Internal clock select and divide register18

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart1_frac_factor Control UART1 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL19_CON
Address: Operational Base + offset (0x00ac)
Internal clock select and divide register19

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart2_frac_factor Control UART2 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL20_CON
Address: Operational Base + offset (0x00b0)
Internal clock select and divide register20

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	uart3_frac_factor Control UART3 fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL21_CON
Address: Operational Base + offset (0x00b4)
Internal clock select and divide register21

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x0b	mac_div_con Control EMAC divider frequency clk_mac_ref=mac_clk_src/(mac_div_con+1)
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	rmii_extclk_sel Control RMII external clock selection 1'b0: select internal divider clock 1'b1: select external input clock
3:2	RO	0x0	reserved
1:0	RW	0x0	mac_pll_sel Control EMAC clock PLL source selection 2'b00: select new pll clock 2'b01: select codec pll clock 2'b10: select general pll clock

CRU_CLKSEL22_CON
Address: Operational Base + offset (0x00b8)
Internal clock select and divide register22

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x09	hsadc_div_con Control HSADC divider frequency clk_hsadc=hsadc_clk_src/(hsadc_div_con+1)
7	RW	0x0	hsadc_inv_sel Control HSADC inverter clock 1'b0: select buffer output 1'b1: select inverter output
6:5	RO	0x0	reserved
4	RW	0x0	hsadc_clk_sel Control HSADC clock work frequency selection 1'b0: select divider ouput from pll divider 1'b1: select external input clock
3:2	RO	0x0	reserved
1	RW	0x0	wifi_pll_sel Control wifi clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
0	RW	0x0	hsadc_pll_sel Control HSADC clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock

CRU_CLKSEL23_CON

Address: Operational Base + offset (0x00bc)
Internal clock select and divide register23

Bit	Attr	Reset Value	Description
31:0	RW	0x001f05dc	wifi_frac_factor Control wifi fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL24_CON
Address: Operational Base + offset (0x00c0)
Internal clock select and divide register24

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x17	saradc_div_con Control SARADC clock divider frequency clk_saradc=24MHz/(saradc_div_con+1)
7:0	RO	0x0	reserved

CRU_CLKSEL25_CON
Address: Operational Base + offset (0x00c4)
Internal clock select and divide register25

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	spi1_pll_sel Control spi1 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14:8	RW	0x07	spi1_div_con Control SPI1 clock divider frequency clk_spi1=general_pll_clk/(spi1_div_con+1)
7	RW	0x0	spi0_pll_sel Control spi0 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
6:0	RW	0x07	spi0_div_con Control SPI0 clock divider frequency clk_spi0=general_pll_clk/(spi0_div_con+1)

CRU_CLKSEL26_CON
Address: Operational Base + offset (0x00c8)
Internal clock select and divide register26

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	cif_clk_out_sel CIF clock output selection 1'b0: select PLL divout 1'b1: select 24MHz
14	RO	0x0	reserved
13:9	RW	0x07	cif_clk_div_con cif clock divider frequency clk=clk_src/(div_con+1)
8	RW	0x0	cif_clk_pll_sel CIF clock pll source selection 1'b0: select codec PLL 1'b1: select general PLL
7:6	RW	0x3	crypto_div_con crypto clock divider frequency clk=clk_src/(div_con+1)
5:3	RO	0x0	reserved
2	RW	0x0	ddr_clk_pll_sel DDR clock pll source selection 1'b0: select DDR PLL 1'b1: select GENERAL PLL
1:0	RW	0x0	ddr_div_con Control DDR divider frequency 2'b00: clk_ddr_src:clk_ddrphy = 1:1 2'b01: clk_ddr_src:clk_ddrphy = 2:1 2'b11: clk_ddr_src:clk_ddrphy = 4:1

CRU_CLKSEL27_CON
Address: Operational Base + offset (0x00cc)
Internal clock select and divide register27

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

Bit	Attr	Reset Value	Description
15:8	RW	0x07	lcdc0_div_con Control LCDC0 clock divider frequency clk_lcdc0=lcdc0_clk_src/(lcdc0_div_con+1)
7:2	RO	0x0	reserved
1:0	RW	0x0	lcdc0_pll_sel Control LCDC0 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock

CRU_CLKSEL28_CON
Address: Operational Base + offset (0x00d0)
Internal clock select and divide register28

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	edp_24m_sel eDP 24M clock source selection 1'b00: select 27M clock 1'b01: select 24M clock
14:13	RO	0x0	reserved
12:8	RW	0x0f	hclk_vio_div_con VIO AHB clock divider frequency clk=clk_src/(div_con+1)
7:6	RW	0x0	edp_pll_sel eDP clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
5:0	RW	0x03	edp_div_con eDP clock divider frequency clk=clk_src/(div_con+1)

CRU_CLKSEL29_CON
Address: Operational Base + offset (0x00d4)
Internal clock select and divide register29

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x07	lcdc1_div_con Control LCDC1 clock divider frequency clk_lcdc1=lcdc1_clk_src/(lcdc1_div_con+1)
7:6	RW	0x1	lcdc1_pll_sel Control LCDC1 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
5	RO	0x0	reserved
4	RW	0x0	cif_clkin_inv_sel CIF clkin invert selection 1'b0: normal 1'b1: invert
3	RW	0x0	isp_clkin_inv_sel ISP clkin invert selection 1'b0: normal 1'b1: invert
2	RW	0x0	reserved
1:0	RW	0x2	reserved

CRU_CLKSEL30_CON
Address: Operational Base + offset (0x00d8)
Internal clock select and divide register30

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	rga_core_clk_pll_sel rga func clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
13	RO	0x0	reserved
12:8	RW	0x00	rga_core_clk_div_con rga func clock divider frequency clk_rga_func = clk_rga_func_src/(rga_core_clk_div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	rga_aclk_pll_sel Control rga AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
5	RO	0x0	reserved
4:0	RW	0x00	rga_aclk_div_con Control rga AXI clock divider frequency aclk_lcdc1=lcdc1_aclk_src/(rga_aclk_div_con+1)

CRU_CLKSEL31_CON
Address: Operational Base + offset (0x00dc)
Internal clock select and divide register31

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	vio1_aclk_pll_sel Control VIO1 AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
13	RO	0x0	reserved
12:8	RW	0x00	vio1_aclk_div_con Control VIO1 AXI clock divider frequency aclk_vio1=vio1_aclk_src/(vio1_aclk_div_con+1)
7:6	RW	0x0	vio0_aclk_pll_sel Control VIO0 AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
5	RO	0x0	reserved
4:0	RW	0x00	vio0_aclk_div_con Control VIO0 AXI clock divider frequency aclk_vio0=vio0_aclk_src/(vio0_aclk_div_con+1)

CRU_CLKSEL32_CON
Address: Operational Base + offset (0x00e0)
Internal clock select and divide register32

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	vdpu_aclk_pll_sel Control VDPU AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
13	RO	0x0	reserved
12:8	RW	0x01	vdpu_aclk_div_con Control VDPU AXI clock divider frequency aclk_vdpu=vdpu_aclk_src/(vdpu_aclk_div_con+1)
7:6	RW	0x0	vepu_aclk_pll_sel Control VEPU AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock
5	RO	0x0	reserved
4:0	RW	0x01	vepu_aclk_div_con Control VEPU AXI clock divider frequency aclk_vepu=vepu_aclk_src/(vepu_aclk_div_con+1)

CRU_CLKSEL33_CON
Address: Operational Base + offset (0x00e4)
Internal clock select and divide register33

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask When HIGH, enable the writing corresponding bit When LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x03	alive_pclk_div_con alive apb clock divider frequency alive_pclk =alive_pclk_src/(alive_pclk_div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x03	pmu_pclk_div_con pmu apb clock divider frequency pmu_pclk =pmu_pclk_src/(pmu_pclk_div_con+1)

CRU_CLKSEL34_CON
Address: Operational Base + offset (0x00e8)
Internal clock select and divide register34

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x2	sdio1_pll_sel Control sdio1 clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select 24MHz
13:8	RW	0x00	sdio1_div_con Control SDIO1 divider frequency clk_sdio=general_pll_clk/(sdio_div_con+1)
7:6	RW	0x0	gpu_aclk_pll_sel Control GPU AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select usbphy pll 480M clock 2'b11: select new pll clock
5	RO	0x0	reserved
4:0	RW	0x00	gpu_aclk_div_con Control GPU AXI clock divider frequency aclk_gpu=gpu_aclk_src/(gpu_aclk_div_con+1)

CRU_CLKSEL35_CON
Address: Operational Base + offset (0x00ec)
Internal clock select and divide register35

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	tspout_clk_pll_sel Control tspout clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock 2'b11: select 27MHz IO input
13	RO	0x0	reserved
12:8	RW	0x03	tspout_clk_div_con Control tspout clock divider frequency clk=clk_src/(clk_div_con+1)

Bit	Attr	Reset Value	Description
7:6	RW	0x0	tsp_clk_pll_sel Control tsp clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
5	RO	0x0	reserved
4:0	RW	0x03	tsp_clk_div_con Control tsp clock divider frequency clk=clk_src/(clk_div_con+1)

CRU_CLKSEL36_CON
Address: Operational Base + offset (0x00f0)
Internal clock select and divide register36

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x0	clk_core3_div_con Control clk_core3 clock divider frequency clk=clk_src/(clk_div_con+1)
11	RO	0x0	reserved
10:8	RW	0x0	clk_core2_div_con Control clk_core2 clock divider frequency clk=clk_src/(clk_div_con+1)
7	RO	0x0	reserved
6:4	RW	0x0	clk_core1_div_con Control clk_core1 clock divider frequency clk=clk_src/(clk_div_con+1)
3	RO	0x0	reserved
2:0	RW	0x0	clk_core0_div_con Control clk_core0 clock divider frequency clk=clk_src/(clk_div_con+1)

CRU_CLKSEL37_CON
Address: Operational Base + offset (0x00f4)
Internal clock select and divide register37

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:9	RW	0x0f	pclk_core_dbg_div_con Control core debg APB bus clock divider frequency clk=clk_src/(clk_div_con+1)
8:4	RW	0x0f	atclk_core_div_con Control core ATB BUS clock divider frequency clk=clk_src/(clk_div_con+1)
3	RO	0x0	reserved
2:0	RW	0x3	clk_l2ram_div_con Control clk_l2ram clock divider frequency clk=clk_src/(clk_div_con+1)

CRU_CLKSEL38_CON
Address: Operational Base + offset (0x00f8)
Internal clock select and divide register38

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	nandc1_clk_pll_sel Control nandc1 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
14:13	RO	0x0	reserved
12:8	RW	0x03	nandc1_clk_div_con Control nandc1 clock divider frequency clk_nandc=nandc_clk_src/(nandc_clk_div_con+1)
7	RW	0x0	nandc0_clk_pll_sel Control nandc0 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
6:5	RO	0x0	reserved
4:0	RW	0x03	nandc0_clk_div_con Control nandc0 clock divider frequency clk_nandc=nandc_clk_src/(nandc_clk_div_con+1)

CRU_CLKSEL39_CON

Address: Operational Base + offset (0x00fc)
Internal clock select and divide register39

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	aclk_hevc_pll_sel HEVC AXI clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
13	RO	0x0	reserved
12:8	RW	0x00	aclk_hevc_div_con HEVC AXI clock divider frequency clk=clk_src/(clk_div_con+1)
7	RW	0x0	spi2_pll_sel Control spi2 clock PLL source selection 1'b0: select codec pll clock 1'b1: select general pll clock
6:0	RW	0x07	spi2_div_con Control SPI2 clock divider frequency clk=clk_src/(div_con+1)

CRU_CLKSEL40_CON
Address: Operational Base + offset (0x0100)
Internal clock select and divide register40

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13:12	RW	0x0	hclk_hevc_div_con HEVC AHB clock divider frequency clk=clk_src/(clk_div_con+1)
11:10	RO	0x0	reserved
9:8	RW	0x2	spdif_8ch_clk_sel Control SPDIF 8ch clock work frequency selection 2'b00: select divider ouput from pll divider 2'b01: select divider ouput from fraction divider 2'b10: select 12MHz from osc inpu
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	spdif_8ch_pll_div_con Control SPDIF 8ch PLL output divider freuency spdif_div_clk=spdif_div_src/(spdif_pll_div_con+1)

CRU_CLKSEL41_CON
Address: Operational Base + offset (0x0104)
Internal clock select and divide register41

Bit	Attr	Reset Value	Description
31:0	RW	0x0bb8ea60	spdif_8ch_frac_factor Control SPDIF 8ch fraction divider frequency High 16-bit for numerator Low 16-bit for denominator

CRU_CLKSEL42_CON
Address: Operational Base + offset (0x0108)
Internal clock select and divide register42

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_hevc_core_pll_sel HEVC CORE clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
13	RO	0x0	reserved
12:8	RW	0x00	clk_hevc_core_div_con HEVC CORE clock divider frequency clk=clk_src/(clk_div_con+1)
7:6	RW	0x0	clk_hevc_cabac_pll_sel HEVC CABAC clock PLL source selection 2'b00: select codec pll clock 2'b01: select general pll clock 2'b10: select new pll clock
5	RO	0x0	reserved
4:0	RW	0x00	clk_hevc_cabac_div_con HEVC CABAC clock divider frequency clk=clk_src/(clk_div_con+1)

CRU_CLKGATE0_CON

Address: Operational Base + offset (0x0160)
Internal clock gating control register0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_acc_efuse_gate_en acc efuse clock disable. When HIGH, disable clock
11	RW	0x0	pd_bus_cpll_clk_gate_en pd_bus clock CPLL path clock disable. When HIGH, disable clock
10	RW	0x0	pd_bus_gpll_clk_gate_en pd_bus clock GPLL path clock disable. When HIGH, disable clock
9	RW	0x0	ddr_gpll_clk_gate_en DDR clock GPLL path clock disable. When HIGH, disable clock
8	RW	0x0	ddr_dpll_clk_gate_en DDR clock DPLL path clock disable. When HIGH, disable clock
7	RW	0x0	aclk_bus_2pmu_gate_en pd_bus AXI clock to pd_pmu clock disable. When HIGH, disable clock
6	RO	0x0	reserved
5	RW	0x0	pclk_bus_gate_en pd_bus APB clock(pclk_cpu_pre) disable. When HIGH, disable clock
4	RW	0x0	hclk_bus_gate_en pd_bus AHB clock disable. When HIGH, disable clock
3	RW	0x0	aclk_bus_gate_en pd_bus AXI clock disable. When HIGH, disable clock
2	RW	0x0	core_gpll_clk_gate_en CORE clock GPLL path clock disable. When HIGH, disable clock
1	RW	0x0	core_apll_clk_gate_en CORE clock APLL path clock disable. When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE1_CON

Address: Operational Base + offset (0x0164)
Internal clock gating control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart3_frac_src_gate_en UART3 fraction divider source clock disable. When HIGH, disable clock
14	RW	0x0	clk_uart3_src_gate_en UART3 source clock disable. When HIGH, disable clock
13	RW	0x0	clk_uart2_frac_src_gate_en UART2 fraction divider source clock disable. When HIGH, disable clock
12	RW	0x0	clk_uart2_src_gate_en UART2 source clock disable. When HIGH, disable clock
11	RW	0x0	clk_uart1_frac_src_gate_en UART1 fraction divider source clock disable. When HIGH, disable clock
10	RW	0x0	clk_uart1_src_gate_en UART1 source clock disable. When HIGH, disable clock
9	RW	0x0	clk_uart0_frac_src_gate_en UART0 fraction divider source clock disable. When HIGH, disable clock
8	RW	0x0	clk_uart0_src_gate_en UART0 source clock disable. When HIGH, disable clock
7:6	RO	0x0	reserved
5	RW	0x0	clk_timer5_gate_en Timer5 clock(clk_timer5) disable. When HIGH, disable clock
4	RW	0x0	clk_timer4_gate_en Timer4 clock(clk_timer4) disable. When HIGH, disable clock
3	RW	0x0	clk_timer3_gate_en Timer3 clock(clk_timer3) disable. When HIGH, disable clock
2	RW	0x0	clk_timer2_gate_en Timer2 clock(clk_timer2) disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	clk_timer1_gate_en Timer1 clock(clk_timer1) disable. When HIGH, disable clock
0	RW	0x0	clk_timer0_gate_en Timer0 clock(clk_timer0) disable. When HIGH, disable clock

CRU_CLKGATE2_CON
Address: Operational Base + offset (0x0168)
Internal clock gating control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	clk_uart4_frac_src_gate_en UART4 fraction divider source clock disable. When HIGH, disable clock
12	RW	0x0	clk_uar4_src_gate_en UART4 source clock disable. When HIGH, disable clock
11	RW	0x0	clk_spi2_src_gate_en SPI2 source clock disable. When HIGH, disable clock
10	RW	0x0	clk_spi1_src_gate_en SPI1 source clock disable. When HIGH, disable clock
9	RW	0x0	clk_spi0_src_gate_en SPI0 source clock disable. When HIGH, disable clock
8	RW	0x0	clk_saradc_src_gate_en SARADC source clock disable. When HIGH, disable clock
7	RW	0x0	clk_tsadc_src_gate_en TSADC source clock disable. When HIGH, disable clock
6	RW	0x0	clk_hsadc_src_gate_en Field0000 Abstract When HIGH, disable clock
5	RW	0x0	clk_mac_src_gate_en MAC source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RO	0x0	reserved
3	RW	0x0	pclk_periph_gate_en PERIPH system APB clock(pclk_periph) disable. When HIGH, disable clock
2	RW	0x0	hclk_periph_gate_en PERIPH system AHB clock(hclk_periph) disable. When HIGH, disable clock
1	RW	0x0	aclk_periph_gate_en PERIPH system AXI clock(aclk_periph) disable. When HIGH, disable clock
0	RW	0x0	clk_periph_src_gate_en PERIPH system source clock disable. When HIGH, disable clock

CRU_CLKGATE3_CON
Address: Operational Base + offset (0x016c)
Internal clock gating control register3

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_isp_jpeg_gate_en ISP jpeg source clock disable. When HIGH, disable clock
14	RW	0x0	clk_isp_gate_en ISP clock clock disable. When HIGH, disable clock
13	RW	0x0	clk_edp_gate_en eDP clock clock disable. When HIGH, disable clock
12	RW	0x0	clk_edp_24m_gate_en eDP 24M ref clock clock disable. When HIGH, disable clock
11	RW	0x0	aclk_vdpu_src_gate_en VDPU AXI source clock disable. When HIGH, disable clock
10	RW	0x0	hclk_vpu_gate_en VPU AHB source clock disable. When HIGH, disable clock
9	RW	0x0	aclk_vepu_src_gate_en VEPU AXI source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RO	0x0	reserved
7	RW	0x0	clk_cif_out_gate_en CIF output clock disable. When HIGH, disable clock
6	RW	0x0	reserved
5	RW	0x0	aclk_rga_src_gate_en RGA AXI souce clock disable. When HIGH, disable clock
4	RW	0x0	clk_rga_core_src_gate_en RGA func souce clock disable. When HIGH, disable clock
3	RW	0x0	dclk_lcdc1_src_gate_en LCDC1 DCLK source clock disable. When HIGH, disable clock
2	RW	0x0	aclk_lcdc1_src_gate_en LCDC1 AXI source clock disable. When HIGH, disable clock
1	RW	0x0	dclk_lcdc0_src_gate_en LCDC0 DCLK source clock disable. When HIGH, disable clock
0	RW	0x0	aclk_lcdc0_src_gate_en LCDC0 AXI source clock disable. When HIGH, disable clock

CRU_CLKGATE4_CON
Address: Operational Base + offset (0x0170)
Internal clock gating control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	testclk_gate_en Test output clock disable When HIGH, disable clock
14	RW	0x0	clk_jtag_gate_en JTAG clock disable. When HIGH, disable clock
13	RW	0x0	clk_ddrphy1_gate_en DDRPHY1 clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
12	RW	0x0	clk_ddrphy0_gate_en DDRPHY0 clock disable. When HIGH, disable clock
11	RW	0x0	clk_tspout_gate_en TSP output clock disable. When HIGH, disable clock
10	RW	0x0	clk_tsp_gate_en TSP clock disable. When HIGH, disable clock
9	RW	0x0	clk_spdif_8ch_gate_en SPDIF 8ch clock disable. When HIGH, disable clock
8	RW	0x0	clk_spdif_8ch_frac_src_gate_en SPDIF 8ch fraction divider source clock disable. When HIGH, disable clock
7	RW	0x0	clk_spdif_8ch_src_gate_en SPDIF 8ch source clock disable. When HIGH, disable clock
6	RW	0x0	clk_spdif_gate_en SPDIF clock disable. When HIGH, disable clock
5	RW	0x0	clk_spdif_frac_src_gate_en SPDIF fraction divider source clock disable. When HIGH, disable clock
4	RW	0x0	clk_spdif_src_gate_en SPDIF source clock disable. When HIGH, disable clock
3	RW	0x0	clk_i2s0_gate_en I2S clock disable. When HIGH, disable clock
2	RW	0x0	clk_i2s0_frac_src_gate_en I2S fraction divider source clock disable. When HIGH, disable clock
1	RW	0x0	clk_i2s0_src_gate_en I2S source clock disable. When HIGH, disable clock
0	RW	0x0	clk_i2s0_out_gate_en I2S output clock disable. When HIGH, disable clock

CRU_CLKGATE5_CON
Address: Operational Base + offset (0x0174)
Internal clock gating control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_mipidsi_24m_gate_en mipi dsi 24M clock disable. When HIGH, disable clock
14	RW	0x0	clk_usbphy480m_gate_en usbphy480M clock disable. When HIGH, disable clock
13	RW	0x0	ps2c_clk_gate_en PS2 controlor clock disable. When HIGH, disable clock
12	RW	0x0	hdmi_hdcp_clk_gate_en HDMI HDCP clock disable. When HIGH, disable clock
11	RW	0x0	hdmi_cec_clk_gate_en HDMI CEC clock disable. When HIGH, disable clock
10	RW	0x0	clk_pvtm_gpu_gate_en pd_gpu PVTM clock disable. When HIGH, disable clock
9	RW	0x0	clk_pvtm_core_gate_en pd_core PVTM clock disable. When HIGH, disable clock
8	RW	0x0	pclk_pmu_gate_en pd_pmu APB bus clock disable. When HIGH, disable clock
7	RW	0x0	clk_gpu_gate_en gpu clock disable. When HIGH, disable clock
6	RW	0x0	clk_nandc1_gate_en nandc1 clock disable. When HIGH, disable clock
5	RW	0x0	clk_nandc0_gate_en nandc0 clock disable. When HIGH, disable clock
4	RW	0x0	clk_crypto_gate_en crypto clock disable. When HIGH, disable clock
3	RW	0x0	clk_mac_refout_gate_en MAC ref output clock clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
2	RW	0x0	clk_mac_ref_gate_en MAC ref clock clock disable. When HIGH, disable clock
1	RW	0x0	clk_mac_tx_gate_en MAC tx clock clock disable. When HIGH, disable clock
0	RW	0x0	clk_mac_rx_gate_en MAC rx clock clock disable. When HIGH, disable clock

CRU_CLKGATE6_CON
Address: Operational Base + offset (0x0178)
Internal clock gating control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_i2c4_gate_en I2C4 APB clock disable. When HIGH, disable clock
14	RW	0x0	pclk_i2c3_gate_en I2C3 APB clock disable. When HIGH, disable clock
13	RW	0x0	pclk_i2c1_gate_en I2C1 APB clock disable. When HIGH, disable clock
12	RW	0x0	pclk_uart_exp_gate_en UART_exp APB clock disable. When HIGH, disable clock
11	RW	0x0	pclk_uart_gps_gate_en UART_gps APB clock disable. When HIGH, disable clock
10	RO	0x0	reserved
9	RW	0x0	pclk_uart_bb_gate_en UART_bb APB clock disable. When HIGH, disable clock
8	RW	0x0	pclk_uart_bt_gate_en UART_bt APB clock disable. When HIGH, disable clock
7	RW	0x0	pclk_ps2c0_gate_en PS2C0 APB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_spi2_gate_en SPI2 APB clock disable. When HIGH, disable clock
5	RW	0x0	pclk_spi1_gate_en SPI1 APB clock disable. When HIGH, disable clock
4	RW	0x0	pclk_spi0_gate_en SPI0 APB clock disable. When HIGH, disable clock
3	RW	0x0	aclk_dmac_peri_gate_en DMAC peri AXI clock disable. When HIGH, disable clock
2	RW	0x0	aclk_peri_axi_matrix_gate_en Peripheral matrix axi clock disable. When HIGH, disable clock
1	RW	0x0	pclk_peri_axi_matrix_gate_en Peripheral matrix apb clock disable. When HIGH, disable clock
0	RW	0x0	hclk_peri_matrix_gate_en Peripheral matrix ahb clock disable. When HIGH, disable clock

CRU_CLKGATE7_CON
Address: Operational Base + offset (0x017c)
Internal clock gating control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_nand1_gate_en NAND1 AHB clock disable. When HIGH, disable clock
14	RW	0x0	hclk_nand0_gate_en NAND0 AHB clock disable. When HIGH, disable clock
13	RW	0x0	hclk_mmc_peri_gate_en arbiter in peri_ahb_mmc module AHB clock disable. When HIGH, disable clock
12	RW	0x0	hclk_emem_peri_gate_en arbiter in peri_ahb_emem module AHB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	aclk_peri_niu_gate_en NIU in peripheral power domain AXI clock disable. When HIGH, disable clock
10	RW	0x0	hclk_peri_ahb_arbi_gate_en AHB arbiter in peripheral power domain AHB clock disable. When HIGH, disable clock
9	RW	0x0	hclk_usb_peri_gate_en USB arbiter AHB clock disable. When HIGH, disable clock
8	RW	0x0	reserved
7	RW	0x0	hclk_host1_gate_en HOST1 AHB clock disable. Field0000 Description
6	RW	0x0	hclk_host0_gate_en HOST0 AHB clock disable. Field0000 Description
5	RW	0x0	pmu_hclk_otg0_gate_en USB OTG PMU AHB clock disable. When HIGH, disable clock
4	RW	0x0	hclk_otg0_gate_en USB OTG AHB clock disable. When HIGH, disable clock
3	RW	0x0	pclk_sim_gate_en SIM APB clock disable. When HIGH, disable clock
2	RW	0x0	pclk_tsadc_gate_en TSADC APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_saradc_gate_en SARADC APB clock disable. When HIGH, disable clock
0	RW	0x0	pclk_i2c5_gate_en I2C5 APB clock disable. When HIGH, disable clock

CRU_CLKGATE8_CON
Address: Operational Base + offset (0x0180)
Internal clock gating control register8

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

Bit	Attr	Reset Value	Description
15:13	RO	0x0	reserved
12	RW	0x0	aclk_peri_mmu_gate_en PERI_MMU aclk clock disable. When HIGH, disable clock
11	RW	0x0	clk_27m_tsp_gate_en 27M_TSP clock disable. When HIGH, disable clock
10	RW	0x0	clk_hsadc_1_tsp_gate_en HSADC_1_TSP clock disable. When HIGH, disable clock
9	RW	0x0	clk_hsadc_0_tsp_gate_en HSADC_0_TSP clock disable. When HIGH, disable clock
8	RW	0x0	hclk_tsp_gate_en TSP AHB clock disable. When HIGH, disable clock
7	RW	0x0	hclk_hsadc_gate_en HSADC AHB clock disable. When HIGH, disable clock
6	RW	0x0	hclk_emmc_gate_en EMMC AHB clock disable. When HIGH, disable clock
5	RW	0x0	hclk_sdio1_gate_en SDIO1 AHB clock disable. When HIGH, disable clock
4	RW	0x0	hclk_sdio0_gate_en SDIO0 AHB clock disable. When HIGH, disable clock
3	RW	0x0	hclk_sdmmc_gate_en SDMMC AHB clock disable. When HIGH, disable clock
2	RW	0x0	hclk_gps_gate_en GPS hclk clock disable. When HIGH, disable clock
1	RW	0x0	pclk_gmac_gate_en GMAC pclk clock disable. When HIGH, disable clock
0	RW	0x0	aclk_gmac_gate_en GMAC aclk clock disable. When HIGH, disable clock

CRU_CLKGATE9_CON
Address: Operational Base + offset (0x0184)
Internal clock gating control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:2	RO	0x0	reserved
1	RW	0x0	hclk_video_clock_en VIDEO AHB clock disable. When HIGH, disable clock
0	RW	0x0	aclk_video_gate_en VIDEO AXI clock disable. When HIGH, disable clock

CRU_CLKGATE10_CON
Address: Operational Base + offset (0x0188)
Internal clock gating control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_publ0_gate_en DDR0 PUBL apb clock disable When HIGH, disable clock
14	RW	0x0	pclk_ddrupctl0_gate_en DDRUPCTL0 apb clock disable When HIGH, disable clock
13	RW	0x0	aclk_strc_sys_gate_en aclk_strc_sys (CPU Structure system) clock disable. When HIGH, disable clock
12	RW	0x0	aclk_dmac_bus_gate_en DMAC_BUS aclk clock disable. When HIGH, disable clock
11	RW	0x0	hclk_spdif_8ch_gate_en hclk_spdif_8ch clock disable. When HIGH, disable clock
10	RW	0x0	hclk_spdif_gate_en hclk_spdif clock disable. When HIGH, disable clock
9	RW	0x0	hclk_rom_gate_en hclk_rom clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	hclk_i2s_8ch_gate_en hclk_i2s_8ch AHB clock disable. When HIGH, disable clock
7	RW	0x0	clk_intmem2_gate_en intmem2 clock disable. When HIGH, disable clock
6	RW	0x0	clk_intmem1_gate_en intmem1 clock disable. When HIGH, disable clock
5	RW	0x0	clk_intmem0_gate_en intmem0 clock disable. When HIGH, disable clock
4	RW	0x0	aclk_intmem_gate_en intmem axi clock disable. When HIGH, disable clock
3	RW	0x0	pclk_i2c2_gate_en pclk_i2c2 disable. When HIGH, disable clock
2	RW	0x0	pclk_i2c0_gate_en pclk_i2c0 disable. When HIGH, disable clock
1	RW	0x0	pclk_timer_gate_en pclk_timer disable. When HIGH, disable clock
0	RW	0x0	pclk_pwm_gate_en pclk_pwm disable. When HIGH, disable clock

CRU_CLKGATE11_CON
Address: Operational Base + offset (0x018c)
Internal clock gating control register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	pclk_rkpwm_gate_en pclk_rkpwm disable. When HIGH, disable clock
10	RW	0x0	pclk_efuse_256_gate_en EFUSE256 APB clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	pclk_uart_dbg_gate_en UART_DBG APB clock disable. When HIGH, disable clock
8	RW	0x0	aclk_ccp_gate_en CCP aclk clock disable. When HIGH, disable clock
7	RW	0x0	hclk_crypto_gate_en CRYPTO sclk clock disable. When HIGH, disable clock
6	RW	0x0	aclk_crypto_gate_en CRYPTO mclk clock disable. When HIGH, disable clock
5	RW	0x0	nclk_ddrupctl1_gate_en DDR Controller PHY clock disable. When HIGH, disable clock
4	RW	0x0	nclk_ddrupctl0_gate_en DDR Controller PHY clock disable. When HIGH, disable clock
3	RW	0x0	pclk_tzpc_gate_en TZPC APB clock disable. When HIGH, disable clock
2	RW	0x0	pclk_efuse_1024_gate_en EFUSE1024 APB clock disable. When HIGH, disable clock
1	RW	0x0	pclk_publ1_gate_en DDR1 PUBL apb clock disable When HIGH, disable clock
0	RW	0x0	pclk_ddrupctl1_gate_en DDRUPCTL1 apb clock disable When HIGH, disable clock

CRU_CLKGATE12_CON
Address: Operational Base + offset (0x0190)
Internal clock gating control register12

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	pclk_core_niu_gate_en core NIU APB bus clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
10	RW	0x0	cs_dbg_clk_gate_en coresight debug clock disable. When HIGH, disable clock
9	RW	0x0	dbg_core_clk_gate_en core debug clock disable. When HIGH, disable clock
8	RW	0x0	dbg_src_clk_gate_en Debug source clock disable. When HIGH, disable clock
7	RW	0x0	atclk_core_gate_en core ATB bus clock disable. When HIGH, disable clock
6	RW	0x0	aclk_mp_gate_en core MP AXI bus clock disable. When HIGH, disable clock
5	RW	0x0	aclk_core_m0_gate_en CORE m0 AXI bus clock disable. When HIGH, disable clock
4	RW	0x0	l2_ram_clk_gate_en L2 RAM clock disable. When HIGH, disable clock
3	RW	0x0	core3_clk_gate_en core3 clock disable. When HIGH, disable clock
2	RW	0x0	core2_clk_gate_en core2 clock disable. When HIGH, disable clock
1	RW	0x0	coer1_clk_gate_en core1 clock disable. When HIGH, disable clock
0	RW	0x0	core0_clk_gate_en core0 clock disable. When HIGH, disable clock

CRU_CLKGATE13_CON
Address: Operational Base + offset (0x0194)
Internal clock gating control register13

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

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_hevc_core_gate_en HEVC CORE clock disable. When HIGH, disable clock
14	RW	0x0	clk_hevc_cabac_gate_en HEVC cabac clock disable. When HIGH, disable clock
13	RW	0x0	aclk_hevc_gate_en HEVC AXI clock disable. When HIGH, disable clock
12	RW	0x0	clk_wifi_gate_en wifi/gps/bt 3in1 16.384M clock disable. When HIGH, disable clock
11	RW	0x0	clk_lcdc_pwm1_gate_en lcdc_pwm1 clock disable. When HIGH, disable clock
10	RW	0x0	clk_lcdc_pwm0_gate_en lcdc_pwm0 clock disable. When HIGH, disable clock
9	RW	0x0	reserved
8	RW	0x0	clk_c2c_host_gate_en C2C HOST clock disable. When HIGH, disable clock
7	RW	0x0	clk_otg_adp_gate_en OTG adp clock disable. When HIGH, disable clock
6	RW	0x0	clk_otgphy2_gate_en OTGPHY2 clock(clk_otgphy2) disable. When HIGH, disable clock
5	RW	0x0	clk_otgphy1_gate_en OTGPHY1 clock(clk_otgphy1) disable. When HIGH, disable clock
4	RW	0x0	clk_otgphy0_gate_en OTGPHY0 clock(clk_otgphy0) disable. When HIGH, disable clock
3	RW	0x0	clk_emmc_src_gate_en EMMC source clock disable. When HIGH, disable clock
2	RW	0x0	clk_sdio1_src_gate_en SDIO1 source clock disable. When HIGH, disable clock
1	RW	0x0	clk_sdio0_src_gate_en SDIO0 source clock disable. When HIGH, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	clk_mmc0_src_gate_en SDMMC0 source clock disable. When HIGH, disable clock

CRU_CLKGATE14_CON
Address: Operational Base + offset (0x0198)
Internal clock gating control register14

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	pclk_alive_niu_gate_en ALIVE_NIU pclk disable When HIGH, disable clock
11	RW	0x0	pclk_grf_gate_en GRF pclk disable When HIGH, disable clock
10:9	RO	0x0	reserved
8	RW	0x0	pclk_gpio8_gate_en GPIO8 pclk disable When HIGH, disable clock
7	RW	0x0	pclk_gpio7_gate_en GPIO7 pclk disable When HIGH, disable clock
6	RW	0x0	pclk_gpio6_gate_en GPIO6 pclk disable When HIGH, disable clock
5	RW	0x0	pclk_gpio5_gate_en GPIO5 pclk disable When HIGH, disable clock
4	RW	0x0	pclk_gpio4_gate_en GPIO4 pclk disable When HIGH, disable clock
3	RW	0x0	pclk_gpio3_gate_en GPIO3 pclk disable When HIGH, disable clock
2	RW	0x0	pclk_gpio2_gate_en GPIO2 pclk disable When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	pclk_gpio1_gate_en GPIO1 pclk disable When HIGH, disable clock
0	RO	0x0	reserved

CRU_CLKGATE15_CON
Address: Operational Base + offset (0x019c)
Internal clock gating control register15

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hclk_vip_gate_en VIP hclk disable When HIGH, disable clock
14	RW	0x0	aclk_vip_gate_en VIP aclk disable When HIGH, disable clock
13	RW	0x0	aclk_vio2_noc_gate_en VIO2_NOC aclk disable When HIGH, disable clock
12	RW	0x0	aclk_vio1_noc_gate_en VIO1_NOC aclk disable When HIGH, disable clock
11	RW	0x0	aclk_vio0_noc_gate_en VIO0_NOC aclk disable When HIGH, disable clock
10	RW	0x0	hclk_vio_noc_gate_en VIO_NOC hclk disable When HIGH, disable clock
9	RW	0x0	hclk_vio_ahb_arbi_gate_en VIO_AHB_ARBI hclk disable When HIGH, disable clock
8	RW	0x0	hclk_lcdc1_gate_en LCDC1 hclk disable When HIGH, disable clock
7	RW	0x0	aclk_lcdc1_gate_en LCDC1 aclk disable When HIGH, disable clock
6	RW	0x0	hclk_lcdc0_gate_en LCDC0 hclk disable When HIGH, disable clock

Bit	Attr	Reset Value	Description
5	RW	0x0	aclk_lcdc0_gate_en LCDC0 aclk disable When HIGH, disable clock
4	RW	0x0	aclk_lcdc_iep_gate_en LCDC_IEP aclk disable When HIGH, disable clock
3	RW	0x0	hclk_iep_gate_en IEP hclk disable When HIGH, disable clock
2	RW	0x0	aclk_iep_gate_en IEP aclk disable When HIGH, disable clock
1	RW	0x0	hclk_rga_gate_en RGA hclk disable When HIGH, disable clock
0	RW	0x0	aclk_rga_gate_en RGA aclk disable When HIGH, disable clock

CRU_CLKGATE16_CON
Address: Operational Base + offset (0x01a0)
Internal clock gating control register16

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	pclk_vio2_h2p_gate_en VIO2_H2P pclk disable When HIGH, disable clock
10	RW	0x0	hclk_vio2_h2p_gate_en VIO2_H2P hclk disable When HIGH, disable clock
9	RW	0x0	pclk_hdmi_ctrl_gate_en HDMI_CTRL pclk disable When HIGH, disable clock
8	RW	0x0	pclk_edp_ctrl_gate_en EDP_CTRL pclk disable When HIGH, disable clock
7	RW	0x0	pclk_lvds_phy_gate_en LVDS_PHY pclk disable When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	pclk_mipi_csi_gate_en MIPI_CSI pclk disable When HIGH, disable clock
5	RW	0x0	pclk_mipi_dsi1_gate_en MIPI_DSI1 pclk disable When HIGH, disable clock
4	RW	0x0	pclk_mipi_dsi0_gate_en MIPI_DSI0 pclk disable When HIGH, disable clock
3	RW	0x0	pclkin_isp_gate_en ISP pclkin disable When HIGH, disable clock
2	RW	0x0	aclk_isp_gate_en ISP aclk disable When HIGH, disable clock
1	RW	0x0	hclk_isp_gate_en ISP hclk disable When HIGH, disable clock
0	RW	0x0	pclkin_vip_gate_en VIP pclkin disable When HIGH, disable clock

CRU_CLKGATE17_CON
Address: Operational Base + offset (0x01a4)
Internal clock gating control register17

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. 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_gpio0_gate_en GPIO0 pclk disable When HIGH, disable clock
3	RO	0x0	Reserved
2	RW	0x0	pclk_pmu_noc_gate_en PMU_NOC pclk disable When HIGH, disable clock
1	RW	0x0	pclk_intmem1_gate_en INTMEM1 pclk disable When HIGH, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	pclk_pmu_gate_en PMU pclk disable When HIGH, disable clock

CRU_CLKGATE18_CON
Address: Operational Base + offset (0x01a8)
Internal clock gating control register18

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:1	RO	0x0	reserved
0	RW	0x0	aclk_gpu_gate_en GPU aclk disable When HIGH, disable clock

CRU_GLB_SRST_FST_VALUE
Address: Operational Base + offset (0x01b0)
The first global software reset config 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 If config 0xfdb9, it will generate first global software reset.

CRU_GLB_SRST_SND_VALUE
Address: Operational Base + offset (0x01b4)
The second global software reset config 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 If config 0xeca8, it will generate second global software reset.

CRU_SOFTRST0_CON
Address: Operational Base + offset (0x01b8)
Internal software reset control register0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	core3_dbg_srstn_req Core3 CPU debug software reset request. When HIGH, reset relative logic
14	RW	0x0	core2_dbg_srstn_req Core2 CPU debug software reset request. When HIGH, reset relative logic
13	RW	0x0	core1_dbg_srstn_req Core1 CPU debug software reset request. When HIGH, reset relative logic
12	RW	0x0	core0_dbg_srstn_req Core0 CPU debug software reset request. When HIGH, reset relative logic
11	RW	0x0	topdbg_srstn_req CPU top debug software reset request. When HIGH, reset relative logic
10	RW	0x0	l2c_srstn_req L2 controller software reset request. When HIGH, reset relative logic
9	RW	0x0	pd_bus_str_sys_asrstn_req PD BUS NOC AXI software reset request. When HIGH, reset relative logic
8	RW	0x0	pd_core_str_sys_asrstn_req PD CORE NOC AXI software reset request. When HIGH, reset relative logic
7	RW	0x0	core3_po_srstn_req Core3 CPU PO software reset request. When HIGH, reset relative logic
6	RW	0x0	core2_po_srstn_req Core2 CPU PO software reset request. When HIGH, reset relative logic
5	RW	0x0	core1_po_srstn_req Core1 CPU PO software reset request. When HIGH, reset relative logic
4	R/WSC	0x0	core0_po_srstn_req Core0 CPU PO software reset request. When HIGH, reset relative logic
3	RW	0x0	core3_srstn_req Core3 CPU software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	core2_srstn_req Core2 CPU software reset request. When HIGH, reset relative logic
1	RW	0x0	core1_srstn_req Core1 CPU software reset request. When HIGH, reset relative logic
0	R/WSC	0x0	core0_srstn_req Core0 CPU software reset request. When HIGH, reset relative logic

CRU_SOFTRST1_CON
Address: Operational Base + offset (0x01bc)
Internal software reset control register1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	efuse_psrstn_req EFUSE APB software reset request. When HIGH, reset relative logic
14	RW	0x0	timer5_srstn_req Timer5 software reset request. When HIGH, reset relative logic
13	RW	0x0	timer4_srstn_req Timer4 software reset request. When HIGH, reset relative logic
12	RW	0x0	timer3_srstn_req Timer3 software reset request. When HIGH, reset relative logic
11	RW	0x0	timer2_srstn_req Timer2 software reset request. When HIGH, reset relative logic
10	RW	0x0	timer1_srstn_req Timer1 software reset request. When HIGH, reset relative logic
9	RW	0x0	timer0_srstn_req Timer0 software reset request. When HIGH, reset relative logic
8	RW	0x0	spdif_srstn_req SPDIF software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
7	RW	0x0	i2s_srstn_req I2S software reset request. When HIGH, reset relative logic
6	RW	0x0	timer_psrstn_req Timer APB software reset request. When HIGH, reset relative logic
5	RW	0x0	spdif_8ch_srstn_req SPDIF 8ch software reset request. When HIGH, reset relative logic
4	RW	0x0	rom_srstn_req ROM software reset request. When HIGH, reset relative logic
3	RW	0x0	intmem_srstn_req Internal memory software reset request. When HIGH, reset relative logic
2	RW	0x0	dma1_srstn_req DMA1 software reset request. When HIGH, reset relative logic
1	RW	0x0	efuse_256bit_psrstn_req 256bit EFUSE APB software reset request. When HIGH, reset relative logic
0	RW	0x0	pd_bus_ahb_arbitor_srstn_req pd_bus ahb arbitor software reset request. pd_cpu AHB arbitor reset control When HIGH, reset relative logic

CRU_SOFTRST2_CON
Address: Operational Base + offset (0x01c0)
Internal software reset control register2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	i2c5_srstn_req I2C5 software reset request. When HIGH, reset relative logic
14	RW	0x0	i2c4_srstn_req I2C4 software reset request. When HIGH, reset relative logic
13	RW	0x0	i2c3_srstn_req I2C3 software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	i2c2_srstn_req I2C2 software reset request. When HIGH, reset relative logic
11	RW	0x0	i2c1_srstn_req I2C1 software reset request. When HIGH, reset relative logic
10	RW	0x0	i2c0_srstn_req I2C0 software reset request. When HIGH, reset relative logic
9	RO	0x0	reserved
8	RW	0x0	gpio8_srstn_req GPIO8 software reset request. When HIGH, reset relative logic
7	RW	0x0	gpio7_srstn_req GPIO7 software reset request. When HIGH, reset relative logic
6	RW	0x0	gpio6_srstn_req GPIO6 software reset request. When HIGH, reset relative logic
5	RW	0x0	gpio5_srstn_req GPIO5 software reset request. When HIGH, reset relative logic
4	RW	0x0	gpio4_srstn_req GPIO4 software reset request. When HIGH, reset relative logic
3	RW	0x0	gpio3_srstn_req GPIO3 software reset request. When HIGH, reset relative logic
2	RW	0x0	gpio2_srstn_req GPIO2 software reset request. When HIGH, reset relative logic
1	RW	0x0	gpio1_srstn_req GPIO1 software reset request. When HIGH, reset relative logic
0	RW	0x0	gpio0_srstn_req GPIO0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST3_CON
Address: Operational Base + offset (0x01c4)
Internal software reset control register3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	usb_peri_srstn_req USB PERIPH software reset request. When HIGH, reset relative logic
14	RW	0x0	emem_peri_srstn_req EMEM ahb bus software reset request. When HIGH, reset relative logic
13	RW	0x0	reserved
12	RW	0x0	periph_niu_srstn_req PERIPH NIU software reset request. When HIGH, reset relative logic
11	RW	0x0	periphsys_psrstn_req PERIPH APB software reset request. pd_peri bus matrix apb softreset When HIGH, reset relative logic
10	RW	0x0	periphsys_hsrstn_req PERIPH AHB software reset request. pd_peri bus matrix ahb softreset When HIGH, reset relative logic
9	RW	0x0	periphsys_asrstn_req PERIPH AXI software reset request. pd_peri bus matrix axi softreset When HIGH, reset relative logic
8	RW	0x0	pmu_srstn_req PMU software reset request. When HIGH, reset relative logic
7	RW	0x0	grf_srstn_req GRF software reset request. When HIGH, reset relative logic
6	RW	0x0	pmu_psrstn_req PMU APB bus software reset request. When HIGH, reset relative logic
5	RW	0x0	tpiu_atsrstn_req TPIU ATB software reset request. When HIGH, reset relative logic
4	RW	0x0	dap_sys_srstn_req DAP system software reset request. When HIGH, reset relative logic
3	RW	0x0	dap_srstn_req DAP software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	periph_mmu_srstn_req PERIPH MMU software reset request. When HIGH, reset relative logic
1	RW	0x0	mmc_peri_srstn_req pd_peri mmc AHB bus software reset request. emmc, sdio, sdmmc AHB arbitor reset control When HIGH, reset relative logic
0	RW	0x0	pwm_srstn_req PWM software reset request. When HIGH, reset relative logic

CRU_SOFTRST4_CON
Address: Operational Base + offset (0x01c8)
Internal software reset control register4

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	nandc1_srstn_req NANDC1 software reset request. When HIGH, reset relative logic
13	RW	0x0	nandc0_srstn_req NANDC0 software reset request. When HIGH, reset relative logic
12	RW	0x0	hsadc_srstn_req HSADC software reset request. When HIGH, reset relative logic
11:9	RW	0x0	reserved
8	RW	0x0	usb_host0_srstn_req USB HOST0 AHB software reset request. When HIGH, reset relative logic
7	RW	0x0	ccp_srstn_req CCP software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	rk_pwm_srstn_req RK_PWM software reset request. When HIGH, reset relative logic
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	gps_srstn_req GPS software reset request. When HIGH, reset relative logic
2	RW	0x0	mac_srstn_req MAC software reset request. When HIGH, reset relative logic
1	RO	0x0	reserved
0	RW	0x0	dma2_srstn_req DMA2 software reset request. When HIGH, reset relative logic

CRU_SOFTRST5_CON
Address: Operational Base + offset (0x01cc)
Internal software reset control register5

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	RW	0x0	pd_pmu_niu_psrstn_req pd_pmu niu APB software reset request. When HIGH, reset relative logic
9	RW	0x0	pd_pmu_intmem_psrstn_req pd_pmu internal memory apb software reset request. When HIGH, reset relative logic
8	RW	0x0	pd_alive_niu_psrstn_req pd_alive niu APB software reset request. When HIGH, reset relative logic
7	RW	0x0	saradc_srstn_req SARADC software reset request. When HIGH, reset relative logic
6	RO	0x0	reserved
5	RW	0x0	spi2_srstn_req SPI2 software reset request. When HIGH, reset relative logic
4	RW	0x0	spi1_srstn_req SPI1 software reset request. When HIGH, reset relative logic
3	RW	0x0	spi0_srstn_req SPI0 software reset request. When HIGH, reset relative logic
2:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	tzpc_srstn_req TZPC software reset request. When HIGH, reset relative logic

CRU_SOFTRST6_CON
Address: Operational Base + offset (0x01d0)
Internal software reset control register6

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	edp_srstn_req eDP software reset request. When HIGH, reset relative logic
14	RW	0x0	isp_srstn_req ISP software reset request. When HIGH, reset relative logic
13	RW	0x0	rga_hsrstn_req RGA AHB software reset request. When HIGH, reset relative logic
12	RW	0x0	rga_asrstn_req RGA AXI software reset request. When HIGH, reset relative logic
11	RW	0x0	iep_hsrstn_req IEP AHB software reset request. When HIGH, reset relative logic
10	RW	0x0	iep_asrstn_req IEP AXI software reset request. When HIGH, reset relative logic
9	RW	0x0	rga_core_srstn_req RGA func software reset request. When HIGH, reset relative logic
8	RW	0x0	vip_srstn_req VIP software reset request. IEP ISP VOP's NIU software reset. When HIGH, reset relative logic
7	RW	0x0	vio1_niu_asrstn_req VIO1 NIU AXI software reset request. IEP ISP VOP's NIU software reset. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
6	RW	0x0	lcdc0_dsrstn_req LCDC0 DCLK software reset request. When HIGH, reset relative logic
5	RW	0x0	lcdc0_hsrstn_req LCDC0 AHB software reset request. When HIGH, reset relative logic
4	RW	0x0	lcdc0_asrstn_req LCDC0 AXI software reset request. When HIGH, reset relative logic
3	RW	0x0	vio_niu_hsrstn_req VIO NIU AHB software reset request. When HIGH, reset relative logic
2	RW	0x0	vio0_niu_asrstn_req VIO0 NIU AXI software reset request. IEP ISP VOP's NIU software reset. When HIGH, reset relative logic
1	RW	0x0	rga_niu_asrstn_req RGA NIU AXI software reset request. IEP ISP VOP's NIU software reset. When HIGH, reset relative logic
0	RW	0x0	vio_arbi_hsrstn_req VIO arbitor AHB software reset request. When HIGH, reset relative logic

CRU_SOFTRST7_CON
Address: Operational Base + offset (0x01d4)
Internal software reset control register7

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	gpu_pvtm_srstn_req gpu pvtm software reset request. When HIGH, reset relative logic
12	RW	0x0	core_pvtm_srstn_req core pvtm software reset request. When HIGH, reset relative logic
11:10	RO	0x0	reserved
9	RW	0x0	hdmi_srstn_req HDMI software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
8	RW	0x0	gpu_srstn_req GPU core software reset request. When HIGH, reset relative logic
7	RW	0x0	lvds_con_srstn_req LVDS controller software reset request. When HIGH, reset relative logic
6	RW	0x0	lvds_phy_psrstn_req LVDS PHY APB software reset request. When HIGH, reset relative logic
5	RW	0x0	mipicsi_psrstn_req MIPi CSI APB software reset request. When HIGH, reset relative logic
4	RW	0x0	mipidsi1_psrstn_req MIPi DSI1 APB software reset request. When HIGH, reset relative logic
3	RW	0x0	mipidsi0_psrstn_req MIPi DSI0 APB software reset request. When HIGH, reset relative logic
2	RW	0x0	vio_h2p_hsrstn_req VIO ahb to apb bridge AHB software reset request. When HIGH, reset relative logic
1	RW	0x0	vcodec_hsrstn_req VCODEC AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	vcodec_asrstn_req VCODEC AXI software reset request. When HIGH, reset relative logic

CRU_SOFTRST8_CON
Address: Operational Base + offset (0x01d8)
Internal software reset control register8

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	acc_efuse_srstn_req acc efuse software reset request. When HIGH, reset relative logic
13	RW	0x0	usb_adp_srstn_req OTG adp clock software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	usbhost1c_srstn_req USBHOST1 controller software reset request. When HIGH, reset relative logic
11	RW	0x0	usbhost1phy_srstn_req USBHOST1 PHY software reset request. When HIGH, reset relative logic
10	RW	0x0	usbhost1_hsrstn_req USBHOST1 AHB BUS software reset request. When HIGH, reset relative logic
9	RW	0x0	usbhost0c_srstn_req USBHOST0 controller software reset request. When HIGH, reset relative logic
8	RW	0x0	usbhost0phy_srstn_req USBHOST0 PHY software reset request. When HIGH, reset relative logic
7	RW	0x0	usbhost0_hsrstn_req USBHOST0 AHB BUS software reset request. When HIGH, reset relative logic
6	RW	0x0	usbotgc_srstn_req USBOTG controller software reset request. When HIGH, reset relative logic
5	RW	0x0	usbotgphy_srstn_req USBOTG PHY software reset request. When HIGH, reset relative logic
4	RW	0x0	usbotg_hsrstn_req USBOTG AHB BUS software reset request. When HIGH, reset relative logic
3	RW	0x0	emmc_srstn_req EMMC software reset request. When HIGH, reset relative logic
2	RW	0x0	sdio1_srstn_req SDIO1 software reset request. When HIGH, reset relative logic
1	RW	0x0	sdio0_srstn_req SDIO0 software reset request. When HIGH, reset relative logic
0	RW	0x0	mmc0_srstn_req SDMMC0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST9_CON
Address: Operational Base + offset (0x01dc)
Internal software reset control register9

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	tsadc_psrstn_req TSADC APB software reset request. When HIGH, reset relative logic
14:11	RO	0x0	reserved
10	RW	0x0	hevc_srstn_req HEVC software reset request. When HIGH, reset relative logic
9	RW	0x0	rga_h2p_brg_srstn_req RGA AHB to APB bridge software reset request. When HIGH, reset relative logic
8	RW	0x0	vio1_h2p_brg_srstn_req VIO1 AHB to APB bridge software reset request. When HIGH, reset relative logic
7	RW	0x0	vio0_h2p_brg_srstn_req VIO0 AHB to APB bridge software reset request. When HIGH, reset relative logic
6	RW	0x0	lcdcpwm1_srstn_req lcdc_pwm1 software reset request. When HIGH, reset relative logic
5	RW	0x0	lcdcpwm0_srstn_req lcdc_pwm0 software reset request. When HIGH, reset relative logic
4	RW	0x0	gic_srstn_req GIC software reset request. When HIGH, reset relative logic
3	RW	0x0	pd_core_mp_axi_srstn_req pd_croe periph axi software reset request. When HIGH, reset relative logic
2	RW	0x0	pd_core_apb_noc_srstn_req pd_core APB software reset request. When HIGH, reset relative logic
1	RW	0x0	pd_core_ahb_noc_srstn_req PD_CORE AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	coresight_srstn_req coresight software reset request. When HIGH, reset relative logic

CRU_SOFTRST10_CON
Address: Operational Base + offset (0x01e0)

Internal software reset control register10

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	c2c_host_srstn_req c2c host clk domain software reset request. When HIGH, reset relative logic
14	RW	0x0	crypto_srstn_req crypto working clk domain software reset request. When HIGH, reset relative logic
13:12	RO	0x0	reserved
11	RW	0x0	ddrmsch1_srstn_req DDR1 memory scheduler software reset request. When HIGH, reset relative logic
10	RW	0x0	ddrmsch0_srstn_req DDR0 memory scheduler software reset request. When HIGH, reset relative logic
9	RW	0x0	ddrphy1_ctl_srstn_req DDR1 PUB software reset request. When HIGH, reset relative logic
8	RW	0x0	ddrctrl1_psrstn_req DDR controller1 APB software reset request. When HIGH, reset relative logic
7	RW	0x0	ddrctrl1_srstn_req DDR controller1 software reset request. When HIGH, reset relative logic
6	RW	0x0	ddrphy1_psrstn_req DDR PHY1 APB software reset request. When HIGH, reset relative logic
5	RW	0x0	ddrphy1_srstn_req DDR PHY1 software reset request. When HIGH, reset relative logic
4	RW	0x0	ddrphy0_ctl_srstn_req DDR0 PUB software reset request. When HIGH, reset relative logic
3	RW	0x0	ddrctrl0_psrstn_req DDR controller0 APB software reset request. When HIGH, reset relative logic
2	RW	0x0	ddrctrl0_srstn_req DDR controller0 software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
1	RW	0x0	ddrphy0_psrstn_req DDR PHY0 APB software reset request. When HIGH, reset relative logic
0	RW	0x0	ddrphy0_srstn_req DDR PHY0 software reset request. When HIGH, reset relative logic

CRU_SOFTRST11_CON
Address: Operational Base + offset (0x01e4)
Internal software reset control register11

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	tsp_27m_srstn_req TSP 27M lock domain software reset request. When HIGH, reset relative logic
14	RW	0x0	tsp_clkin1_srstn_req TSP clockin1 software reset request. When HIGH, reset relative logic
13	RW	0x0	tsp_clkin0_srstn_req TSP clockin 0 software reset request. When HIGH, reset relative logic
12	RW	0x0	tsp_srstn_req tsp software reset request. When HIGH, reset relative logic
11	RW	0x0	ps2c_srstn_req ps2 controlor software reset request. When HIGH, reset relative logic
10	RW	0x0	simc_srstn_req cim card controlor software reset request. When HIGH, reset relative logic
9:8	RO	0x0	reserved
7	RW	0x0	uart4_srstn_req UART4 software reset request. When HIGH, reset relative logic
6	RW	0x0	uart3_srstn_req UART3 software reset request. When HIGH, reset relative logic
5	RW	0x0	uart2_srstn_req UART2 software reset request. When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
4	RW	0x0	uart1_srstn_req UART1 software reset request. When HIGH, reset relative logic
3	RW	0x0	uart0_srstn_req UART0 software reset request. When HIGH, reset relative logic
2	RW	0x0	lcdc1_dsrstn_req LCDC1 DCLK software reset request. When HIGH, reset relative logic
1	RW	0x0	lcdc1_hsrstn_req LCDC1 AHB software reset request. When HIGH, reset relative logic
0	RW	0x0	lcdc1_asrstn_req LCDC1 AXI software reset request. When HIGH, reset relative logic

CRU_MISC_CON
Address: Operational Base + offset (0x01e8)
SCU control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11:8	RW	0x0	testclk_sel Output clock selection for test 4'b0000: aclk_periph 4'b0001: clk_core 4'b0010: aclk_vio0 4'b0011: clk_ddrphy 4'b0100: aclk_vcodec 4'b0101: aclk_gpu 4'b0110: clk_rga_core 4'b0111: aclk_cpu 4'b1000: 24MHz 4'b1001: 27MHz 4'b1010: 32KHz 4'b1011: clk_wifi(16.368MHz) 4'b1100: dclk_lcdc0 4'b1101: dclk_lcdc1 4'b1110: clk_isp_jpeg 4'b1111: clk_isp

Bit	Attr	Reset Value	Description
7:0	RO	0x0	reserved

CRU_GLB_CNT_TH
Address: Operational Base + offset (0x01ec)
global reset wait counter threshold

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x064	glb_rst_cnt_th Global soft reset counter threshold

CRU_GLB_RST_CON
Address: Operational Base + offset (0x01f0)
global reset trigger select

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x0	pmu_glb_srst_ctrl pmu reset by global soft reset select 2'b00: pmu reset by first global soft reset 2'b01: pmu reset by second global soft reset 2'b10: pmu not reset by any global soft reset
1	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
0	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

CRU_GLB_RST_ST
Address: Operational Base + offset (0x01f8)
global reset status

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	W1C	0x0	snd_glb_wdt_rst_st second global watch_dog triggered reset flag 1'b0: last hot reset is not second global watch_dog triggered reset 1'b1: last hot reset is second global watch_dog triggered reset

Bit	Attr	Reset Value	Description
4	W1C	0x0	fst_glb_wdt_rst_st first global watch_dog triggered reset flag 1'b0: last hot reset is not first global watch_dog triggered reset 1'b1: last hot reset is first global watch_dog triggered reset
3	W1C	0x0	snd_glb_tsadc_rst_st second global TSADC triggered reset flag 1'b0: last hot reset is not second global TSADC triggered reset 1'b1: last hot reset is second global TSADC triggered reset
2	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
1	W1C	0x0	snd_glb_rst_st second global rst flag 1'b0: last hot reset is not second global rst 1'b1: last hot reset is second global rst
0	W1C	0x0	fst_glb_rst_st first global rst flag 1'b0: last hot reset is not first global rst 1'b1: last hot reset is first global rst

CRU_SDMMC_CON0*
Address: Operational Base + offset (0x0200)
sdmmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	sdmmc_drv_sel sdmmc drive select sdmmc drive select
10:3	WO	0x00	sdmmc_drv_delaynum sdmmc drive delay number sdmmc drive delay number
2:1	WO	0x1	sdmmc_drv_degree sdmmc drive degree sdmmc drive degree
0	WO	0x0	sdmmc_init_state sdmmc initial state sdmmc initial state

CRU_SDMMC_CON1*
Address: Operational Base + offset (0x0204)
sdmmc control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	sdmmc_sample_sel sdmmc sample select sdmmc sample select
9:2	WO	0x00	sdmmc_sample_delaynum sdmmc sample delay number sdmmc sample delay number
1:0	WO	0x0	sdmmc_sample_degree sdmmc sample degree sdmmc sample degree

CRU_SDIO0_CON0*
Address: Operational Base + offset (0x0208)
sdio0 control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	sdio0_drv_sel sdio0 drive select sdio0 drive select
10:3	WO	0x00	sdio0_drv_delaynum sdio0 drive delay number sdio0 drive delay number
2:1	WO	0x1	sdio0_drv_degree sdio0 drive degree sdio0 drive degree
0	WO	0x0	sdio0_init_state sdio0 initial state sdio0 initial state

CRU_SDIO0_CON1*
Address: Operational Base + offset (0x020c)

sdio0 control1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	sdio0_sample_sel sdio0 sample select sdio0 sample select
9:2	WO	0x00	sdio0_sample_delaynum sdio0 sample delay number sdio0 sample delay number
1:0	WO	0x0	sdio0_sample_degree sdio0 sample degree sdio0 sample degree

CRU_SDIO1_CON0*
Address: Operational Base + offset (0x0210)
sdio1 control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	sdio1_drv_sel sdio1 drive select sdio1 drive select
10:3	WO	0x00	sdio1_drv_delaynum sdio1 drive delay number sdio1 drive delay number
2:1	WO	0x1	sdio1_drv_degree sdio1 drive degree sdio1 drive degree
0	WO	0x0	sdio1_init_state sdio1 initial state sdio1 initial state

CRU_SDIO1_CON1*
Address: Operational Base + offset (0x0214)
sdio1 control1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	sdio1_sample_sel sdio1 sample select sdio1 sample select
9:2	WO	0x00	sdio1_sample_delaynum sdio1 sample delay number sdio1 sample delay number
1:0	WO	0x0	sdio1_sample_degree sdio1 sample degree sdio1 sample degree

CRU_EMMC_CON0*
Address: Operational Base + offset (0x0218)
emmc control0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	WO	0x0	emmc_drv_sel emmc drive select emmc drive select
10:3	WO	0x00	emmc_drv_delaynum emmc drive delay number emmc drive delay number
2:1	WO	0x1	emmc_drv_degree emmc drive degree emmc drive degree
0	WO	0x0	emmc_init_state emmc initial state emmc initial state

CRU_EMMC_CON1*
Address: Operational Base + offset (0x021c)
emmc control1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask write mask. When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:11	RO	0x0	reserved
10	WO	0x0	emmc_sample_sel emmc sample select emmc sample select
9:2	WO	0x00	emmc_sample_delaynum emmc sample delay number emmc sample delay number
1:0	WO	0x0	emmc_sample_degree emmc sample degree emmc sample degree

**Notes: CRU_SDMMC_CON0/1, CRU_SDIO1_CON0/1, CRU_SDIO0_CON0/1, CRU_EMMC_CON0/1, detail description please refer to chapter1 of part3 Mobile Storage Host Controller 1.6.10.*

2.8 Timing Diagram

Power on reset timing is shown as follow:

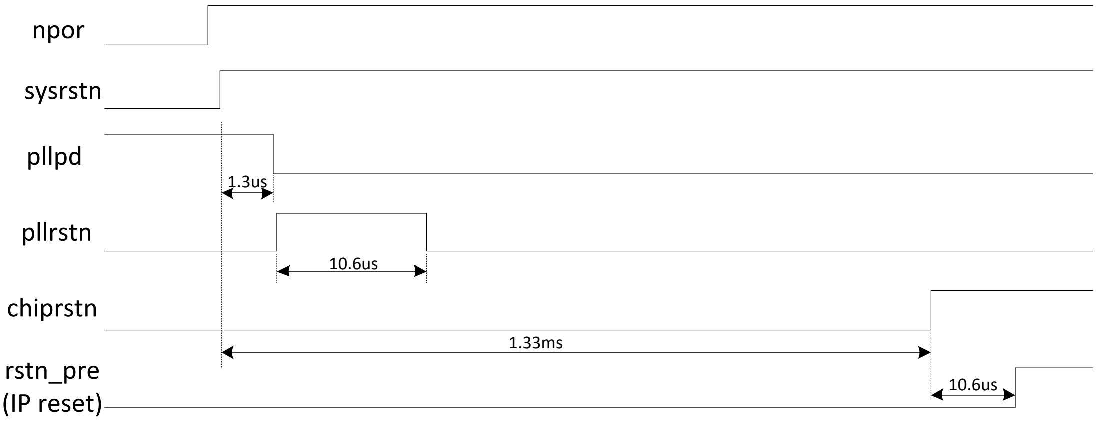


Fig. 2-8 Chip Power On Reset Timing Diagram

Npor is hardware reset signal from out-chip and power-off mode wakeup reset from PMU, which is filtered glitch to obtain signal sysrstn. To make PLLs work normally, the power down signal (pllpd) must be high when reset, and maintains high for more then 1us when sysrstn de-active. Then PLL reset signals (pllrstn) are asserted for about 10.6us, and PLLs start to lock when pllrstn de-assert, and consume about 1330us to lock. So the system will wait about 1330us, then de-active reset signal chiprstn. The signal chiprstn is used to generate output clocks in CRU. After CRU start output clocks, the system waits again for 256 cycles (10.7us) to de-active signal rstn_pre, which is used to generate power on reset of all IP.

2.9 Application Notes

2.9.1 PLL usage

PLL output frequency configuration

The output frequency Fout is related to the input frequency Fin by:

Fout = ((Fin /NR) * NF)/ NO

Fout is clock output of PLL, and Fin is clock input of PLL from external oscillators (24MHz).

Another, other factors such as NF, NR, NO can be configured by programming CRU_APLL_CONi, CRU_DPLL_CONi, CRU_CPLL_CONi and CRU_GPLL_CONi registers (i=0,1,2), and their value will affect Fout as follows.

(1) CLKR: A 6-bit bus that selects the values 1-64 for the reference divider (NR)
NR = CLKR[5:0] + 1
Example:
/1 pgm 000000
/4 pgm 000011
/8 pgm 000111

(2) CLKF: A 13-bit bus that selects the values 1-4096 for the PLL multiplication factor (NF)
NF = CLKF[12:0] + 1
Example:
X1 pgm 0000000000000
X2 pgm 0000000000001
X4096 pgm 0111111111111

(3) CLKOD: A 4-bit bus that selects the value1,2-16(even only) for the PLL post VCO divider (NO)
NO = CLKOD[3:0] + 1
Example:
/1 pgm 0000
/2 pgm 0001
/4 pgm 0011
/8 pgm 0111

BWADJ: A 12-bit bus that selects the values 1-4096 for the bandwidth divider (NB)
NB = BWADJ[11:0] + 1
Example:
/1 pgm 000000000000
/4 pgm 000000000011
/8 pgm 000000000111

The recommended setting of NB: NB = NF / 2.

PLL setting consideration

Optimization of the PLL settings for jitter < +/- 2.5% of the output period/sq rt(NO) require running the VCO at maximum frequency and dividing down using the NO divider to get the required Fout, i.e. maximum NO.
Optimization for minimum power (Fvco/1100MHz * 3.3 mA) requires setting the VCO frequency at the minimum frequency and using the lowest NO setting.
These two values, minimum jitter or minimum power will determine your choice of settings.
A larger value of input divider NR gives a longer lock time, and higher long term as well as period jitter. It is better to use a lower value of NR where possible.

2.9.2 PLL frequency change method

When the PLL settings are changed, it has to reset PLL by programming registers CRU_APLL_CON3, CRU_DPLL_CON3, CRU_CPLL_CON3, CRU_GPLL_CON3, CRU_NPLL_CON3, and reserve at least 5us after valid settings, referring to the following figure.

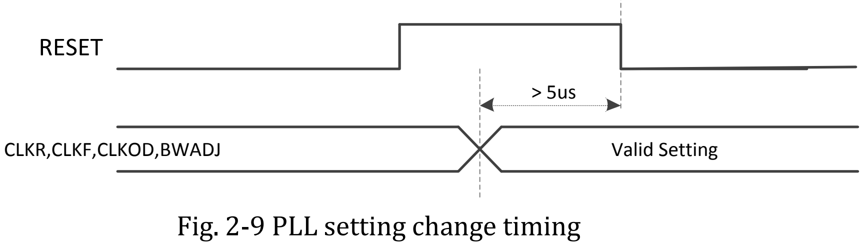


Fig. 2-9 PLL setting change timing

Before set some factors such as NR/NF/NO/BS to change PLL output frequency, you must change chip from normal to slow mode by programming CRU_MODE_CON. Then until PLL is lock state by checking GRF_SOC_STATUS0[8:5] register, or after delay about (NR * 500) / Fin, you can change PLL into normal mode.

2.9.3 Fractional divider usage

To get specific frequency, clocks of I2S, SPDIF, UART, HSADC can be generated by fractional divider. Generally you must set that denominator is 20 times larger than numerator to generate precise clock frequency. So the fractional divider applies only to generate low frequency clock like I2S, UART and HSADC.
All the fractional divider has auto-gating control. When fractional divider is not selected, the divider clock is gated. So fractional divider must be selected before changing configuration.

2.9.4 Global software reset

Two global software resets are designed in this 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.
TSADC, WDT and PMU also can trigger glb_srstn_1 or glb_srstn_2.
CRU_GLB_RST_CON controls which global soft-reset will be triggered.
After global reset, the reset trigger source can be check in CRU_GLB_RST_ST.
glb_srstn_1 resets almost all chip logic except PMU_SYS_REG0~3, which can be used to store something when reset.
glb_srstn_2 resets almost all chip logic except PMU_SYS_REG0~3, GRF and GPIOs.

2.9.5 Pre-shift for test

Pre-shift registers is designed in this chip for flexible test.
The key configuration registers can be shifted with a initial value in testmode. The pre-shift registers including 191 bit pre_shift_test_reg.
The following table describes the pre-shift registers of the design.

Name	Bit number	Default value	Description
armpll_clkr	pre_shift_test_reg[5:0]	6'd0	armpll_clkr control
armpll_clkf	pre_shift_test_reg[18:6]	13'd199	armpll_clkf control
armpll_bwadj	pre_shift_test_reg[30:19]	12'd49	armpll_bwadj control
armpll_clkod	pre_shift_test_reg[35:31]	5'd1	armpll_clkod control
ddrpll_clkr	pre_shift_test_reg[41:36]	6'd1	ddrpll_clkr control
ddrpll_clkf	pre_shift_test_reg[54:42]	13'd99	ddrpll_clkf control
ddrpll_bwadj	pre_shift_test_reg[66:55]	12'd49	ddrpll_bwadj control
ddrpll_clkod	pre_shift_test_reg[71:67]	5'd5	ddrpll_clkod control
codecpll_clkr	pre_shift_test_reg[77:72]	6'd1	codecpll_clkr control
codecpll_clkf	pre_shift_test_reg[90:78]	13'd99	codecpll_clkf control
codecpll_bwadj	pre_shift_test_reg[102:91]	12'd49	codecpll_bwadj control
codecpll_clkod	pre_shift_test_reg[107:103]	5'd5	codecpll_clkod control
generalpll_clkr	pre_shift_test_reg[113:108]	6'd1	generalpll_clkr control
generalpll_clkf	pre_shift_test_reg[126:114]	13'd99	generalpll_clkf control
generalpll_bwadj	pre_shift_test_reg[138:127]	12'd49	generalpll_bwadj control
generalpll_clkod	pre_shift_test_reg[143:139]	5'd5	generalpll_clkod control
newpll_clkr	pre_shift_test_reg[149:144]	6'd1	newpll_clkr control
newpll_clkf	pre_shift_test_reg[162:150]	13'd99	newpll_clkf control
newpll_bwadj	pre_shift_test_reg[174:163]	12'd49	newpll_bwadj control
newpll_clkod	pre_shift_test_reg[179:175]	5'd5	newpll_clkod control
testclk_sel	pre_shift_test_reg[182:180]	3'd0	testclk_out select in testmode
aclk_core_m_div_con	pre_shift_test_reg[185:183]	3'd1	Aclk_m divider

Name	Bit number	Default value	Description
			configuration in testmode
Io_sr	pre_shift_test_reg[186]	1’d1	IO slew rate
io_drive	pre_shift_test_reg[188:187]	2’b10	IO drive configuration
Io_vsel	pre_shift_test_reg[189]	1’d0	IO voltage select
Io_smt	pre_shift_test_reg[190]	1’d0	IO smt control

Pre-shift relative controls IO are as follow.

Name	IO	Description
Pre_shift_datain	IO_UART3GPSsout_GPSsig_HSADCT1data1_GPIO30gpio7b0	Pre-shift data in
Pre_shift_en	IO_UART3GPSctsn_GPSrfclk_GPST1clk_GPIO30gpio7b1	Pre-shift enable
Pre_shift_clk	IO_UART3GPSrtsn_USBdrvvbus0_GPIO30gpio7b2	Pre-shift clock
Pre-shift_default_select	IO_USBdrvvbus1_EDPhotplug_GPIO30gpio7b3	1’b0: pre_shift use default value; 1’b1: pre_shift use shift in value;
Pre-shift_select	IO_ISPshutteren_SPI1clk_GPIO30gpio7b4	1’b0: disable, testmode do not use pre-shift config value; use internal 6 configs. 1’b1: enable, testmode use pre-shift config value;

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 two sections, one is GRF for non-secure system, the other is SGRF for secure system.

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

4.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO1D_IOMUX	0x000c	W	0x00000000	GPIO1D iomux control
GRF_GPIO2A_IOMUX	0x0010	W	0x00000000	GPIO2A iomux control
GRF_GPIO2B_IOMUX	0x0014	W	0x00000000	GPIO2B iomux control
GRF_GPIO2C_IOMUX	0x0018	W	0x00000000	GPIO2C iomux control
GRF_GPIO3A_IOMUX	0x0020	W	0x00000000	GPIO3A iomux control
GRF_GPIO3B_IOMUX	0x0024	W	0x00000000	GPIO3B iomux control
GRF_GPIO3C_IOMUX	0x0028	W	0x00000000	GPIO3C iomux control
GRF_GPIO3DL_IOMUX	0x002c	W	0x00000000	GPIO3D iomux control
GRF_GPIO3DH_IOMUX	0x0030	W	0x00000000	GPIO3D iomux control
GRF_GPIO4AL_IOMUX	0x0034	W	0x00000000	GPIO4A iomux control
GRF_GPIO4AH_IOMUX	0x0038	W	0x00000000	GPIO4A iomux control
GRF_GPIO4BL_IOMUX	0x003c	W	0x00000000	GPIO4B iomux control
GRF_GPIO4C_IOMUX	0x0044	W	0x00000000	GPIO4C iomux control
GRF_GPIO4D_IOMUX	0x0048	W	0x00000000	GPIO4D iomux control
GRF_GPIO5B_IOMUX	0x0050	W	0x00000000	GPIO5B iomux control
GRF_GPIO5C_IOMUX	0x0054	W	0x00000000	GPIO5C iomux control
GRF_GPIO6A_IOMUX	0x005c	W	0x00000000	GPIO6A iomux control
GRF_GPIO6B_IOMUX	0x0060	W	0x00000000	GPIO6B iomux control
GRF_GPIO6C_IOMUX	0x0064	W	0x00001555	GPIO6C iomux control
GRF_GPIO7A_IOMUX	0x006c	W	0x00000000	GPIO7A iomux control
GRF_GPIO7B_IOMUX	0x0070	W	0x00000000	GPIO7B iomux control
GRF_GPIO7CL_IOMUX	0x0074	W	0x00000000	GPIO7CL iomux control
GRF_GPIO7CH_IOMUX	0x0078	W	0x00000000	GPIO7CH iomux control
GRF_GPIO8A_IOMUX	0x0080	W	0x00000000	GPIO8A iomux control

Name	Offset	Size	Reset Value	Description
GRF_GPIO8B_IOMUX	0x0084	W	0x00000000	GPIO8B iomux control
GRF_GPIO1H_SR	0x0104	W	0x00000f00	GPIO1C/D SR control
GRF_GPIO2L_SR	0x0108	W	0x00000000	GPIO2A/B SR control
GRF_GPIO2H_SR	0x010c	W	0x00000000	GPIO2C/D SR control
GRF_GPIO3L_SR	0x0110	W	0x000020ff	GPIO3A/B SR control
GRF_GPIO3H_SR	0x0114	W	0x0000ff04	GPIO3C/D SR control
GRF_GPIO4L_SR	0x0118	W	0x00000120	GPIO4A/B SR control
GRF_GPIO4H_SR	0x011c	W	0x00000000	GPIO4C/D SR control
GRF_GPIO5L_SR	0x0120	W	0x00000000	GPIO5A/B SR control
GRF_GPIO5H_SR	0x0124	W	0x00000000	GPIO5C/D SR control
GRF_GPIO6L_SR	0x0128	W	0x00000100	GPIO6A/B SR control
GRF_GPIO6H_SR	0x012c	W	0x00000010	GPIO6C/D SR control
GRF_GPIO7L_SR	0x0130	W	0x00000000	GPIO7A/B SR control
GRF_GPIO7H_SR	0x0134	W	0x00000000	GPIO7C/D SR control
GRF_GPIO8L_SR	0x0138	W	0x00000000	GPIO8A/B SR control
GRF_GPIO1D_P	0x014c	W	0x0000aaaa	GPIO1D PU/PD control
GRF_GPIO2A_P	0x0150	W	0x0000aaaa	GPIO2A PU/PD control
GRF_GPIO2B_P	0x0154	W	0x0000aaaa	GPIO2B PU/PD control
GRF_GPIO2C_P	0x0158	W	0x0000aaa5	GPIO2C PU/PD control
GRF_GPIO3A_P	0x0160	W	0x00005555	GPIO3A PU/PD control
GRF_GPIO3B_P	0x0164	W	0x00005699	GPIO3B PU/PD control
GRF_GPIO3C_P	0x0168	W	0x0000aaa5	GPIO3C PU/PD control
GRF_GPIO3D_P	0x016c	W	0x00005555	GPIO3D PU/PD control
GRF_GPIO4A_P	0x0170	W	0x00005555	GPIO4A PU/PD control
GRF_GPIO4B_P	0x0174	W	0x0000aaa5	GPIO4B PU/PD control
GRF_GPIO4C_P	0x0178	W	0x00005559	GPIO4C PU/PD control
GRF_GPIO4D_P	0x017c	W	0x00005a99	GPIO4D PU/PD control
GRF_GPIO5B_P	0x0184	W	0x00006559	GPIO5B PU/PD control
GRF_GPIO5C_P	0x0188	W	0x0000aaa9	GPIO5C PU/PD control
GRF_GPIO6A_P	0x0190	W	0x0000aaaa	GPIO6A PU/PD control
GRF_GPIO6B_P	0x0194	W	0x0000aa96	GPIO6B PU/PD control
GRF_GPIO6C_P	0x0198	W	0x00005655	GPIO6C PU/PD control
GRF_GPIO7A_P	0x01a0	W	0x000059aa	GPIO7A PU/PD control
GRF_GPIO7B_P	0x01a4	W	0x0000a696	GPIO7B PU/PD control
GRF_GPIO7C_P	0x01a8	W	0x00005955	GPIO7C PU/PD control
GRF_GPIO8A_P	0x01b0	W	0x00006555	GPIO8A PU/PD control
GRF_GPIO8B_P	0x01b4	W	0x0000aaaa	GPIO8B PU/PD control
GRF_GPIO1D_E	0x01cc	W	0x000055aa	GPIO1D drive strength control
GRF_GPIO2A_E	0x01d0	W	0x0000aaaa	GPIO2A drive strength control
GRF_GPIO2B_E	0x01d4	W	0x0000aaaa	GPIO2B drive strength control
GRF_GPIO2C_E	0x01d8	W	0x00005555	GPIO2C drive strength control
GRF_GPIO3A_E	0x01e0	W	0x0000aaaa	GPIO3A drive strength control
GRF_GPIO3B_E	0x01e4	W	0x00005955	GPIO3B drive strength control

Name	Offset	Size	Reset Value	Description
GRF_GPIO3C_E	0x01e8	W	0x00005565	GPIO3C drive strength control
GRF_GPIO3D_E	0x01ec	W	0x0000aaaa	GPIO3D drive strength control
GRF_GPIO4A_E	0x01f0	W	0x00005955	GPIO4A drive strength control
GRF_GPIO4B_E	0x01f4	W	0x00005556	GPIO4B drive strength control
GRF_GPIO4C_E	0x01f8	W	0x00005555	GPIO4C drive strength control
GRF_GPIO4D_E	0x01fc	W	0x00005555	GPIO4D drive strength control
GRF_GPIO5B_E	0x0204	W	0x00005555	GPIO5B drive strength control
GRF_GPIO5C_E	0x0208	W	0x00005555	GPIO5C drive strength control
GRF_GPIO6A_E	0x0210	W	0x00005555	GPIO6A drive strength control
GRF_GPIO6B_E	0x0214	W	0x00005555	GPIO6B drive strength control
GRF_GPIO6C_E	0x0218	W	0x00005555	GPIO6C drive strength control
GRF_GPIO7A_E	0x0220	W	0x00005555	GPIO7A drive strength control
GRF_GPIO7B_E	0x0224	W	0x00005555	GPIO7B drive strength control
GRF_GPIO7C_E	0x0228	W	0x00005555	GPIO7C drive strength control
GRF_GPIO8A_E	0x0230	W	0x00005555	GPIO8A drive strength control
GRF_GPIO8B_E	0x0234	W	0x00005555	GPIO8B drive strength control
GRF_GPIO_SMT	0x0240	W	0x00000fff	GPIO smitter control register
GRF_SOC_CON0	0x0244	W	0x00001c18	SoC control register 0
GRF_SOC_CON1	0x0248	W	0x00004040	SoC control register 1
GRF_SOC_CON2	0x024c	W	0x00000002	SoC control register 2
GRF_SOC_CON3	0x0250	W	0x00000810	SoC control register 3
GRF_SOC_CON4	0x0254	W	0x00000607	SoC control register 4
GRF_SOC_CON5	0x0258	W	0x00008c87	SoC control register 5
GRF_SOC_CON6	0x025c	W	0x00008000	SoC control register 6
GRF_SOC_CON7	0x0260	W	0x00000000	SoC control register 7
GRF_SOC_CON8	0x0264	W	0x0000000e	SoC control register 8
GRF_SOC_CON9	0x0268	W	0x0000000e	SoC control register 9
GRF_SOC_CON10	0x026c	W	0x0000000f	SoC control register 10
GRF_SOC_CON11	0x0270	W	0x00000000	SoC control register 11
GRF_SOC_CON12	0x0274	W	0x00000013	SoC control register 12
GRF_SOC_CON13	0x0278	W	0x00000000	SoC control register 13
GRF_SOC_CON14	0x027c	W	0x00000000	SoC control register 14
GRF_SOC_STATUS0	0x0280	W	0x00000000	SoC status register 0
GRF_SOC_STATUS1	0x0284	W	0x00000000	SoC status register 1
GRF_SOC_STATUS2	0x0288	W	0x00000000	SoC status register 2
GRF_SOC_STATUS3	0x028c	W	0x00000000	SoC status register 3
GRF_SOC_STATUS4	0x0290	W	0x00000000	SoC status register 4
GRF_SOC_STATUS5	0x0294	W	0x00000000	SoC status register 5
GRF_SOC_STATUS6	0x0298	W	0x00000000	SoC status register 6
GRF_SOC_STATUS7	0x029c	W	0x00000000	SoC status register 7
GRF_SOC_STATUS8	0x02a0	W	0x00000000	SoC status register 8
GRF_SOC_STATUS9	0x02a4	W	0x00000000	SoC status register 9
GRF_SOC_STATUS10	0x02a8	W	0x00000000	SoC status register 10

Name	Offset	Size	Reset Value	Description
GRF_SOC_STATUS11	0x02ac	W	0x00000000	SoC status register 11
GRF_SOC_STATUS12	0x02b0	W	0x00000000	SoC status register 12
GRF_SOC_STATUS13	0x02b4	W	0x00000000	SoC status register 13
GRF_SOC_STATUS14	0x02b8	W	0x00000000	SoC status register 14
GRF_SOC_STATUS15	0x02bc	W	0x00000000	SoC status register 15
GRF_SOC_STATUS16	0x02c0	W	0x00000000	SoC status register 16
GRF_SOC_STATUS17	0x02c4	W	0x00000000	SoC status register 17
GRF_SOC_STATUS18	0x02c8	W	0x00000000	SoC status register 18
GRF_SOC_STATUS19	0x02cc	W	0x00000000	SoC status register 19
GRF_SOC_STATUS20	0x02d0	W	0x00000000	SoC status register 20
GRF_SOC_STATUS21	0x02d4	W	0x00000000	SoC status register 21
GRF_PERIDMAC_CON0	0x02e0	W	0x000000fa	PERI DMAC control register 0
GRF_PERIDMAC_CON1	0x02e4	W	0x00000000	PERI DMAC control register 1
GRF_PERIDMAC_CON2	0x02e8	W	0x0000ffff	PERI DMAC control register 2
GRF_PERIDMAC_CON3	0x02ec	W	0x0000ffff	PERI DMAC control register 3
GRF_DDRC0_CON0	0x02f0	W	0x00000000	DDRC0 control register 0
GRF_DDRC1_CON0	0x02f4	W	0x00000000	DDRC1 control register 0
GRF_CPU_CON0	0x02f8	W	0x00008220	CPU control register 0
GRF_CPU_CON1	0x02fc	W	0x00000ff0	CPU control register 1
GRF_CPU_CON2	0x0300	W	0x00000fff	CPU control register 2
GRF_CPU_CON3	0x0304	W	0x00000000	CPU control register 3
GRF_CPU_CON4	0x0308	W	0x00002400	CPU control register 4
GRF_CPU_STATUS0	0x0318	W	0x00000000	CPU status register 0
GRF_UOC0_CON0	0x0320	W	0x00000089	UOC0 control register 0
GRF_UOC0_CON1	0x0324	W	0x00007333	UOC0 control register 1
GRF_UOC0_CON2	0x0328	W	0x00000d08	UOC0 control register 2
GRF_UOC0_CON3	0x032c	W	0x00000001	UOC0 control register 3
GRF_UOC0_CON4	0x0330	W	0x00000003	UOC0 control register 4
GRF_UOC1_CON0	0x0334	W	0x00000b89	UOC1 control register 0
GRF_UOC1_CON1	0x0338	W	0x00007333	UOC1 control register 1
GRF_UOC1_CON2	0x033c	W	0x00000d08	UOC1 control register 2
GRF_UOC1_CON3	0x0340	W	0x00001c41	UOC1 control register 3
GRF_UOC1_CON4	0x0344	W	0x0000c820	UOC1 control register 4
GRF_UOC2_CON0	0x0348	W	0x00000089	UOC2 control register 0
GRF_UOC2_CON1	0x034c	W	0x00007333	UOC2 control register 1
GRF_UOC2_CON2	0x0350	W	0x00000d08	UOC2 control register 2
GRF_UOC2_CON3	0x0354	W	0x00001c01	UOC2 control register 3
GRF_UOC3_CON0	0x0358	W	0x000030eb	UOC3 control register 0
GRF_UOC3_CON1	0x035c	W	0x00000003	UOC3 control register 1
GRF_UOC4_CON0	0x0360	W	0x00001080	UOC4 control register 0
GRF_UOC4_CON1	0x0364	W	0x00000820	UOC4 control register 1
GRF_PVTM_CON0	0x0368	W	0x00000000	PVT monitor control register 0
GRF_PVTM_CON1	0x036c	W	0x016e3600	PVT monitor control register 1

Name	Offset	Size	Reset Value	Description
GRF_PVTM_CON2	0x0370	W	0x016e3600	PVT monitor control register 2
GRF_PVTM_STATUS0	0x0374	W	0x00000000	PVT monitor status register 0
GRF_PVTM_STATUS1	0x0378	W	0x00000000	PVT monitor status register 1
GRF_PVTM_STATUS2	0x037c	W	0x00000000	PVT monitor status register 2
GRF_IO_VSEL	0x0380	W	0x00000004	IO voltage select
GRF_SARADC_TESTBIT	0x0384	W	0x00000000	SARADC Test bit register
GRF_TSADC_TESTBIT_L	0x0388	W	0x00000000	TSADC Test bit low register
GRF_TSADC_TESTBIT_H	0x038c	W	0x00000000	TSADC Test bit high register
GRF_OS_REG0	0x0390	W	0x00000000	OS register 0
GRF_OS_REG1	0x0394	W	0x00000000	OS register 1
GRF_OS_REG2	0x0398	W	0x00000000	OS register 2
GRF_OS_REG3	0x039c	W	0x00000000	OS register 3
GRF_SOC_CON15	0x03a4	W	0x00000000	SoC control register 15
GRF_SOC_CON16	0x03a8	W	0x00000000	SoC control register 16

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

4.3.2 Detail Register Description

GRF_GPIO1D_IOMUX
Address: Operational Base + offset (0x000c)
GPIO1D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	gpio1d3_sel GPIO1D[3] iomux select 1'b0: gpio 1'b1: lcdc0_dclk
5	RO	0x0	reserved
4	RW	0x0	gpio1d2_sel GPIO1D[2] iomux select 1'b0: gpio 1'b1: lcdc0_den
3	RO	0x0	reserved
2	RW	0x0	gpio1d1_sel GPIO1D[1] iomux select 1'b0: gpio 1'b1: lcdc0_vsync
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio1d0_sel GPIO1D[0] iomux select 1'b0: gpio 1'b1: lcdc0_hsync

GRF_GPIO2A_IOMUX

Address: Operational Base + offset (0x0010)

GPIO2A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	gpio2a7_sel GPIO2A[7] iomux select 2'b00: gpio 2'b01: cif_data9 2'b10: host_din5 2'b11: hsadc_data7
13:12	RW	0x0	gpio2a6_sel GPIO2A[6] iomux select 2'b00: gpio 2'b01: cif_data8 2'b10: host_din4 2'b11: hsadc_data6
11:10	RW	0x0	gpio2a5_sel GPIO2A[5] iomux select 2'b00: gpio 2'b01: cif_data7 2'b10: host_ckinn 2'b11: hsadc_data5
9:8	RW	0x0	gpio2a4_sel GPIO2A[4] iomux select 2'b00: gpio 2'b01: cif_data6 2'b10: host_ckinp 2'b11: hsadc_data4
7:6	RW	0x0	gpio2a3_sel GPIO2A[3] iomux select 2'b00: gpio 2'b01: cif_data5 2'b10: host_din3 2'b11: hsadc_data3

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio2a2_sel GPIO2A[2] iomux select 2'b00: gpio 2'b01: cif_data4 2'b10: host_din2 2'b11: hsadc_data2
3:2	RW	0x0	gpio2a1_sel GPIO2A[1] iomux select 2'b00: gpio 2'b01: cif_data3 2'b10: host_din1 2'b11: hsadc_data1
1:0	RW	0x0	gpio2a0_sel GPIO2A[0] iomux select 2'b00: gpio 2'b01: cif_data2 2'b10: host_din0 2'b11: hsadc_data0

GRF_GPIO2B_IOMUX
Address: Operational Base + offset (0x0014)
GPIO2B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio2b7_sel GPIO2B[7] iomux select 1'b0: gpio 1'b1: cif_data11
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio2b6_sel GPIO2B[6] iomux select 1'b0: gpio 1'b1: cif_data10
11	RO	0x0	reserved
10	RW	0x0	gpio2b5_sel GPIO2B[5] iomux select 1'b0: gpio 1'b1: cif_data1
9	RO	0x0	reserved
8	RW	0x0	gpio2b4_sel GPIO2B[4] iomux select 1'b0: gpio 1'b1: cif_data0
7:6	RW	0x0	gpio2b3_sel GPIO2B[3] iomux select 2'b00: gpio 2'b01: cif_clkout 2'b10: host_wkreq 2'b11: hsadcts_fail
5:4	RW	0x0	gpio2b2_sel GPIO2B[2] iomux select 2'b00: gpio 2'b01: cif_clkin 2'b10: host_wkack 2'b11: gps_clk (when hsadc_clkout_en==0) hsadc_clkout (when hsadc_clkout_en==1)
3:2	RW	0x0	gpio2b1_sel GPIO2B[1] iomux select 2'b00: gpio 2'b01: cif_href 2'b10: host_din7 2'b11: hsadcts_valid
1:0	RW	0x0	gpio2b0_sel GPIO2B[0] iomux select 2'b00: gpio 2'b01: cif_vsync 2'b10: host_din6 2'b11: hsadcts_sync

GRF_GPIO2C_IOMUX
Address: Operational Base + offset (0x0018)
GPIO2C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	gpio2c1_sel GPIO2C[1] iomux select 1'b0: gpio 1'b1: i2c3cam_sda
1	RO	0x0	reserved
0	RW	0x0	gpio2c0_sel GPIO2C[0] iomux select 1'b0: gpio 1'b1: i2c3cam_scl

GRF_GPIO3A_IOMUX
Address: Operational Base + offset (0x0020)
GPIO3A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	gpio3a7_sel GPIO3A[7] iomux select 2'b00: gpio 2'b01: flash0_data7 2'b10: emmc_data7 2'b11: reserved
13:12	RW	0x0	gpio3a6_sel GPIO3A[6] iomux select 2'b00: gpio 2'b01: flash0_data6 2'b10: emmc_data6 2'b11: reserved
11:10	RW	0x0	gpio3a5_sel GPIO3A[5] iomux select 2'b00: gpio 2'b01: flash0_data5 2'b10: emmc_data5 2'b11: reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio3a4_sel GPIO3A[4] iomux select 2'b00: gpio 2'b01: flash0_data4 2'b10: emmc_data4 2'b11: reserved
7:6	RW	0x0	gpio3a3_sel GPIO3A[3] iomux select 2'b00: gpio 2'b01: flash0_data3 2'b10: emmc_data3 2'b11: reserved
5:4	RW	0x0	gpio3a2_sel GPIO3A[2] iomux select 2'b00: gpio 2'b01: flash0_data2 2'b10: emmc_data2 2'b11: reserved
3:2	RW	0x0	gpio3a1_sel GPIO3A[1] iomux select 2'b00: gpio 2'b01: flash0_data1 2'b10: emmc_data1 2'b11: reserved
1:0	RW	0x0	gpio3a0_sel GPIO3A[0] iomux select 2'b00: gpio 2'b01: flash0_data0 2'b10: emmc_data0 2'b11: reserved

GRF_GPIO3B_IOMUX
Address: Operational Base + offset (0x0024)
GPIO3B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	gpio3b7_sel GPIO3B[7] iomux select 1'b0: gpio 1'b1: flash0_csn1
13	RO	0x0	reserved
12	RW	0x0	gpio3b6_sel GPIO3B[6] iomux select 1'b0: gpio 1'b1: flash0_csn0
11	RO	0x0	reserved
10	RW	0x0	gpio3b5_sel GPIO3B[5] iomux select 1'b0: gpio 1'b1: flash0_wrn
9	RO	0x0	reserved
8	RW	0x0	gpio3b4_sel GPIO3B[4] iomux select 1'b0: gpio 1'b1: flash0_cle
7	RO	0x0	reserved
6	RW	0x0	gpio3b3_sel GPIO3B[3] iomux select 1'b0: gpio 1'b1: flash0_ale
5	RO	0x0	reserved
4	RW	0x0	gpio3b2_sel GPIO3B[2] iomux select 1'b0: gpio 1'b1: flash0_rdn
3:2	RW	0x0	gpio3b1_sel GPIO3B[1] iomux select 2'b00: gpio 2'b01: flash0_wp 2'b10: emmc_pwren 2'b11: reserved
1	RO	0x0	reserved
0	RW	0x0	gpio3b0_sel GPIO3B[0] iomux select 1'b0: gpio 1'b1: flash0_rdy

GRF_GPIO3C_IOMUX
Address: Operational Base + offset (0x0028)
GPIO3C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:6	RO	0x0	reserved
5:4	RW	0x0	gpio3c2_sel GPIO3C[2] iomux select 2'b00: gpio 2'b01: flash0_dqs 2'b10: emmc_clkout 2'b11: reserved
3:2	RW	0x0	gpio3c1_sel GPIO3C[1] iomux select 2'b00: gpio 2'b01: flash0_csn3 2'b10: emmc_rstnout 2'b11: reserved
1:0	RW	0x0	gpio3c0_sel GPIO3C[0] iomux select 2'b00: gpio 2'b01: flash0_csn2 2'b10: emmc_cmd 2'b11: reserved

GRF_GPIO3DL_IOMUX
Address: Operational Base + offset (0x002c)
GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable 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	0x0	gpio3d3_sel GPIO3D[3] iomux select 3'b000: gpio 3'b001: flash1_data3 3'b010: host_dout3 3'b011: mac_rxd3 3'b100: sdio1_data3 other: reserved

Bit	Attr	Reset Value	Description
11	RO	0x0	reserved
10:8	RW	0x0	gpio3d2_sel GPIO3D[2] iomux select 3'b000: gpio 3'b001: flash1_data2 3'b010: host_dout2 3'b011: mac_rxd2 3'b100: sdio1_data2 other: reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3d1_sel GPIO3D[1] iomux select 3'b000: gpio 3'b001: flash1_data1 3'b010: host_dout1 3'b011: mac_txd3 3'b100: sdio1_data1 other: reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3d0_sel GPIO3D[0] iomux select 3'b000: gpio 3'b001: flash1_data0 3'b010: host_dout0 3'b011: mac_txd2 3'b100: sdio1_data0 other: reserved

GRF_GPIO3DH_IOMUX
Address: Operational Base + offset (0x0030)
GPIO3D iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable 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
14:12	RW	0x0	gpio3d7_sel GPIO3D[7] iomux select 3'b000: gpio 3'b001: flash1_data7 3'b010: host_dout7 3'b011: mac_rxd1 3'b100: sdio1_intn other: reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio3d6_sel GPIO3D[6] iomux select 3'b000: gpio 3'b001: flash1_data6 3'b010: host_dout6 3'b011: mac_rxd0 3'b100: sdio1_bkpwr other: reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio3d5_sel GPIO3D[5] iomux select 3'b000: gpio 3'b001: flash1_data5 3'b010: host_dout5 3'b011: mac_txd1 3'b100: sdio1_wrprt other: reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio3d4_sel GPIO3D[4] iomux select 3'b000: gpio 3'b001: flash1_data4 3'b010: host_dout4 3'b011: mac_txd0 3'b100: sdio1_detectn other: reserved

GRF_GPIO4AL_IOMUX
Address: Operational Base + offset (0x0034)
GPIO4A iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	0x0	gpio4a3_sel GPIO4A[3] iomux select 3'b001: flash1_ale 3'b010: host_dout9 3'b011: mac_clk 3'b100: flash0_csn6 other: reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio4a2_sel GPIO4A[2] iomux select 3'b000: gpio 3'b001: flash1_rdn 3'b010: host_dout8 3'b011: mac_rxer 3'b100: flash0_csn5 other: reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio4a1_sel GPIO4A[1] iomux select 3'b000: gpio 3'b001: flash1_wp 3'b010: host_ckoutn 3'b011: mac_rxdv 3'b100: flash0_csn4 other: reserved
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio4a0_sel GPIO4A[0] iomux select 2'b00: gpio 2'b01: flash1_rdy 2'b10: host_ckoutp 2'b11: mac_mdc

GRF_GPIO4AH_IOMUX
Address: Operational Base + offset (0x0038)
GPIO4A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	0x0	gpio4a7_sel GPIO4A[7] iomux select 3'b000: gpio 3'b001: flash1_csn1 3'b010: host_dout13 3'b011: mac_crs 3'b100: sdio1_clkout other: reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio4a6_sel GPIO4A[6] iomux select 3'b000: gpio 3'b001: flash1_csn0 3'b010: host_dout12 3'b011: mac_rxclk 3'b100: sdio1_cmd other: reserved
7:6	RO	0x0	reserved
5:4	RW	0x0	gpio4a5_sel GPIO4A[5] iomux select 2'b00: gpio 2'b01: flash1_wrn 2'b10: host_dout11 2'b11: mac_mdio
3	RO	0x0	reserved
2:0	RW	0x0	gpio4a4_sel GPIO4A[4] iomux select 3'b000: gpio 3'b001: flash1_cle 3'b010: host_dout10 3'b011: mac_txen 3'b100: flash0_csn7 other: reserved

GRF_GPIO4BL_IOMUX
Address: Operational Base + offset (0x003c)
GPIO4B iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	0x0	gpio4b1_sel GPIO4B[1] iomux select 3'b000: gpio 3'b001: flash1_csn2 3'b010: host_dout15 3'b011: mac_txclk 3'b100: sdio1_pwren other: reserved
3	RO	0x0	reserved
2:0	RW	0x0	gpio4b0_sel GPIO4B[0] iomux select 3'b000: gpio 3'b001: flash1_dqs 3'b010: host_dout14 3'b011: mac_col 3'b100: flash1_csn3 other: reserved

GRF_GPIO4C_IOMUX
Address: Operational Base + offset (0x0044)
GPIO4C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	gpio4c7_sel GPIO4C[7] iomux select 1'b0: gpio 1'b1: sdio0_data3
13	RO	0x0	reserved
12	RW	0x0	gpio4c6_sel GPIO4C[6] iomux select 1'b0: gpio 1'b1: sdio0_data2

Bit	Attr	Reset Value	Description
11	RO	0x0	reserved
10	RW	0x0	gpio4c5_sel GPIO4C[5] iomux select 1'b0: gpio 1'b1: sdio0_data1
9	RO	0x0	reserved
8	RW	0x0	gpio4c4_sel GPIO4C[4] iomux select 1'b0: gpio 1'b1: sdio0_data0
7	RO	0x0	reserved
6	RW	0x0	gpio4c3_sel GPIO4C[3] iomux select 1'b0: gpio 1'b1: uart0bt_rtsn
5	RO	0x0	reserved
4	RW	0x0	gpio4c2_sel GPIO4C[2] iomux select 1'b0: gpio 1'b1: uart0bt_ctsn
3	RO	0x0	reserved
2	RW	0x0	gpio4c1_sel GPIO4C[1] iomux select 1'b0: gpio 1'b1: uart0bt_sout
1	RO	0x0	reserved
0	RW	0x0	gpio4c0_sel GPIO4C[0] iomux select 1'b0: gpio 1'b1: uart0bt_sin

GRF_GPIO4D_IOMUX
Address: Operational Base + offset (0x0048)
GPIO4D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	RW	0x0	gpio4d6_sel GPIO4D[6] iomux select 1'b0: gpio 1'b1: sdio0_intn
11	RO	0x0	reserved
10	RW	0x0	gpio4d5_sel GPIO4D[5] iomux select 1'b0: gpio 1'b1: sdio0_bkpwr
9	RO	0x0	reserved
8	RW	0x0	gpio4d4_sel GPIO4D[4] iomux select 1'b0: gpio 1'b1: sdio0_pwren
7	RO	0x0	reserved
6	RW	0x0	gpio4d3_sel GPIO4D[3] iomux select 1'b0: gpio 1'b1: sdio0_wrprt
5	RO	0x0	reserved
4	RW	0x0	gpio4d2_sel GPIO4D[2] iomux select 1'b0: gpio 1'b1: sdio0_detectn
3	RO	0x0	reserved
2	RW	0x0	gpio4d1_sel GPIO4D[1] iomux select 1'b0: gpio 1'b1: sdio0_clkout
1	RO	0x0	reserved
0	RW	0x0	gpio4d0_sel GPIO4D[0] iomux select 1'b0: gpio 1'b1: sdio0_cmd

GRF_GPIO5B_IOMUX
Address: Operational Base + offset (0x0050)
GPIO5B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio5b7_sel GPIO5B[7] iomux select 2'b00: gpio 2'b01: spi0_rxd 2'b10: ts0_data7 2'b11: uart4exp_sin
13:12	RW	0x0	gpio5b6_sel GPIO5B[6] iomux select 2'b00: gpio 2'b01: spi0_txd 2'b10: ts0_data6 2'b11: uart4exp_sout
11:10	RW	0x0	gpio5b5_sel GPIO5B[5] iomux select 2'b00: gpio 2'b01: spi0_csn0 2'b10: ts0_data5 2'b11: uart4exp_rtsn
9:8	RW	0x0	gpio5b4_sel GPIO5B[4] iomux select 2'b00: gpio 2'b01: spi0_clk 2'b10: ts0_data4 2'b11: uart4exp_ctsn
7:6	RW	0x0	gpio5b3_sel GPIO5B[3] iomux select 2'b00: gpio 2'b01: uart1bb_rtsn 2'b10: ts0_data3 2'b11: reserved
5:4	RW	0x0	gpio5b2_sel GPIO5B[2] iomux select 2'b00: gpio 2'b01: uart1bb_ctsn 2'b10: ts0_data2 2'b11: reserved
3:2	RW	0x0	gpio5b1_sel GPIO5B[1] iomux select 2'b00: gpio 2'b01: uart1bb_sout 2'b10: ts0_data1 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio5b0_sel GPIO5B[0] iomux select 2'b00: gpio 2'b01: uart1bb_sin 2'b10: ts0_data0 2'b11: reserved

GRF_GPIO5C_IOMUX

Address: Operational Base + offset (0x0054)
GPIO5C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	gpio5c3_sel GPIO5C[3] iomux select 1'b0: gpio 1'b1: ts0_err
5	RO	0x0	reserved
4	RW	0x0	gpio5c2_sel GPIO5C[2] iomux select 1'b0: gpio 1'b1: ts0_clk
3	RO	0x0	reserved
2	RW	0x0	gpio5c1_sel GPIO5C[1] iomux select 1'b0: gpio 1'b1: ts0_valid
1:0	RW	0x0	gpio5c0_sel GPIO5C[0] iomux select 2'b00: gpio 2'b01: spi0_csn1 2'b10: ts0_sync 2'b11: reserved

GRF_GPIO6A_IOMUX

Address: Operational Base + offset (0x005c)
GPIO6A iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	gpio6a7_sel GPIO6A[7] iomux select 1'b0: gpio 1'b1: i2s_sdo3
13	RO	0x0	reserved
12	RW	0x0	gpio6a6_sel GPIO6A[6] iomux select 1'b0: gpio 1'b1: i2s_sdo2
11	RO	0x0	reserved
10	RW	0x0	gpio6a5_sel GPIO6A[5] iomux select 1'b0: gpio 1'b1: i2s_sdo1
9	RO	0x0	reserved
8	RW	0x0	gpio6a4_sel GPIO6A[4] iomux select 1'b0: gpio 1'b1: i2s_sdo0
7	RO	0x0	reserved
6	RW	0x0	gpio6a3_sel GPIO6A[3] iomux select 1'b0: gpio 1'b1: i2s_sdi
5	RO	0x0	reserved
4	RW	0x0	gpio6a2_sel GPIO6A[2] iomux select 1'b0: gpio 1'b1: i2s_lrcktx
3	RO	0x0	reserved
2	RW	0x0	gpio6a1_sel GPIO6A[1] iomux select 1'b0: gpio 1'b1: i2s_lrckrx
1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio6a0_sel GPIO6A[0] iomux select 1'b0: gpio 1'b1: i2s_sclk

GRF_GPIO6B_IOMUX

Address: Operational Base + offset (0x0060)

GPIO6B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	gpio6b3_sel GPIO6B[3] iomux select 1'b0: gpio 1'b1: spdif_tx
5	RO	0x0	reserved
4	RW	0x0	gpio6b2_sel GPIO6B[2] iomux select 1'b0: gpio 1'b1: i2c1audio_scl
3	RO	0x0	reserved
2	RW	0x0	gpio6b1_sel GPIO6B[1] iomux select 1'b0: gpio 1'b1: i2c1audio_sda
1	RO	0x0	reserved
0	RW	0x0	gpio6b0_sel GPIO6B[0] iomux select 1'b0: gpio 1'b1: i2s_clk

GRF_GPIO6C_IOMUX

Address: Operational Base + offset (0x0064)

GPIO6C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable 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	0x1	gpio6c6_sel GPIO6C[6] iomux select 1'b0: gpio 1'b1: sdmmc0_dectn
11	RO	0x0	reserved
10	RW	0x1	gpio6c5_sel GPIO6C[5] iomux select 1'b0: gpio 1'b1: sdmmc0_cmd
9:8	RW	0x1	gpio6c4_sel GPIO6C[4] iomux select 2'b00: gpio 2'b01: sdmmc0_clkout 2'b10: jtag_tdo 2'b11: reserved
7:6	RW	0x1	gpio6c3_sel GPIO6C[3] iomux select 2'b00: gpio 2'b01: sdmmc0_data3 2'b10: jtag_tck 2'b11: reserved
5:4	RW	0x1	gpio6c2_sel GPIO6C[2] iomux select 2'b00: gpio 2'b01: sdmmc0_data2 2'b10: jtag_tdi 2'b11: reserved
3:2	RW	0x1	gpio6c1_sel GPIO6C[1] iomux select 2'b00: gpio 2'b01: sdmmc0_data1 2'b10: jtag_trstn 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio6c0_sel GPIO6C[0] iomux select 2'b00: gpio 2'b01: sdmmc0_data0 2'b10: jtag_tms 2'b11: reserved

GRF_GPIO7A_IOMUX
Address: Operational Base + offset (0x006c)
GPIO7A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	gpio7a7_sel GPIO7A[7] iomux select 2'b00: gpio 2'b01: uart3gps_sin 2'b10: gps_mag 2'b11: hsadct1_data0
13:3	RO	0x0	reserved
2	RW	0x0	gpio7a1_sel GPIO7A[1] iomux select 1'b0: gpio 1'b1: pwm_1
1:0	RW	0x0	gpio7a0_sel GPIO7A[0] iomux select 2'b00: gpio 2'b01: pwm_0 2'b10: vop0_pwm 2'b11: vop1_pwm

GRF_GPIO7B_IOMUX
Address: Operational Base + offset (0x0070)
GPIO7B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio7b7_sel GPIO7B[7] iomux select 2'b00: gpio 2'b01: isp_shuttertrig 2'b10: spi1_txd 2'b11: reserved
13:12	RW	0x0	gpio7b6_sel GPIO7B[6] iomux select 2'b00: gpio 2'b01: isp_prelighttrig 2'b10: spi1_rxd 2'b11: reserved
11:10	RW	0x0	gpio7b5_sel GPIO7B[5] iomux select 2'b00: gpio 2'b01: isp_flashtrigout 2'b10: spi1_csn0 2'b11: reserved
9:8	RW	0x0	gpio7b4_sel GPIO7B[4] iomux select 2'b00: gpio 2'b01: isp_shutteren 2'b10: spi1_clk 2'b11: reserved
7:6	RW	0x0	gpio7b3_sel GPIO7B[3] iomux select 2'b00: gpio 2'b01: usb_drvvbus1 2'b10: edp_hotplug 2'b11: reserved
5:4	RW	0x0	gpio7b2_sel GPIO7B[2] iomux select 2'b00: gpio 2'b01: uart3gps_rtsn 2'b10: usb_drvvbus0 2'b11: reserved
3:2	RW	0x0	gpio7b1_sel GPIO7B[1] iomux select 2'b00: gpio 2'b01: uart3gps_ctsn 2'b10: gps_rfclk 2'b11: gpst1_clk

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio7b0_sel GPIO7B[0] iomux select 2'b00: gpio 2'b01: uart3gps_sout 2'b10: gps_sig 2'b11: hsadct1_data1

GRF_GPIO7CL_IOMUX
Address: Operational Base + offset (0x0074)
GPIO7CL iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable 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	gpio7c3_sel GPIO7C[3] iomux select 2'b00: gpio 2'b01: i2c5hdmi_sda 2'b10: edphdmii2c_sda 2'b11: reserved
11:9	RO	0x0	reserved
8	RW	0x0	gpio7c2_sel GPIO7C[2] iomux select 1'b0: gpio 1'b1: i2c4tp_scl
7:5	RO	0x0	reserved
4	RW	0x0	gpio7c1_sel GPIO7C[1] iomux select 1'b0: gpio 1'b1: i2c4tp_sda
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio7c0_sel GPIO7C[0] iomux select 2'b00: gpio 2'b01: isp_flashtrigin 2'b10: edphdmi_cecinoutt1 2'b11: reserved

GRF_GPIO7CH_IOMUX
Address: Operational Base + offset (0x0078)
GPIO7CH iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable 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	0x0	gpio7c7_sel GPIO7C[7] iomux select 3'b000: gpio 3'b001: uart2dbg_sout 3'b010: uart2dbg_sirout 3'b011: pwm_3 3'b100: edphdmi_cecinout other: reserved
11:10	RO	0x0	reserved
9:8	RW	0x0	gpio7c6_sel GPIO7C[6] iomux select 2'b00: gpio 2'b01: uart2dbg_sin 2'b10: uart2dbg_sirin 2'b11: pwm_2
7:2	RO	0x0	reserved
1:0	RW	0x0	gpio7c4_sel GPIO7C[4] iomux select 2'b00: gpio 2'b01: i2c5hdmi_scl 2'b10: edphdmii2c_scl 2'b11: reserved

GRF_GPIO8A_IOMUX
Address: Operational Base + offset (0x0080)
GPIO8A iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio8a7_sel GPIO8A[7] iomux select 2'b00: gpio 2'b01: spi2_csn0 2'b10: sc_detect 2'b11: reserve
13:12	RW	0x0	gpio8a6_sel GPIO8A[6] iomux select 2'b00: gpio 2'b01: spi2_clk 2'b10: sc_io 2'b11: reserve
11:10	RW	0x0	gpio8a5_sel GPIO8A[5] iomux select 2'b00: gpio 2'b01: i2c2sensor_scl 2'b10: sc_clk 2'b11: reserved
9:8	RW	0x0	gpio8a4_sel GPIO8A[4] iomux select 2'b00: gpio 2'b01: i2c2sensor_sda 2'b10: sc_rst 2'b11: reserved
7:6	RW	0x0	gpio8a3_sel GPIO8A[3] iomux select 2'b00: gpio 2'b01: spi2_csn1 2'b10: sc_iot1 2'b11: reserved
5	RO	0x0	reserved
4	RW	0x0	gpio8a2_sel GPIO8A[2] iomux select 1'b0: gpio 1'b1: sc_detectt1
3:2	RW	0x0	gpio8a1_sel GPIO8A[1] iomux select 2'b00: gpio 2'b01: ps2_data 2'b10: sc_vcc33v 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio8a0_sel GPIO8A[0] iomux select 2'b00: gpio 2'b01: ps2_clk 2'b10: sc_vcc18v 2'b11: reserved

GRF_GPIO8B_IOMUX

Address: Operational Base + offset (0x0084)

GPIO8B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable 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
3:2	RW	0x0	gpio8b1_sel GPIO8B[1] iomux select 2'b00: gpio 2'b01: spi2_txd 2'b10: sc_clk 2'b11: reserve
1:0	RW	0x0	gpio8b0_sel GPIO8B[0] iomux select 2'b00: gpio 2'b01: spi2_rxd 2'b10: sc_rst 2'b11: reserve

GRF_GPIO1H_SR

Address: Operational Base + offset (0x0104)

GPIO1C/D SR control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x0f	gpio1d_sr GPIO1D slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RO	0x0	reserved

GRF_GPIO2L_SR

Address: Operational Base + offset (0x0108)
GPIO2A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2b_sr GPIO2B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio2a_sr GPIO2A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO2H_SR
Address: Operational Base + offset (0x010c)
GPIO2C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio2c_sr GPIO2C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO3L_SR
Address: Operational Base + offset (0x0110)
GPIO3A/B SR control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x20	gpio3b_sr GPIO3B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0xff	gpio3a_sr GPIO3A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO3H_SR
Address: Operational Base + offset (0x0114)
GPIO3C/D SR control

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

Bit	Attr	Reset Value	Description
15:8	RW	0xff	gpio3d_sr GPIO3D slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x04	gpio3c_sr GPIO3C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO4L_SR
Address: Operational Base + offset (0x0118)
GPIO4A/B SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x01	gpio4b_sr GPIO4B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x20	gpio4a_sr GPIO4A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO4H_SR
Address: Operational Base + offset (0x011c)
GPIO4C/D SR control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4d_sr GPIO4D slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x00	gpio4c_sr GPIO4C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO5L_SR

Address: Operational Base + offset (0x0120)
GPIO5A/B SR control

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

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio5b_sr GPIO5B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RO	0x0	reserved

GRF_GPIO5H_SR
Address: Operational Base + offset (0x0124)
GPIO5C/D SR control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio5c_sr GPIO5C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO6L_SR
Address: Operational Base + offset (0x0128)
GPIO6A/B SR control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x01	gpio6b_sr GPIO6B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x00	gpio6a_sr GPIO6A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO6H_SR
Address: Operational Base + offset (0x012c)
GPIO6C/D SR control

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

Bit	Attr	Reset Value	Description
7:0	RW	0x10	gpio6c_sr GPIO6C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO7L_SR
Address: Operational Base + offset (0x0130)
GPIO7A/B SR control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio7b_sr GPIO7B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast
7:0	RW	0x00	gpio7a_sr GPIO7A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO7H_SR
Address: Operational Base + offset (0x0134)
GPIO7C/D SR control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	gpio7c_sr GPIO7C slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO8L_SR

Address: Operational Base + offset (0x0138)
GPIO8A/B SR control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio8b_sr GPIO8B slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio8a_sr GPIO8A slew rate control for each bit 1'b0: slow (half frequency) 1'b1: fast

GRF_GPIO1D_P

Address: Operational Base + offset (0x014c)

GPIO1D PU/PD control

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

15:0	RW	0xaaaa	gpio1d_p
			GPIO1D PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2A_P

Address: Operational Base + offset (0x0150)

GPIO2A PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0xaaaa	gpio2a_p
			GPIO2A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2B_P
Address: Operational Base + offset (0x0154)
GPIO2B PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0xaaaa	gpio2b_p
			GPIO2B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO2C_P
Address: Operational Base + offset (0x0158)
GPIO2C PU/PD control

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

15:0	RW	0xaaa5	gpio2c_p
			GPIO2C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3A_P
Address: Operational Base + offset (0x0160)
GPIO3A PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0x5555	gpio3a_p
			GPIO3A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3B_P
Address: Operational Base + offset (0x0164)
GPIO3B PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0x5699	gpio3b_p
			GPIO3B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3C_P
Address: Operational Base + offset (0x0168)
GPIO3C PU/PD control

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

15:0	RW	0xaaa5	gpio3c_p
			GPIO3C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO3D_P
Address: Operational Base + offset (0x016c)
GPIO3D PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0x5555	gpio3d_p
			GPIO3D PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO4A_P
Address: Operational Base + offset (0x0170)
GPIO4A PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0x5555	gpio4a_p
			GPIO4A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO4B_P
Address: Operational Base + offset (0x0174)
GPIO4B PU/PD control

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

15:0	RW	0xaaa5	gpio4b_p
			GPIO4B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO4C_P
Address: Operational Base + offset (0x0178)
GPIO4C PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0x5559	gpio4c_p
			GPIO4C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO4D_P
Address: Operational Base + offset (0x017c)
GPIO4D PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0x5a99	gpio4d_p
			GPIO4D PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO5B_P
Address: Operational Base + offset (0x0184)
GPIO5B PU/PD control

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

15:0	RW	0x6559	gpio5b_p
			GPIO5B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO5C_P
Address: Operational Base + offset (0x0188)
GPIO5C PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0xaaa9	gpio5c_p
			GPIO5C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO6A_P
Address: Operational Base + offset (0x0190)
GPIO6A PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0xaaaa	gpio6a_p
			GPIO6A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO6B_P
Address: Operational Base + offset (0x0194)
GPIO6B PU/PD control

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

15:0	RW	0xaa96	gpio6b_p
			GPIO6B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO6C_P
Address: Operational Base + offset (0x0198)
GPIO6C PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0x5655	gpio6c_p
			GPIO6C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO7A_P

Address: Operational Base + offset (0x01a0)

GPIO7A PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0x59aa	gpio7a_p
			GPIO7A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO7B_P
Address: Operational Base + offset (0x01a4)
GPIO7B PU/PD control

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

15:0	RW	0xa696	gpio7b_p
			GPIO7B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO7C_P
Address: Operational Base + offset (0x01a8)
GPIO7C PU/PD control

Bit	Attr	Reset Value	Description

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

15:0	RW	0x5955	gpio7c_p
			GPIO7C PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO8A_P
Address: Operational Base + offset (0x01b0)
GPIO8A PU/PD control

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

Bit	Attr	Reset Value	Description

15:0	RW	0x6555	gpio8a_p
			GPIO8A PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO8B_P
Address: Operational Base + offset (0x01b4)
GPIO8B PU/PD control

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

15:0	RW	0xaaaa	gpio8b_p
			GPIO8B PU/PD programmation section, every GPIO bit corresponding to 2bits
			2'b00: Z(Noraml operaton);
			2'b01: weak 1(pull-up);
			2'b10: weak 0(pull-down);
			2'b11: Repeater(Bus keeper)

GRF_GPIO1D_E
Address: Operational Base + offset (0x01cc)
GPIO1D drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x55aa	gpio1d_e 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 (0x01d0)
GPIO2A drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0xaaaa	gpio2a_e 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 (0x01d4)
GPIO2B drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xaaaa	gpio2b_e 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 (0x01d8)
GPIO2C drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio2c_e GPIO2C 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 (0x01e0)
GPIO3A drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0xaaaa	gpio3a_e 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 (0x01e4)
GPIO3B drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5955	gpio3b_e 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 (0x01e8)
GPIO3C drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5565	gpio3c_e GPIO3C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3D_E
Address: Operational Base + offset (0x01ec)
GPIO3D drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0xaaaa	gpio3d_e GPIO3D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4A_E
Address: Operational Base + offset (0x01f0)
GPIO4A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5955	gpio4a_e GPIO4A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4B_E
Address: Operational Base + offset (0x01f4)
GPIO4B drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5556	gpio4b_e GPIO4B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4C_E
Address: Operational Base + offset (0x01f8)
GPIO4C drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0x5555	gpio4c_e GPIO4C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4D_E
Address: Operational Base + offset (0x01fc)
GPIO4D drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio4d_e GPIO4D drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO5B_E
Address: Operational Base + offset (0x0204)
GPIO5B drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio5b_e GPIO5B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO5C_E
Address: Operational Base + offset (0x0208)
GPIO5C drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0x5555	gpio5c_e GPIO5C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO6A_E
Address: Operational Base + offset (0x0210)
GPIO6A drive strength control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio6a_e GPIO6A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO6B_E
Address: Operational Base + offset (0x0214)
GPIO6B drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio6b_e GPIO6B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO6C_E
Address: Operational Base + offset (0x0218)
GPIO6C drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0x5555	gpio6c_e GPIO6C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO7A_E
Address: Operational Base + offset (0x0220)
GPIO7A drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio7a_e GPIO7A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO7B_E
Address: Operational Base + offset (0x0224)
GPIO7B drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio7b_e GPIO7B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO7C_E
Address: Operational Base + offset (0x0228)
GPIO7C drive strength control

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

Bit	Attr	Reset Value	Description
15:0	RW	0x5555	gpio7c_e GPIO7C drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO8A_E
Address: Operational Base + offset (0x0230)
GPIO8A drive strength control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio8a_e GPIO8A drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO8B_E
Address: Operational Base + offset (0x0234)
GPIO8B drive strength control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x5555	gpio8b_e GPIO8B drive strength control, every GPIO bit corresponding to 2bits 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO_SMT
Address: Operational Base + offset (0x0240)
GPIO smitter control register

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

Bit	Attr	Reset Value	Description
11	RW	0x1	gpio8a1_smt GPIO8A_1 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
10	RW	0x1	gpio8a0_smt GPIO8A_0 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
9	RW	0x1	gpio7c4_smt GPIO7C_4 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
8	RW	0x1	gpio7c3_smt GPIO7C_3 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
7	RW	0x1	gpio7c2_smt GPIO7C_2 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x1	gpio7c1_smt GPIO7C_1 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x1	gpio2c1_smt GPIO2C_1 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x1	gpio2c0_smt GPIO2C_0 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x1	gpio6b2_smt GPIO6B_2 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x1	gpio6b1_smt GPIO6B_1 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x1	gpio8a5_smt GPIO8A_5 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
0	RW	0x1	gpio8a4_smt GPIO8A_4 SMT control 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_SOC_CON0
Address: Operational Base + offset (0x0244)
SoC control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	pause_mmc_peri PERI MMC AHB bus arbiter pause control
14	RW	0x0	pause_emem_peri PERI EMEM AHB bus arbiter pause control
13	RW	0x0	pause_usb_peri PERI USB AHB bus arbiter pause control
12	RW	0x1	grf_force_jtag Force select jtag function from sdmmc0 IO
11	RW	0x1	grf_core_idle_req_mode_sel1 core idle request mode selection 1
10	RW	0x1	grf_core_idle_req_mode_sel0 core idle request mode selection 0
9	RW	0x0	ddr1_16bit_en DDR Channel 1 interface 16bit enable
8	RW	0x0	ddr0_16bit_en DDR Channel 0 interface 16bit enable
7	RW	0x0	vcodec_sel vdpu vepu clock select 1'b0: select vepu aclk as vcodec main clock 1'b1: select vdpu aclk as vcodec main clock

Bit	Attr	Reset Value	Description
6	RW	0x0	upctl1_c_active_in Channel 1 DDR clock active in External signal from system that flags if a hardware low power request can be accepted or should always be denied. 1'b0: may be accepted 1'b1: will be denied
5	RW	0x0	upctl0_c_active_in Channel 0 DDR clock active in External signal from system that flags if a hardware low power request can be accepted or should always be denied. 1'b0: may be accepted 1'b1: will be denied
4	RW	0x1	msch1_mainddr3 Channel 1 DDR3 mode control 1'b1: DDR3 mode
3	RW	0x1	msch0_mainddr3 Channel 0 DDR3 mode control 1'b1: DDR3 mode
2	RW	0x0	msch1_mainpartialpop msch1_mainpartialpop bit control
1	RW	0x0	msch0_mainpartialpop msch0_mainpartialpop bit control
0	RO	0x0	reserved

GRF_SOC_CON1
Address: Operational Base + offset (0x0248)
SoC control register 1

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

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x1	rmii_mode RMII mode selection 1'b1: RMII mode
13:12	RW	0x0	gmac_clk_sel RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
11	RW	0x0	rmii_clk_sel RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
10	RW	0x0	gmac_speed MAC speed 1'b1: 100-Mbps 1'b0: 10-Mbps
9	RW	0x0	gmac_flowctrl 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
8:6	RW	0x1	gmac_phy_intf_sel PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
5	RW	0x0	host_remap Host interface remap control
4:0	RO	0x0	reserved

GRF_SOC_CON2
Address: Operational Base + offset (0x024c)
SoC control register 2

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RO	0x0	reserved
13	RW	0x0	upctl1_lpddr3_odt_en Channel 1 DDR odt enable in LPDDR3 mode 1'b1: ODT enable 1'b0: ODT disable
12	RW	0x0	upctl1_bst_diable Channel 1 DDR controller burst termination disable control 1'b1: disable 1'b0: enable
11	RW	0x0	lpddr3_en1 Channel 1 LPDDR3 mode control 1'b1: LPDDR3 mode
10	RW	0x0	upctl0_lpddr3_odt_en Channel 0 DDR odt enable in LPDDR3 mode 1'b1: ODT enable 1'b0: ODT disable
9	RW	0x0	upctl0_bst_diable Channel 0 DDR controller burst termination disable control 1'b1: disable 1'b0: enable
8	RW	0x0	lpddr3_en0 Channel 0 LPDDR3 mode control 1'b1: LPDDR3 mode

Bit	Attr	Reset Value	Description
7	RW	0x0	grf_poc_flash0_ctrl Flash0 IO domain 1.8V selection source 1'b0: GPIO3C_3 to decide the flash0 IO domain voltage, when GPIO3C_3 high, the voltage is 1.8V 1'b1: grf_io_vsel[2] to decide the flash0 IO domain voltage, when grf_io_vsel[2] high, the voltage is 1.8V
6	RW	0x0	simcard_mux_sel sim card iomux solution selection 1'b1: use GPIO8A[5:2] 1'b0: use GPIO8A[7:6] and GPIO8B[1:0]
5:2	RO	0x0	reserved
1	RW	0x1	grf_spdif_2ch_en SPDIF solution selection 1'b0: 8CH SPDIF 1'b1: 2CH SPDIF
0	RW	0x0	pwm_sel PWM solution selection 1'b1: RK PWM 1'b0: PWM(old)

GRF_SOC_CON3
Address: Operational Base + offset (0x0250)
SoC control register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	rxclk_dly_ena_gmac RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
14	RW	0x0	txclk_dly_ena_gmac RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
13:7	RW	0x10	clk_rx_dl_cfg_gmac RGMII RX clock delayline value
6:0	RW	0x10	clk_tx_dl_cfg_gmac RGMII TX clock delayline value

GRF_SOC_CON4
Address: Operational Base + offset (0x0254)
SoC control register 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	dfi_eff_stat_en1 Channel 1 DFI monitor efficiency statistics enable
14	RW	0x0	dfi_eff_stat_en0 Channel 0 DFI monitor efficiency statistics enable
13	RW	0x0	mobile_ddr_sel Mobile DDR selection in DFI monitor 1'b1: mobile DDR(LPDDR2/LPDDR3) 1'b0: DDR3
12:10	RW	0x1	host_l3_ocp_sconnect_grf Host interface l3_ocp_sconnect signal control
9:8	RW	0x2	host_txport_rst_val_grf Host interface txport_rst_val signal control
7:6	RW	0x0	host_rxport_rst_val_grf Host interface rxport_rst_val signal control

Bit	Attr	Reset Value	Description
5	RW	0x0	host_wakereq_grf Host interface wakereq signal control
4:3	RW	0x0	host_eoi_in_grf Host interface eoi_in signal control
2	RW	0x1	host_mstandy_in_grf Host interface mstandy_in signal control
1	RW	0x1	host_mwait_grf Host interface mwait signal control
0	RW	0x1	host_sidle_req_grf Host interface sidle_req signal control

GRF_SOC_CON5
Address: Operational Base + offset (0x0258)
SoC control register 5

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	host_mux_sel Host interface mux selection 1'b1: 8bits input, 16bits output 1'b0: 8bits output, 16bits input
14	RW	0x0	tsp0_inout_sel TSP0 input/output selection 1'b1: output 1'b0: input
13	RW	0x0	hsadc_clkout_en hsadc clkout enable 1'b1: hsadc_clkout 1'b0: gps_clk
12:3	RW	0x190	host_fclk_freq_rst_val_grf Host interface fclk_freq_rst_val signal control

Bit	Attr	Reset Value	Description
2:1	RW	0x3	host_l3_iocp_mconnect_grf Host interface l3_iocp_mconnect signal control
0	RW	0x1	host_l3_ocp_mdiscbehave_grf Host interface l3_ocp_mdiscbehave signal control

GRF_SOC_CON6
Address: Operational Base + offset (0x025c)
SoC control register 6

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	grf_hdmi_edp_sel HDMI source selection 1'b1: from HDMI controller 1'b0: from eDP controller
14	RW	0x0	dsi_csi_testbus_sel MIPI PHY TX1RX1 test bus source selection 1'b1: CSI host 1'b0: DSI host1
13	RW	0x0	hsadc_extclk_mux_sel HSADC external clock source selection 1'b1: GPIO7B[1] 1'b0: GPIO2B[2]
12	RW	0x0	clk_27m_mux_sel 27M clock input source selection 1'b1: GPIO0C[1] 1'b0: GPIO0B[5]
11	RW	0x0	grf_con_dsi1_dpicolorm DSI host1 dpicolorm bit control

Bit	Attr	Reset Value	Description
10	RW	0x0	grf_con_dsi1_dpishutdn DSI host1 dpishutdn bit control
9	RW	0x0	grf_con_dsi1_lcdc_sel DSI host1 data from VOP selection 1'b1: VOP LIT output to DSI host1 1'b0: VOP BIG output to DSI host1
8	RW	0x0	grf_con_dsi0_dpicolorm DSI host0 dpicolorm bit control
7	RW	0x0	grf_con_dsi0_dpishutdn DSI host0 dpishutdn bit control
6	RW	0x0	grf_con_dsi0_lcdc_sel DSI host0 data from VOP selection 1'b1: VOP LIT output to DSI host0 1'b0: VOP BIG output to DSI host0
5	RW	0x0	grf_con_edp_lcdc_sel eDP data from VOP selection 1'b1: VOP LIT output to eDP 1'b0: VOP BIG output to eDP
4	RW	0x0	grf_con_hdmi_lcdc_sel HDMI data from VOP selection 1'b1: VOP LIT output to HDMI 1'b0: VOP BIG output to HDMI
3	RW	0x0	grf_con_lvds_lcdc_sel LVDS data from VOP selection 1'b1: VOP LIT output to LVDS 1'b0: VOP BIG output to LVDS
2	RW	0x0	grf_con_iep_vop_sel IEP connect to VOP selection 1'b1: IEP connect to VOP LIT 1'b0: IEP connect to VOP BIG
1	RW	0x0	grf_con_isp_dphy_sel ISP connect to MIPI PHY selection 1'b1: MIPI PHY TX1RX1 1'b0: MIPI PHY RX0
0	RW	0x0	grf_con_disable_isp Disable ISP control

GRF_SOC_CON7
Address: Operational Base + offset (0x0260)
SoC control register 7

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_lvds_pwrdwn LVDS PHY power down control 1'b1: power down 1'b0: power up
14	RO	0x0	reserved
13	RW	0x0	grf_lvds_lcdc_trace_sel LVDS IO used as trace bus enable 1'b1: used as trace bus 1'b0: used as LVDS IO or LCDC RGB output port
12	RW	0x0	grf_lvds_con_enable_2 LVDS controller enable_2 signal control
11	RW	0x0	grf_lvds_con_enable_1 LVDS controller enable_1 signal control
10	RW	0x0	grf_lvds_con_den_polarity LVDS controller den_polarity signal control
9	RW	0x0	grf_lvds_con_hs_polarity LVDS controller hs_polarity signal control
8	RW	0x0	grf_lvds_con_clkinv LVDS controller clkinv signal control
7	RW	0x0	grf_lvds_con_startphase LVDS controller startphase signal control
6	RW	0x0	grf_lvds_con_ttl_en LVDS controller ttl_en signal control
5	RW	0x0	grf_lvds_con_startsel LVDS controller startsel signal control
4	RW	0x0	grf_lvds_con_chasel LVDS controller chasel signal control
3	RW	0x0	grf_lvds_con_msbsel LVDS controller msbsel signal control

Bit	Attr	Reset Value	Description
2:0	RW	0x0	grf_lvds_con_select LVDS controller select signal control

GRF_SOC_CON8
Address: Operational Base + offset (0x0264)
SoC control register 8

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_edp_hdcp_protect eDP HDCP function protection 1'b1: protect 1'b0: not protect
14	RW	0x0	grf_edp_bist_en eDP PHY BIST function enabled 1'b1: enable 1'b0: disable
13	RW	0x0	grf_edp_mem_ctrl_sel eDP memory control selection 1'b1: controlled by eDP controller internal logic 1'b0: controlled by APB BUS
12	RW	0x0	grf_hdmi_cec_mux_sel HDMI cec source selection 1'b1: from GPIO7C[0] 1'b0: from GPIO7C[7]
11:8	RW	0x0	grf_dphy_tx0_forcetxstopmode MIPI DPHY TX0 force lane into transmit mode and generate stop sate. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

Bit	Attr	Reset Value	Description
7:4	RW	0x0	grf_dphy_tx0_forcerxmode MIPI DPHY TX0 force lane into receive mode/wait for stop stat. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
3:0	RW	0xe	grf_dphy_tx0_turndisable MIPI DPHY TX0 disable turn around control Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

GRF_SOC_CON9
Address: Operational Base + offset (0x0268)
SoC control register 9

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	grf_dphy_tx1rx1_enable MIPI DPHY TX1RX1 enable lane N module(N=0~3). Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
11:8	RW	0x0	grf_dphy_tx1rx1_forcetxstopmode MIPI DPHY TX1RX1 force lane into transmit mode and generate stop sate. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
7:4	RW	0x0	grf_dphy_tx1rx1_forcerxmode MIPI DPHY TX1RX1 force lane into receive mode/wait for stop stat. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

Bit	Attr	Reset Value	Description
3:0	RW	0xe	grf_dphy_tx1rx1_turndisable MIPI DPHY TX1RX1 disable turn around control Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

GRF_SOC_CON10
Address: Operational Base + offset (0x026c)
SoC control register 10

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	grf_dphy_rx0_enable MIPI DPHY RX0 enable lane N module(N=0~3). Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
11:8	RW	0x0	grf_dphy_rx0_forcetxstopmode MIPI DPHY RX0 force lane into transmit mode and generate stop sate. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
7:4	RW	0x0	grf_dphy_rx0_forcerxmode MIPI DPHY RX0 force lane into receive mode/wait for stop stat. Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
3:0	RW	0xf	grf_dphy_rx0_turndisable MIPI DPHY RX0 disable turn around control Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

GRF_SOC_CON11
Address: Operational Base + offset (0x0270)
SoC control register 11

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	gpio8_a2_fall_edge_irq_pd GIIO8A[2] fall edge interrupt pending status 1'b1: enable 1'b0: disable
14	RW	0x0	gpio8_a2_fall_edge_irq_en GIIO8A[2] fall edge interrupt enable 1'b1: enable 1'b0: disable
13	RW	0x0	gpio8_a2_rise_edge_irq_pd GIIO8A[2] rise edge interrupt pending status 1'b1: enable 1'b0: disable
12	RW	0x0	gpio8_a2_rise_edge_irq_en GIIO8A[2] rise edge interrupt enable 1'b1: enable 1'b0: disable
11	RW	0x0	gpio7_c6_fall_edge_irq_pd GIIO7C[6] fall edge interrupt pending status 1'b1: enable 1'b0: disable
10	RW	0x0	gpio7_c6_fall_edge_irq_en GIIO7C[6] fall edge interrupt enable 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
9	RW	0x0	gpio7_c6_rise_edge_irq_pd GIIO7C[6] rise edge interrupt pending status 1'b1: enable 1'b0: disable
8	RW	0x0	gpio7_c6_rise_edge_irq_en GIIO7C[6] rise edge interrupt enable 1'b1: enable 1'b0: disable
7	RW	0x0	gpio7_b3_fall_edge_irq_pd GIIO7B[3] fall edge interrupt pending status 1'b1: enable 1'b0: disable
6	RW	0x0	gpio7_b3_fall_edge_irq_en GIIO7B[3] fall edge interrupt enable 1'b1: enable 1'b0: disable
5	RW	0x0	gpio7_b3_rise_edge_irq_pd GIIO7B[3] rise edge interrupt pending status 1'b1: enable 1'b0: disable
4	RW	0x0	gpio7_b3_rise_edge_irq_en GIIO7B[3] rise edge interrupt enable 1'b1: enable 1'b0: disable
3	RW	0x0	sd_detectn_fall_edge_irq_pd sdmmc detect_n fall edge interrupt pending status 1'b1: enable 1'b0: disable
2	RW	0x0	sd_detectn_fall_edge_irq_en sdmmc0 detect_n signal fall edge interrupt enable 1'b1: enable 1'b0: disable
1	RW	0x0	sd_detectn_rise_edge_irq_pd sdmmc detect_n rise edge interrupt pending status 1'b1: enable 1'b0: disable
0	RW	0x0	sd_detectn_rise_edge_irq_en sdmmc0 detect_n signal rise edge interrupt enable 1'b1: enable 1'b0: disable

GRF_SOC_CON12
Address: Operational Base + offset (0x0274)
SoC control register 12

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RW	0x000	grf_edp_frq_vid_ck_in eDP PHY frequency information of vid_ck_in frq_vid_ck_in<8:0>/8 = freq(vid_ck_in)/10
6	RW	0x0	grf_edp_vid_lock eDP PHY input video PLL stable indicator 1'b1: stable 1'b0: unstable
5	RW	0x0	grf_edp_iddq_en eDP PHY IDDQ enable 1'b0: disable 1'b1: enable, all circuits are power down, all IO are high-z
4	RW	0x1	grf_edp_ref_clk_sel eDP PHY reference clock source selection 1'b0: from PAD(IO_EDP_OSC_CLK_24M) 1'b1: from internal 24MHz or 27MHz clock
3	RW	0x0	grf_edp_dc_tp_i eDP PHY analog DC test point input
2	RO	0x0	reserved
1:0	RW	0x3	grf_filter_cnt_sel the counter select for sd card detect filter 2'b00: 5ms 2'b01: 15ms 2'b10: 35ms 2'b11: 50ms

GRF_SOC_CON13
Address: Operational Base + offset (0x0278)
SoC control register 13

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RW	0x0	grf_edp_tx_bscan_data eDP TX boundary data bit0: boundary data to ch0 bit1: boundary data to ch1 bit2: boundary data to ch2 bit3: boundary data to ch3
11	RW	0x0	grf_edp_tx_bscan_en eDP TX boundary enable 1'b0: disable 1'b1: enable
10	RO	0x0	reserved
9:5	RW	0x00	grf_uart_rts_sel UART polarity selection for rts port Every bit for one UART, bit4 is for UART_EXP, bit3 is for UART_GPS, bit2 is for UART_DBG, bit1 is for UART_BB, bit0 is for UART_BT. 1'b1: high asserted 1'b0: low asserted
4:0	RW	0x00	grf_uart_cts_sel UART polarity selection for cts port Every bit for one UART, bit4 is for UART_EXP, bit3 is for UART_GPS, bit2 is for UART_DBG, bit1 is for UART_BB, bit0 is for UART_BT. 1'b1: high asserted 1'b0: low asserted

GRF_SOC_CON14

Address: Operational Base + offset (0x027c)
SoC control register 14

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_dphy_tx1rx1_basedir MIPI DPHY TX1RX1 base direction control
14	RW	0x0	grf_dphy_tx1rx1_masterslavez MIPI DPHY TX1RX1 master/slave control
13	RW	0x0	dphy_rx1_src_sel MIPI DPHY RX1 source selection 1'b1: isp 1'b0: csi host
12	RW	0x0	dphy_tx1rx1_enableclk MIPI DPHY TX1RX1 enable clock Lane module
11	RO	0x0	reserved
10:3	RW	0x00	dphy_rx0_testdin MIPI DPHY RX0 test bus input data
2	RW	0x0	dphy_rx0_testen MIPI DPHY RX0 test bus enable
1	RW	0x0	dphy_rx0_testclk MIPI DPHY RX0 test bus clock
0	RW	0x0	dphy_rx0_testclr MIPI DPHY RX0 test bus clear control

GRF_SOC_STATUS0
Address: Operational Base + offset (0x0280)
SoC status register 0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	ddrupctl1_bbflags DDR channel 1 NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes. Bit0 indication Bank0 busy, bit1 indication Bank1 busy, and so on.
15:0	RW	0x0000	ddrupctl0_bbflags DDR channel 0 NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes. Bit0 indication Bank0 busy, bit1 indication Bank1 busy, and so on.

GRF_SOC_STATUS1
Address: Operational Base + offset (0x0284)
SoC status register 1

Bit	Attr	Reset Value	Description
31	RW	0x0	gmac_portselect MAC Port Select A high indicates an MII interface, and a low a GMII interface.
30:26	RO	0x0	reserved
25:22	RW	0x0	reserved
21:16	RW	0x00	reserved
15:13	RW	0x0	ddrupctl1_stat 3'b000: Init_mem 3'b001: Config 3'b010: Config_req 3'b011: Access 3'b100: Access_req 3'b101: Low_power 3'b110: Low_power_entry_req 3'b111: Low_power_exit_req

Bit	Attr	Reset Value	Description
12:10	RW	0x0	ddrupctl0_stat 3'b000: Init_mem 3'b001: Config 3'b010: Config_req 3'b011: Access 3'b100: Access_req 3'b101: Low_power 3'b110: Low_power_entry_req 3'b111: Low_power_exit_req
9	RW	0x0	newpll_lock NEW PLL lock status
8	RW	0x0	generalpll_lock GENERAL PLL lock status
7	RW	0x0	codecpll_lock CODEC PLL lock status
6	RW	0x0	armpll_lock ARM PLL lock status
5	RW	0x0	ddrpll_lock DDR PLL lock status
4	RW	0x0	newpll_clk NEW PLL clock output
3	RW	0x0	generalpll_clk GENERAL PLL clock output
2	RW	0x0	codecpll_clk CODEC PLL clock output
1	RW	0x0	armpll_clk ARM PLL clock output
0	RW	0x0	ddrpll_clk DDR PLL clock output

GRF_SOC_STATUS2
Address: Operational Base + offset (0x0288)
SoC status register 2

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	usbhost0_stat_ohci_bufacc USB HOST0 ohci_bufacc signal status
29	RW	0x0	usbhost0_stat_ohci_rmtwkp USB HOST0 ohci_rmtwkp signal status
28:27	RW	0x0	usbhost0_utmi_linestate USB HOST0 utmi_linestate signal status
26	RW	0x0	usbhost0_stat_ohci_drwe USB HOST0 ohci_drwe signal status

Bit	Attr	Reset Value	Description
25	RW	0x0	usbhost0_stat_ohci_rwe USB HOST0 ohci_rwe signal status
24	RW	0x0	usbhost0_stat_ohci_ccs USB HOST0 ohci_ccs signal status
23	RW	0x0	usbhost1_utmiotg_iddig USB HOST1 utmiotg_iddig signal status
22:21	RW	0x0	usbhost1_utmi_linestate USB HOST1 utmi_linestate signal status
20	RW	0x0	usbhost1_utmisrp_bvalid USB HOST1 utmisrp_bvalid signal status
19	RW	0x0	usbhost1_utmiotg_vbusvalid USB HOST1 utmiotg_vbusvalid signal status
18	RW	0x0	usbhost1_chirp_on USB HOST1 chirp_on signal status
17	RW	0x0	usbotg_utmiotg_iddiq USB OTG utmiotg_iddig signal status
16:15	RW	0x0	usbotg_utmi_linestate USB OTG utmi_linestate signal status
14	RW	0x0	usbotg_utmisrp_bvalid USB OTG utmisrp_bvalid
13	RW	0x0	usbotg_utmiotg_vbusvalid USB OTG utmiotg_vbusvalid signal status
12	RO	0x0	reserved
11	RW	0x0	reserved
10:0	RW	0x000	reserved

GRF_SOC_STATUS3
Address: Operational Base + offset (0x028c)
SoC status register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo0 DDR channel0 NIF interface FIFO0 status

GRF_SOC_STATUS4
Address: Operational Base + offset (0x0290)
SoC status register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo1 DDR channel0 NIF interface FIFO1 status

GRF_SOC_STATUS5
Address: Operational Base + offset (0x0294)
SoC status register 5

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo2 DDR channel0 NIF interface FIFO2 status

GRF_SOC_STATUS6
Address: Operational Base + offset (0x0298)
SoC status register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif0_fifo3 DDR channel0 NIF interface FIFO3 status

GRF_SOC_STATUS7
Address: Operational Base + offset (0x029c)
SoC status register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif1_fifo0 DDR channel1 NIF interface FIFO0 status

GRF_SOC_STATUS8
Address: Operational Base + offset (0x02a0)
SoC status register 8

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif1_fifo1 DDR channel1 NIF interface FIFO1 status

GRF_SOC_STATUS9
Address: Operational Base + offset (0x02a4)
SoC status register 9

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif1_fifo2 DDR channel1 NIF interface FIFO2 status

GRF_SOC_STATUS10
Address: Operational Base + offset (0x02a8)
SoC status register 10

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	nif1_fifo3 DDR channel1 NIF interface FIFO3 status

GRF_SOC_STATUS11
Address: Operational Base + offset (0x02ac)
SoC status register 11

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi0_eff_wr_num DDR channel0 DFI interface write command number

GRF_SOC_STATUS12
Address: Operational Base + offset (0x02b0)
SoC status register 12

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi0_eff_rd_num DDR channel0 DFI interface read command number

GRF_SOC_STATUS13
Address: Operational Base + offset (0x02b4)
SoC status register 13

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi0_eff_act_num DDR channel0 DFI interface active command number

GRF_SOC_STATUS14
Address: Operational Base + offset (0x02b8)
SoC status register 14

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi0_timer_val DDR channel0 DFI interface statistics timer value

GRF_SOC_STATUS15
Address: Operational Base + offset (0x02bc)
SoC status register 15

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi1_eff_wr_num DDR channel1 DFI interface write command number

GRF_SOC_STATUS16
Address: Operational Base + offset (0x02c0)
SoC status register 16

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi1_eff_rd_num DDR channel1 DFI interface read command number

GRF_SOC_STATUS17

Address: Operational Base + offset (0x02c4)

SoC status register 17

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi1_eff_act_num DDR channel1 DFI interface active command number

GRF_SOC_STATUS18

Address: Operational Base + offset (0x02c8)

SoC status register 18

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi1_timer_val DDR channel1 DFI interface statistics timer value

GRF_SOC_STATUS19

Address: Operational Base + offset (0x02cc)

SoC status register 19

Bit	Attr	Reset Value	Description
31	RW	0x0	usbhost1_fsvminus USB HOST1 PHY fsvminus bit status
30	RW	0x0	usbhost1_fsvplus USB HOST1 PHY fsvplus bit status
29	RW	0x0	usbhost1_chgdet USB HOST1 PHY charge detect status
28	RW	0x0	usbhost0_fsvminus USB HOST0 PHY fsvminus bit status
27	RW	0x0	usbhost0_fsvplus USB HOST0 PHY fsvplus bit status
26	RW	0x0	usbhost0_chgdet USB HOST0 PHY charge detect status
25	RW	0x0	usbotg_fsvminus USB OTG PHY fsvminus bit status
24	RW	0x0	usbotg_fsvplus USB OTG PHY fsvplus bit status
23	RW	0x0	usbotg_chgdet USB OTG PHY charge detect status
22:14	RO	0x0	reserved
13:11	RW	0x0	host_l3_ocp_sconnect Host interface L3 OCP sconnect status
10	RW	0x0	host_l3_ocp_tactive Host interface L3 OCP tactive status
9:8	RW	0x0	host_l3_ocp_mconnect Host interface L3 OCP mconnect status

Bit	Attr	Reset Value	Description
7	RW	0x0	host_wakeack Host interface wakeack status
6:5	RW	0x0	host_eoi_out Host interface eoi_out status
4	RW	0x0	host_mwait_out Host interface mwait_out status
3	RW	0x0	host_mwakeup Host interface mwakeup status
2	RW	0x0	host_mstandby Host interface mstandby status
1:0	RW	0x0	host_sidle_ack Host interface sidle ack

GRF_SOC_STATUS20
Address: Operational Base + offset (0x02d0)
SoC status register 20

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	host_geno Host interface geno bit stauts The GENI GENO is a mechanism that allows the 2 chip to exchange flags (interupts), up to 32 independent flags are available.

GRF_SOC_STATUS21
Address: Operational Base + offset (0x02d4)
SoC status register 21

Bit	Attr	Reset Value	Description
31:26	RW	0x00	usbhost0_stat_ehci_usbsts USB host0 ehci_usbsts bit status
25:15	RW	0x000	usbhost0_stat_ehci_xfer_cnt USB host0 ehci_xfer counter status
14	RW	0x0	usbhost0_stat_ehci_xfer_prdc USB host0 ehci_xfer_prdc bit status
13:10	RW	0x0	usbhost0_stat_ehci_lpsmc_state USB host0 ehci_lpsmc_state bit status
9	RW	0x0	usbhost0_stat_ehci_bufacc USB host0 ehci_bufacc bit status
8	RW	0x0	usbhost0_stat_ohci_globalsuspend USB host0 ohci_globalsuspend bit status
7:0	RW	0x00	dphy_rx0_testdout MIPI DPHY RX0 test bus data output

GRF_PERIDMAC_CON0
Address: Operational Base + offset (0x02e0)

PERI DMAC control register 0

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	wirte_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	peridmac_boot_addr peridmac_boot_addr[19:12] PERI DMAC boot_addr[19:12] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.
7:4	RW	0xf	peridmac_boot_periph_ns peridmac_boot_periph_ns[19:16] PERI DMAC boot_peri_ns input control Controls the security state of a peripheral request interface, when the PERI DMAC exits from reset. Note: PERI DMAC don't support secure feature, these bits don't need to be configured
3	RW	0x1	peridmac_boot_manager_ns PERI DMAC boot_manager_ns input control When the DMAC exits from reset , this signal controls the security state of the DMA manager thread: 1'b0: assigns DMA manager to the secure state 1'b1: assigns DMA manager to the Non-secure state

Bit	Attr	Reset Value	Description
2:1	RW	0x1	grf_drtype_peridmac PERI DMAC type of acknowledgement or request for peripheral signals: 2'b00: single level request 2'b01: burst level request 2'b10: acknowledging a flush request 2'b11: reserved
0	RW	0x0	peridmac_boot_from_pc PERI DMAC boot_from_pc input control Controls the location in which the DMAC0 executes its initial instruction, after it exits from reset : 1'b0: DMAC waits for an instruction from APB interface 1'b1: DMAC manager thread executes the instruction that is located at the address that boot_addr[31:0] provided.

GRF_PERIDMAC_CON1
Address: Operational Base + offset (0x02e4)
PERI DMAC control register 1

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	wirte_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:0	RW	0x000	peridmac_boot_addr peridmac_boot_addr[31:20] PERI DMAC boot_addr[31:20] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

GRF_PERIDMAC_CON2
Address: Operational Base + offset (0x02e8)
PERI DMAC control register 2

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	wirte_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xffff	peridmac_boot_irq_ns PERI DMAC boot_irq_ns input control Controls the security state of an event-interrupt resource , when the PERI DMAC exits from reset. Note : PERI DMAC don't support secure feature, these bits don't need to be configured.

GRF_PERIDMAC_CON3
Address: Operational Base + offset (0x02ec)
PERI DMAC control register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	wirte_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0xffff	peridmac_boot_periph_ns PERI DMAC boot_peri_ns input control Controls the security state of a peripheral request interface, when the DMAC exits from reset. Note: PERI DMAC don't support secure feature, these bits don't need to be configured.

GRF_DDRC0_CON0
Address: Operational Base + offset (0x02f0)
DDRC0 control register 0

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

Bit	Attr	Reset Value	Description
12:11	RW	0x0	ddr0_dto_lb DDR0 DTO I/O internal loopback enable
10:9	RW	0x0	ddr0_dto_te DDR0 DTO I/O on-die termination enable
8:7	RW	0x0	ddr0_dto_pdr DDR0 DTO I/O receiver power down
6:5	RW	0x0	ddr0_dto_pdd DDR0 DTO I/O driver power down
4:3	RW	0x0	ddr0_dto_iom DDR0 DTO I/O mode select
2:1	RW	0x0	ddr0_dto_oe DDR0 DTO I/O output enable
0	RW	0x0	ddr0_ato_ae Enables, if set, the analog test output I/O. Connects to the AE pin of the analog test output I/O

GRF_DDRC1_CON0
Address: Operational Base + offset (0x02f4)
DDRC1 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:11	RW	0x0	ddr1_dto_lb DDR1 DTO I/O internal loopback enable
10:9	RW	0x0	ddr1_dto_te DDR1 DTO I/O on-die termination enable
8:7	RW	0x0	ddr1_dto_pdr DDR1 DTO I/O receiver power down
6:5	RW	0x0	ddr1_dto_pdd DDR1 DTO I/O driver power down

Bit	Attr	Reset Value	Description
4:3	RW	0x0	ddr1_dto_iom DDR1 DTO I/O mode select
2:1	RW	0x0	ddr1_dto_oe DDR1 DTO I/O output enable
0	RW	0x0	ddr1_ato_ae Enables, if set, the analog test output I/O. Connects to the AE pin of the analog test output I/O

GRF_CPU_CON0
Address: Operational Base + offset (0x02f8)
CPU control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	cfgaddrfilt_en_grf A17 cfgaddrfilt_en bit control
14	RO	0x0	reserved
13:10	RW	0x0	cfgend_a17 A17 cfgend bit control Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.
9	RW	0x1	tpiu_ctl_grf tpiu_ctl bit control
8:5	RW	0x1	cs_instid_grf Coresight cs_instid bit control
4	RW	0x0	l2rstdisable_grf A17 l2rstdisable bit control
3:0	RW	0x0	l1rstdisable_grf A17 l1rstdisable bit control Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.

GRF_CPU_CON1
Address: Operational Base + offset (0x02fc)
CPU control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0ff0	cfgaddrfilt_start_grf A17 non secure filter start address[15:0]

GRF_CPU_CON2
Address: Operational Base + offset (0x0300)
CPU control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0fff	cfgaddrfilt_end_grf A17 non secure filter end address[15:0]

GRF_CPU_CON3

Address: Operational Base + offset (0x0304)
CPU control register 3

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:12	RO	0x0	reserved
11:8	RW	0x0	cfgnmfi_a17 A17 cfgnmfi bit control Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.
7:4	RW	0x0	cfgaddrfilt_end_grf A17 non secure filter end address[19:16]
3:0	RW	0x0	cfgaddrfilt_start_grf A17 non secure filter start address[19:16]

GRF_CPU_CON4
Address: Operational Base + offset (0x0308)
CPU control register 4

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RW	0x1	l2_mem_ema_grf L2 memory EMA control
12:10	RW	0x1	owl_mem_ema_grf A17 memory EMA control
9	RW	0x0	evento_clear A17 evento clear bit control
8	RW	0x0	eventi_a17 A17 eventi bit control
7:4	RO	0x0	reserved
3:0	RW	0x0	teinit_a17 A17 teinit bit control Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.

GRF_CPU_STATUS0
Address: Operational Base + offset (0x0318)
CPU status register 0

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	evento_rising_edge evento signal rising edge
13:10	RW	0x0	owl_pmupl1_grf A17 PMU Privilege level 1 event Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.
9:6	RW	0x0	owl_pmupl2_grf A17 PMU Privilege level 2 event Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.

Bit	Attr	Reset Value	Description
5:2	RW	0x0	owl_pmusecure_grf A17 pmu secure event Every bit for one core, bit3 is for core3, bit2 is for core2, bit1 is for core1, bit0 is for core0.
1	RW	0x0	jtagnsw_st_grf JTAG nsw status
0	RW	0x0	jtagtop_st_grf JTAG top status

GRF_UOC0_CON0
Address: Operational Base + offset (0x0320)
UOC0 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbotg_linestate_irq_pd USB OTG linestate interrupt pending bit
14	RW	0x0	usbotg_linestate_irq_en USB OTG line state interrupt enable
13	RW	0x0	usbotg_siddq USB OTG IDDQ test enable This test signal enables you to perform IDDQ testing by powering down all analog blocks. 1'b1: The analog blocks are powered down. 1'b0: The analog blocks are powered up.

Bit	Attr	Reset Value	Description
12	RW	0x0	usbotg_port_reset USB OTG per-port reset When asserted, this customer-specific signal resets the corresponding port transmit and receive logic without disabling the clocks within the PHY. 1'b1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 1'b0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.
11:10	RO	0x0	reserved
9:8	RW	0x0	usbotg_scaledown_mode USB OTG scale down mode control
7:5	RW	0x4	usbotg_tune USB OTG VBUS valid threshold adjustment This bus adjusts the voltage level for the VBUS Valid threshold. 3'b111: +9% 3'b110: +6% 3'b101: +3% 3'b100: Design default 3'b011: -3% 3'b010: -6% 3'b001: -9% 3'b000: -12%
4	RW	0x0	usbotg_disable USB OTG block disable 1'b1: the USB OTG block is power down 1'b0: the USB OTG block is power up
3:1	RW	0x4	usbotg_compdistune Disconnect Threshold Adjustment This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 3'b111: +4.5% 3'b110: +3% 3'b101: +1.5% 3'b100: Design default 3'b011: -1.5% 3'b010: -3% 3'b001: -4.5% 3'b000: -6%

Bit	Attr	Reset Value	Description
0	RW	0x1	usbotg_common_on_n USB OTG common block power-down control This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode. 1'b1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 1'b0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered.

GRF_UOC0_CON1
Address: Operational Base + offset (0x0324)
UOC0 control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	usbotg_txrisetune USB OTG HS transmitter rise/fall time adjustment This bus adjusts the rise/fall times of the high-speed waveform. 2'b11: -20% 2'b10: -15% 2'b01: design default 2'b00: +10%

Bit	Attr	Reset Value	Description
13:12	RW	0x3	usbotg_txhsxvtune USB OTG transmitter high-speed crossover adjustment This bus adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 2'b11: Default setting 2'b10: +15 mV 2'b01: -15 mV 2'b00: Reserved
11:8	RW	0x3	usbotg_txvreftune USB OTG HS DC voltage level adjustment This bus adjusts the high-speed DC level voltage. 4'b1111: +8.75% 4'b1110: +7.5% 4'b1101: +6.25% 4'b1100: +5% 4'b1011: +3.75% 4'b1010: +2.5% 4'b1001: +1.25% 4'b1000: Design default 4'b0111: -1.25% 4'b0110: -2.5% 4'b0101: -3.75% 4'b0100: -5% 4'b0011: -6.25% 4'b0010: -7.5% 4'b0001: -8.75% 4'b0000: -10%
7:4	RW	0x3	usbotg_txfslstune USB OTG FS/LS source impedance adjustment This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 4'b1111: -5% 4'b0111: -2.5% 4'b0011: Design default 4'b0001: +2.5% 4'b0000: +5%

Bit	Attr	Reset Value	Description
3	RW	0x0	usbotg_txpreemppulsetune USB OTG HS transmitter pre-emphasis duration control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. transition in HS mode. 1'b1: 1X, short pre-emphasis current duration 1'b0: (desian default) 2X, long pre-emphasis currrent duration
2:0	RW	0x3	usbotg_sqrxtune USB OTG squelch threshold adjustment This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 3'b111: -20% 3'b110: -15% 3'b101: -10% 3'b100: -5% 3'b011: Design default 3'b010: +5% 3'b001: +10% 3'b000: +15%

GRF_UOC0_CON2
Address: Operational Base + offset (0x0328)
UOC0 control register 2

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

Bit	Attr	Reset Value	Description
15	RW	0x0	usbotg_acaenb USB OTG ACA ID_OTG pin resistance detection enable 1'b1: enable detection on resistance on the ID_OTG pin of an ACA 1'b0: disable detection on resistance on the ID_OTG pin of an ACA
14	RW	0x0	usbotg_dcdenb USB OTG data contact detection enable 1'b1: IDP_SRC current is sourced onto DP, pull-down resistance on DMA is enabled 1'b0: IDP_SRC current is disable, pull-down resistance on DM is disabled
13	RO	0x0	reserved
12:11	RW	0x1	usbotg_txrestune USB OTG source impedance adjustment 2'b11: source impedance is desreased by 4ohm 2'b10: source impedance is desreased by 2ohm 2'b01: design default 2'b00: source impedance is desreased by 1.5ohm
10	RW	0x1	usbotg_sleepm USB OTG sleep mode enable Asserting this signal place the USB PHY in sleep mode. 1'b0: sleep mode enable 1'b1: normal mode
9	RO	0x0	reserved
8	RW	0x1	usbotg_retenable_n USB OTG retention mode enable 0: retention mode enable 1: retention mode disable
7	RW	0x0	usbotg_vdatsrcenb USB OTG battery charging sourcing select 1'b1: data source voltage is enable 1'b0: data source voltage is disable
6	RW	0x0	usbotg_vdatdetenb USB OTG battery charging attach/connect detection enable 1'b1: enable 1'b0: disable

Bit	Attr	Reset Value	Description
5	RW	0x0	usbotg_chrgsel USB OTG battery charging source select 1'b1: data source voltage is sourced onto DM and sunk from DP 1'b0: data source voltage is sourced onto DP and sunk from DM
4:3	RW	0x1	usbotg_txpreempamptune 2'b11: 3X pre-emphasis current 2'b10: 2X pre-emphasis current 2'b01: 1X pre-emphasis current 2'b00: HS Transmitter Pre-Emphasis is disabled
2	RW	0x0	usbotg_soft_con_sel 1'b0: software control usb otg disable 1'b1: software control usb otg enable
1	RW	0x0	usbotg_vbusvldextsel USB OTG external VBUS valid select This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. 1'b1: The VBUSVLDEXT input is used. 1'b0: The internal Session Valid comparator is used.
0	RW	0x0	usbotg_vbusvldext USB OTG external VBUS valid indicator This signal is valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, BUSVLDEXT enables the pullup resistor on the D+ line. 1'b1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 1'b0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.

GRF_UOC0_CON3
Address: Operational Base + offset (0x032c)
UOC0 control register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbotg_dbnce_fltr_bypass USB OTG debounce filter bypass enable
14	RO	0x0	reserved
13	RW	0x0	usbotg_iddiq_sel USB OTG iddig soft control enable 1'b1: software control 1'b0: hardware control
12	RW	0x0	usbotg_iddiq USB OTG iddig software control bit
11:8	RO	0x0	reserved
7	RW	0x0	usbotg_bypasssel transmitter digital bypass select 1'b1: transmitter digital bypass mode is enabled 1'b0: transmitte digital bypass mode is disabled
6	RW	0x0	usbotg_bypassdmen DM0 transmitter digital bypass enable 1'b1: DM0 FS/LS driver is enabled and driven with the BYPASSDPDATA0 signals 1'b0: DM0 FS/LS driver is disabled in transmitter digital byapss mode
5	RW	0x0	usbotg_utmi_termselect USB OTG utmi termination select 1'b1: full speed terminations are enabled 1'b0: high speed terminations are enabled

Bit	Attr	Reset Value	Description
4:3	RW	0x0	usbotg_utmi_xcvrselect USB OTG utmi transceiver select 2'b11: sends an LS packet on an FS bus or receives an LS packet 2'b10: LS transceiver 2'b01: FS transceiver 2'b00: HS transceiver
2:1	RW	0x0	usbotg_utmi_opmode USB OTG utmi operation mode This controller bus selects the UTMI+ operation mode 2'b11: normal operation without SYNC or EOP generation 2'b10: disable bit stuffing and NRZI encoding 2'b01: no-driving 2'b00: normal
0	RW	0x1	usbotg_utmi_suspend_n USB OTG suspend mode enable 1'b1: normal operation mode 1'b0: suspend mode

GRF_UOC0_CON4
Address: Operational Base + offset (0x0330)
UOC0 control register 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	usbotg_id_fall_edge_irq_pd USB OTG id fall edge interrupt pending bit, write 1 to this bit , it will be cleared.

Bit	Attr	Reset Value	Description
6	RW	0x0	usbotg_id_fall_edge_irq_en USB OTG id fall edge interrupt enable
5	RW	0x0	usbotg_id_rise_edge_irq_pd USB OTG id rise edge interrupt pending bit, write 1 to this bit , it will be cleared.
4	RW	0x0	usbotg_id_rise_edge_irq_en USB OTG id rise edge interrupt enable
3	RW	0x0	usbotg_bvalid_irq_pd USB OTG bvalid interrupt pending bit, write 1 to this bit, it will be cleared.
2	RW	0x0	usbotg_bvalid_irq_en USB OTG bvalid interrupt enable
1:0	RW	0x3	linestate_cnt_sel linestate signal filter time select 2'b00: 100us 2'b01: 500us 2'b10: 2.5ms 2'b11: 15ms

GRF_UOC1_CON0
Address: Operational Base + offset (0x0334)
UOC1 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost0_linestate_irq_pd USB HOST0 linestate interrupt pending bit
14	RW	0x0	usbhost0_linestate_irq_en USB HOST0 line state interrupt enable

Bit	Attr	Reset Value	Description
13	RW	0x0	usbhost0_siddq USB HOST0 IDDQ test enable This test signal enables you to perform IDDQ testing by powering down all analog blocks. 1'b1: The analog blocks are powered down. 1'b0: The analog blocks are powered up.
12	RW	0x0	usbhost0_port_reset USB HOST0 per-port reset When asserted, this customer-specific signal resets the corresponding port transmit and receive logic without disabling the clocks within the PHY. 1'b1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 1'b0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.
11	RW	0x1	usbhost0_word_if USB HOST0 word_if bit control
10	RW	0x0	usbhost0_sim_mode USB HOST0 sim_mode bit control
9	RW	0x1	usbhost0_incrx_en USB HOST0 incrx_en bit control
8	RW	0x1	usbhost0_incr8_en USB HOST0 incr8_en bit control
7:5	RW	0x4	usbhost0_tune USB HOST0 VBUS valid threshold adjustment This bus adjusts the voltage level for the VBUS Valid threshold. 3'b111: +9% 3'b110: +6% 3'b101: +3% 3'b100: Design default 3'b011: -3% 3'b010: -6% 3'b001: -9% 3'b000: -12%
4	RW	0x0	usbhost0_disable USB HOST0 block disable 1'b1: the USB HOST0 block is power down 1'b0: the USB HOST0 block is power up

Bit	Attr	Reset Value	Description
3:1	RW	0x4	usbhost0_compdistune USB HOST0 disconnect threshold adjustment This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 3'b111: +4.5% 3'b110: +3% 3'b101: +1.5% 3'b100: Design default 3'b011: -1.5% 3'b010: -3% 3'b001: -4.5% 3'b000: -6%
0	RW	0x1	usbhost0_common_on_n USB HOST0 common block power-down control This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode. 1'b1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 1'b0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered.

GRF_UOC1_CON1
Address: Operational Base + offset (0x0338)
UOC1 control register 1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	usbhost0_txrisetune USB HOST0 HS transmitter rise/fall time adjustment This bus adjusts the rise/fall times of the high-speed waveform. 2'b11: -20% 2'b10: -15% 2'b01: design default 2'b00: +10%
13:12	RW	0x3	usbhost0_txhsxvtune USB HOST0 transmitter high-speed crossover adjustment This bus adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 2'b11: Default setting 2'b10: +15 mV 2'b01: -15 mV 2'b00: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x3	usbhost0_txvreftune USB HOST0 HS DC voltage level adjustment This bus adjusts the high-speed DC level voltage. 4'b1111: +8.75% 4'b1110: +7.5% 4'b1101: +6.25% 4'b1100: +5% 4'b1011: +3.75% 4'b1010: +2.5% 4'b1001: +1.25% 4'b1000: Design default 4'b0111: -1.25% 4'b0110: -2.5% 4'b0101: -3.75% 4'b0100: -5% 4'b0011: -6.25% 4'b0010: -7.5% 4'b0001: -8.75% 4'b0000: -10%
7:4	RW	0x3	usbhost0_txfslstune USB HOST0 FS/LS source impedance adjustment This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 4'b1111: -5% 4'b0111: -2.5% 4'b0011: Design default 4'b0001: +2.5% 4'b0000: +5%
3	RW	0x0	usbhost0_txpreemppulsetune USB HOST0 HS transmitter pre-emphasis duration control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. transition in HS mode. 1'b1: 1X, short pre-emphasis current duration 1'b0: (desian default) 2X, long pre-emphasis currrent duration

Bit	Attr	Reset Value	Description
2:0	RW	0x3	usbhost0_sqrxtune USB HOST0 squelch threshold adjustment This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 3'b111: -20% 3'b110: -15% 3'b101: -10% 3'b100: -5% 3'b011: Design default 3'b010: +5% 3'b001: +10% 3'b000: +15%

GRF_UOC1_CON2

Address: Operational Base + offset (0x033c)

UOC1 control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost0_acaenb USB HOST0 ACA ID_OTG pin resistance detection enable 1'b1: enable detection on resistance on the ID_OTG pin of an ACA 1'b0: disable detection on resistance on the ID_OTG pin of an ACA

Bit	Attr	Reset Value	Description
14	RW	0x0	usbhost0_dcdenb USB HOST0 data contact detection enable 1'b1: IDP_SRC current is sourced onto DP, pull-down resistance on DMA is enabled 1'b0: IDP_SRC current is disable, pull-down resistance on DM is disabled
13	RW	0x0	usbhost0_app_prt_ovrcur USB HOST0 app_prt_ovrcur bit control
12:11	RW	0x1	usbhost0_txrestune USB HOST0 source impedance adjustment 2'b11: source impedance is desreased by 4ohm 2'b10: source impedance is desreased by 2ohm 2'b01: design default 2'b00: source impedance is desreased by 1.5ohm
10	RW	0x1	usbhost0_sleepm USB HOST0 sleep mode enable Asserting this signal place the USB PHY in sleep mode. 1'b0: sleep mode enable 1'b1: normal mode
9	RW	0x0	usbhost0_autoppd_on_overcur USB HOST0 autoppd_on_overcur bit control
8	RW	0x1	usbhost0_retenable_n USB HOST0 retention mode enable 0: retention mode enable 1: retention mode disable
7	RW	0x0	usbhost0_vdatsrcenb USB HOST0 battery charging sourcing select 1'b1: data source voltage is enable 1'b0: data source voltage is disable
6	RW	0x0	usbhost0_vdatdetenb USB HOST0 battery charging attach/connect detection enable 1'b1: enable 1'b0: disable
5	RW	0x0	usbhost0_chrgsel USB HOST0 battery charging source select 1'b1: data source voltage is sourced onto DM and sunk from DP 1'b0: data source voltage is sourced onto DP and sunk from DM

Bit	Attr	Reset Value	Description
4:3	RW	0x1	usbhost0_txpreempamptune 2'b11: 3X pre-emphasis current 2'b10: 2X pre-emphasis current 2'b01: 1X pre-emphasis current 2'b00: HS Transmitter Pre-Emphasis is disabled
2	RW	0x0	usbhost0_soft_con_sel 1'b0: software control usb host0 disable 1'b1: software control usb host0 enable
1	RW	0x0	usbhost0_vbusvldextsel USB HOST0 external VBUS valid select This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. 1'b1: The VBUSVLDEXT input is used. 1'b0: The internal Session Valid comparator is used.
0	RW	0x0	usbhost0_vbusvldext USB HOST0 external VBUS valid indicator This signal is valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, BUSVLDEXT enables the pullup resistor on the D+ line. 1'b1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 1'b0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.

GRF_UOC1_CON3

Address: Operational Base + offset (0x0340)
UOC1 control register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost0_ohci_susp_lgcy USB HOST0 ohci_susp_lgcy bit control
14	RW	0x0	usbhost0_ohci_cntsel USB HOST0 ohci_cntsel bit control
13	RW	0x0	usbhost0_utmiotg_idpullup USB HOST0 idpullup bit control
12	RW	0x1	usbhost0_utmiotg_dppulldown USB HOST0 dppulldown bit control
11	RW	0x1	usbhost0_utmiotg_dmpulldown USB HOST0 dmpulldown bit control
10	RW	0x1	usbhost0_utmiotg_drvvbus USB HOST0 drvvbus bit contrl
9:7	RO	0x0	reserved
6	RW	0x1	usbhost0_ohci_clkcktrst USB HOST0 ohci_clkcktrst bit conrol
5	RW	0x0	usbhost0_utmi_termselect USB HOST0 utmi termination select 1'b1: full speed terminations are enabled 1'b0: high speed terminations are enabled
4:3	RW	0x0	usbhost0_utmi_xcvrselect USB HOST0 utmi transceiver select 2'b11: sends an LS packet on an FS bus or receives an LS packet 2'b10: LS transceiver 2'b01: FS transceiver 2'b00: HS transceiver

Bit	Attr	Reset Value	Description
2:1	RW	0x0	usbhost0_utmi_opmode USB HOST0 utmi operation mode This controller bus selects the UTMI+ operation mode 2'b11: normal operation without SYNC or EOP generation 2'b10: disable bit stuffing and NRZI encoding 2'b01: no-driving 2'b00: normal
0	RW	0x1	usbhost0_utmi_suspend_n USB HOST0 suspend mode enable 1'b1: normal operation mode 1'b0: suspend mode

GRF_UOC1_CON4
Address: Operational Base + offset (0x0344)
UOC1 control register 4

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	usbhost0_incr4_en USB HOST0 incr4_en bit control
14	RW	0x1	usbhost0_incr16_en USB HOST0 incr16_en bit control
13	RW	0x0	usbhost0_hubsetup_min USB HOST0 hubsetup_min bit control
12	RW	0x0	usbhost0_app_start_clk USB HOST0 app_start_clk bit control
11:6	RW	0x20	usbhost0_fladj_val_common USB HOST0 fladj_val_common bit control
5:0	RW	0x20	usbhost0_fladj USB HOST0 fladj bit control

GRF_UOC2_CON0

Address: Operational Base + offset (0x0348)

UOC2 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost1_linestate_irq_pd USB HOST1 linestate interrupt pending bit
14	RW	0x0	usbhost1_linestate_irq_en USB HOST1 line state interrupt enable
13	RW	0x0	usbhost1_siddq USB HOST1 IDDQ test enable This test signal enables you to perform IDDQ testing by powering down all analog blocks. 1'b1: The analog blocks are powered down. 1'b0: The analog blocks are powered up.
12	RW	0x0	usbhost1_port_reset USB HOST1 per-port reset When asserted, this customer-specific signal resets the corresponding port transmit and receive logic without disabling the clocks within the PHY. 1'b1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 1'b0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.
11:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:5	RW	0x4	usbhost1_tune USB HOST1 VBUS valid threshold adjustment This bus adjusts the voltage level for the VBUS Valid threshold. 3'b111: +9% 3'b110: +6% 3'b101: +3% 3'b100: Design default 3'b011: -3% 3'b010: -6% 3'b001: -9% 3'b000: -12%
4	RW	0x0	usbhost1_disable USB HOST1 block disable 1'b1: the USB HOST1 block is power down 1'b0: the USB HOST1 block is power up
3:1	RW	0x4	usbhost1_compdistune USB HOST1 disconnect threshold adjustment This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 3'b111: +4.5% 3'b110: +3% 3'b101: +1.5% 3'b100: Design default 3'b011: -1.5% 3'b010: -3% 3'b001: -4.5% 3'b000: -6%
0	RW	0x1	usbhost1_common_on_n USB HOST1 common block power-down control This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode. 1'b1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 1'b0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered.

GRF_UOC2_CON1
Address: Operational Base + offset (0x034c)

UOC2 control register 1

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x1	usbhost1_txrisetune USB HOST1 HS transmitter rise/fall time adjustment This bus adjusts the rise/fall times of the high-speed waveform. 2'b11: -20% 2'b10: -15% 2'b01: design default 2'b00: +10%
13:12	RW	0x3	usbhost1_txhsxvtune USB HOST1 transmitter high-speed crossover adjustment This bus adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 2'b11: Default setting 2'b10: +15 mV 2'b01: -15 mV 2'b00: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x3	usbhost1_txvreftune USB HOST1 HS DC voltage level adjustment This bus adjusts the high-speed DC level voltage. 4'b1111: +8.75% 4'b1110: +7.5% 4'b1101: +6.25% 4'b1100: +5% 4'b1011: +3.75% 4'b1010: +2.5% 4'b1001: +1.25% 4'b1000: Design default 4'b0111: -1.25% 4'b0110: -2.5% 4'b0101: -3.75% 4'b0100: -5% 4'b0011: -6.25% 4'b0010: -7.5% 4'b0001: -8.75% 4'b0000: -10%
7:4	RW	0x3	usbhost1_txfslstune USB HOST1 FS/LS source impedance adjustment This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 4'b1111: -5% 4'b0111: -2.5% 4'b0011: Design default 4'b0001: +2.5% 4'b0000: +5%
3	RW	0x0	usbhost1_txpreemppulsetune USB HOST1 HS transmitter pre-emphasis duration control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP0 or DM0. transition in HS mode. 1'b1: 1X, short pre-emphasis current duration 1'b0: (desian default) 2X, long pre-emphasis currrent duration

Bit	Attr	Reset Value	Description
2:0	RW	0x3	usbhost1_sqrxtune USB HOST1 squelch threshold adjustment This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 3'b111: -20% 3'b110: -15% 3'b101: -10% 3'b100: -5% 3'b011: Design default 3'b010: +5% 3'b001: +10% 3'b000: +15%

GRF_UOC2_CON2

Address: Operational Base + offset (0x0350)

UOC2 control register 2

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	usbhost1_acaenb USB HOST1 ACA ID_OTG pin resistance detection enable 1'b1: enable detection on resistance on the ID_OTG pin of an ACA 1'b0: disable detection on resistance on the ID_OTG pin of an ACA

Bit	Attr	Reset Value	Description
14	RW	0x0	usbhost1_dcdenb USB HOST1 data contact detection enable 1'b1: IDP_SRC current is sourced onto DP, pull-down resistance on DMA is enabled 1'b0: IDP_SRC current is disable, pull-down resistance on DM is disabled
13	RO	0x0	reserved
12:11	RW	0x1	usbhost1_txrestune USB HOST1 source impedance adjustment 2'b11: source impedance is desreased by 4ohm 2'b10: source impedance is desreased by 2ohm 2'b01: design default 2'b00: source impedance is desreased by 1.5ohm
10	RW	0x1	usbhost1_sleepm USB HOST1 sleep mode enable Asserting this signal place the USB PHY in sleep mode. 1'b0: sleep mode enable 1'b1: normal mode
9	RO	0x0	reserved
8	RW	0x1	usbhost1_retenable_n USB HOST1 retention mode enable 0: retention mode enable 1: retention mode disable
7	RW	0x0	usbhost1_vdatsrcenb USB HOST1 battery charging sourcing select 1'b1: data source voltage is enable 1'b0: data source voltage is disable
6	RW	0x0	usbhost1_vdatdetenb USB HOST1 battery charging attach/connect detection enable 1'b1: enable 1'b0: disable
5	RW	0x0	usbhost1_chrgsel USB HOST1 battery charging source select 1'b1: data source voltage is sourced onto DM and sunk from DP 1'b0: data source voltage is sourced onto DP and sunk from DM

Bit	Attr	Reset Value	Description
4:3	RW	0x1	usbhost1_txpreempamptune 2'b11: 3X pre-emphasis current 2'b10: 2X pre-emphasis current 2'b01: 1X pre-emphasis current 2'b00: HS Transmitter Pre-Emphasis is disabled
2	RW	0x0	usbhost1_soft_con_sel 1'b0: software control usb host1 disable 1'b1: software control usb host1 enable
1	RW	0x0	usbhost1_vbusvldextsel USB HOST1 external VBUS valid select This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. 1'b1: The VBUSVLDEXT input is used. 1'b0: The internal Session Valid comparator is used.
0	RW	0x0	usbhost1_vbusvldext USB HOST1 external VBUS valid indicator This signal is valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, BUSVLDEXT enables the pullup resistor on the D+ line. 1'b1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 1'b0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.

GRF_UOC2_CON3
Address: Operational Base + offset (0x0354)
UOC2 control register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	usbhost1_scaledown_mode USB HOST1 scale down mode control
13	RW	0x0	usbhost1_utmiotg_idpullup USB HOST1 idpullup bit control
12	RW	0x1	usbhost1_utmiotg_dppulldown USB HOST1 dppulldown bit control
11	RW	0x1	usbhost1_utmiotg_dmpulldown USB HOST1 dmpulldown bit control
10	RW	0x1	usbhost1_utmiotg_drvvbus USB HOST1 drvvbus bit contrl
9:6	RO	0x0	reserved
5	RW	0x0	usbhost1_utmi_termselect USB HOST1 utmi termination select 1'b1: full speed terminations are enabled 1'b0: high speed terminations are enabled
4:3	RW	0x0	usbhost1_utmi_xcvrselect USB HOST1 utmi transceiver select 2'b11: sends an LS packet on an FS bus or receives an LS packet 2'b10: LS transceiver 2'b01: FS transceiver 2'b00: HS transceiver

Bit	Attr	Reset Value	Description
2:1	RW	0x0	usbhost1_utmi_opmode USB HOST1 utmi operation mode This controller bus selects the UTMI+ operation mode 2'b11: normal operation without SYNC or EOP generation 2'b10: disable bit stuffing and NRZI encoding 2'b01: no-driving 2'b00: normal
0	RW	0x1	usbhost1_utmi_suspend_n USB HOST1 suspend mode enable 1'b1: normal operation mode 1'b0: suspend mode

GRF_UOC3_CON0
Address: Operational Base + offset (0x0358)
UOC3 control register 0

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

GRF_UOC3_CON1
Address: Operational Base + offset (0x035c)
UOC3 control register 1

Bit	Attr	Reset Value	Description

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

GRF_UOC4_CON0
Address: Operational Base + offset (0x0360)
UOC4 control register 0

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	drvvbus_out_sel0 USB PHY drv vbus output select 0 2'b00: USB OTG drv vbus 2'b01: USB HOST0 drv vbus 2'b1x: USB HOST1 drv vbus

Bit	Attr	Reset Value	Description
13:12	RW	0x1	drvvbus_out_sel1 USB PHY drv vbus output select 1 2'b00: USB OTG drv vbus 2'b01: USB HOST0 drv vbus 2'b1x: USB HOST1 drv vbus
11: 0	RO	0x0	reserved

GRF_UOC4_CON1
Address: Operational Base + offset (0x0364)
UOC4 control register 1

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

GRF_PVTM_CON0
Address: Operational Base + offset (0x0368)
PVT monitor control register 0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9	RW	0x0	pvtm_gpu_osc_en pd_gpu PVT monitor oscilator enable 1'b1: enable 1'b0: disable
8	RW	0x0	pvtm_gpu_start pd_gpu PVT monitor start control
7:2	RO	0x0	reserved
1	RW	0x0	pvtm_core_osc_en pd_core PVT monitor oscilator enable 1'b1: enable 1'b0: disable
0	RW	0x0	pvtm_core_start pd_core PVT monitor start control

GRF_PVTM_CON1
Address: Operational Base + offset (0x036c)
PVT monitor control register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_core_cal_cnt pd_core pvtm calculator counter

GRF_PVTM_CON2
Address: Operational Base + offset (0x0370)
PVT monitor control register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x016e3600	pvtm_gpu_cal_cnt pd_gpu pvtm calculator counter

GRF_PVTM_STATUS0
Address: Operational Base + offset (0x0374)
PVT monitor status register 0

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	pvtm_core_freq_done pd_core pvtm frequency calculate done stutus
0	RW	0x0	pvtm_gpu_freq_done pd_gpu pvtm frequency calculate done stutus

GRF_PVTM_STATUS1
Address: Operational Base + offset (0x0378)
PVT monitor status register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_core_freq_cnt pd_core pvtm frequency count

GRF_PVTM_STATUS2
Address: Operational Base + offset (0x037c)
PVT monitor status register 2

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	pvtm_gpu_freq_cnt pd_gpu pvtm frequency count

GRF_IO_VSEL
Address: Operational Base + offset (0x0380)
IO voltage select

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

Bit	Attr	Reset Value	Description
9	RW	0x0	gpio1830_v18sel GPIO1830 IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
8	RW	0x0	gpio30_v18sel GPIO30 IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
7	RW	0x0	sdcard_v18sel SDCARD IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
6	RW	0x0	audio_v18sel AUDIO IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
5	RW	0x0	bb_v18sel BB IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
4	RW	0x0	wifi_v18sel WIFI IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
3	RW	0x0	flash1_v18sel FLASH1 IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
2	RW	0x1	flash0_v18sel FLASH0 IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
1	RW	0x0	dvp_v18sel DVP IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V
0	RW	0x0	lcdc_v18sel LCDC IO domain 1.8V voltage selection 1'b0: 3.3V 1'b1: 1.8V

GRF_SARADC_TESTBIT
Address: Operational Base + offset (0x0384)
SARADC Test bit register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7:0	RW	0x00	saradc_testbit SARADC test bit

GRF_TSADC_TESTBIT_L
Address: Operational Base + offset (0x0388)
TSADC Test bit low register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	tsadc_testbit_l Low 16bits of TSADC test bit

GRF_TSADC_TESTBIT_H
Address: Operational Base + offset (0x038c)
TSADC Test bit high register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	tsadc_testbit_h High 16bits of TSADC test bit

GRF_OS_REG0
Address: Operational Base + offset (0x0390)
OS register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg0 OS register 0

GRF_OS_REG1
Address: Operational Base + offset (0x0394)
OS register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg1 OS register 1

GRF_OS_REG2
Address: Operational Base + offset (0x0398)
OS register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg2 OS register 2

GRF_OS_REG3
Address: Operational Base + offset (0x039c)
OS register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_reg3 OS register 3

GRF_SOC_CON15
Address: Operational Base + offset (0x03a4)
SoC control register 15

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	grf_dclk1_lvds_inv_sel Invertion of VOP_LIT dclk for LVDS selection 1'b1: invert 1'b0: not invert
14	RW	0x0	grf_dclk1_lvds_div2_sel 2 divide frequency of VOP_LIT dclk for LVDS selection 1'b1: 2 divide frequency 1'b0: no divide frequency
13	RW	0x0	grf_dclk0_lvds_inv_sel Invertion of VOP_BIG dclk for LVDS selection 1'b1: invert 1'b0: not invert
12	RW	0x0	grf_dclk0_lvds_div2_sel 2 divide frequency of VOP_BIG dclk for LVDS selection 1'b1: 2 divide frequency 1'b0: no divide frequency
11	RO	0x0	reserved
10:8	RW	0x0	dphy_tx0_turnrequest MIPI DPHY TX0 turn around request Every bit for one lane, bit2 is for lane3, bit2 is for lane2, bit0 is for lane1.

Bit	Attr	Reset Value	Description
7:4	RW	0x0	dphy_tx1rx1_turnrequest MIPI DPHY TX1RX1 turn around request Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.
3:0	RW	0x0	dphy_rx0_turnrequest MIPI DPHY RX0 turn around request Every bit for one lane, bit3 is for lane3, bit2 is for lane2, bit1 is for lane1, bit0 is for lane0.

GRF_SOC_CON16
Address: Operational Base + offset (0x03a8)
SoC control register 16

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ...... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RO	0x0	reserved
1	RW	0x0	grf_con_dsi1_dpiupdatecfg DSI host1 dpiupdatecfg bit control
0	RW	0x0	grf_con_dsi0_dpiupdatecfg DSI host0 dpiupdatecfg bit control

Chapter 4 Cortex-A17

4.1 Overview

The Core system of the device is based on the symmetric multiprocessor (SMP) architecture, contain quad Cortex-A17.The Cortex-A17 implements 32KB L1 instruction cache, 32KB L1 data cache, 1MB L2 cache.It delivers higher performance and optimal power management, debug and emulation capabilities.

The core system also contain a bus interconnect, which connect the A17's peripheral port and the generic interrupt controller (GIC400).

The core system contain the ARM's coresight cell , which helps to do debug and trace .

The interrupt controller is also included in this system, for detail description , please reference to chapter 12.

The Core system supports following features:

- ARM Coretex-A17 based quad MPU subsystem with SMP architecture
- Cortex-A17 core revision r0p1
- Full implements the ARMv7-A architecture profile that includes SIMDv2 and VFPv4-D32 extensions
- 32KB L1 I-cache and 32KB L1 D-cache with 64-byte line size and 4-way set associative per CPU
- A 32-entry,fully associative, instruction micro TLB.
- A 32-entry,fully associative, data micro TLB.
- A 1024-entry, 4-way, unified main TLB with hit-under-miss capabilities.
- Memory management unit (MMU)
- Automatic cache coherency between L1 data caches within the cluster.
- Interrupt controller with 128 hardware interrupt inputs.
- 1MB L2 cache with 64-byte line size, and 16-way set associative.
- standard CoreSight™ components to support SMP debug and emulation
- Program trace macrocell (PTM)
- Emulation logic (cross-triggers)
- TPIU for trace.

4.2 Block Diagram

The Core system comprises with:

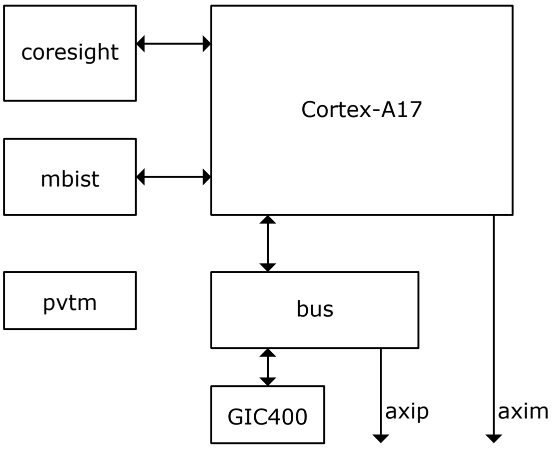


Fig. 4-1 Block Diagram

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 96KB 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

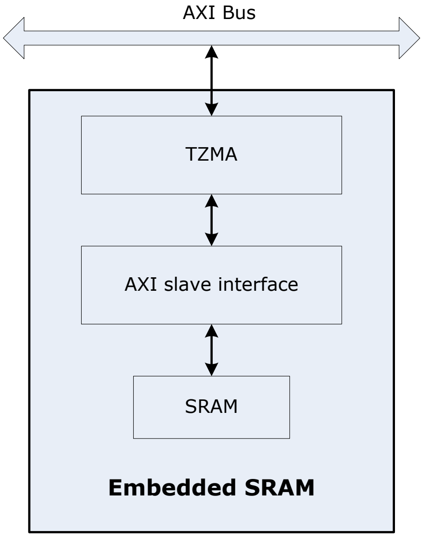


Fig. 5-1 Embedded SRAM block diagram

5.3 Function Description

5.3.1 TZMA

Please refer to 16.3.3 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-A17 and DMAC_BUS.

5.3.4 Remap

The Embedded SRAM support remap.

Before remap, the Embedded SRAM address range is 0xff70_0000~0xff71_7fff,
After set remap, (ref Security GRF register SGRF_SCON0, bit[7]), 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 0xfffe_0000 ~ 0xffff_7fff (include the bootaddr)

Chapter 6 Power Management Unit (PMU)

6.1 Overview

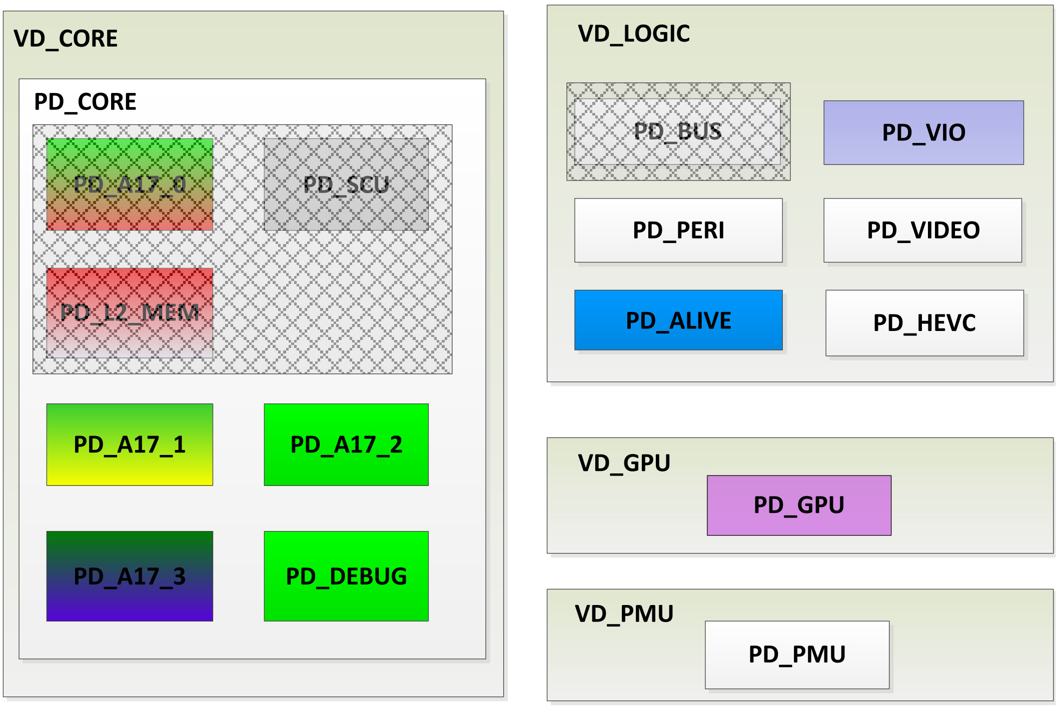
In order to meet high performance and low power requirements, a power management unit is designed for saving power when RK3288 in low power mode. The RK3288 PMU is dedicated for managing the power of the whole chip.

6.1.1 Features

- Support 4 voltage domains including VD_CORE, VD_LOGIC, VD_GPU and VD_PMU
- Support 15 separate power domains in the whole chip, which can be power up/down by software based on different application scenes
- In low power mode, the pmu could power up/down pd_A17_0, pd_scu, vd_core, and pd_bus by hardware
- Support CORTEX-A17 core source clock gating in low power mode
- Support global interrupt disable in low power mode
- Support PLLs power down/up in low power mode
- Support VD_CORE/VD_LOGIC power down/up in low power mode
- Support pd_alive clock switch to 32KHz in low power mode
- Support pd_pmu clock switch to 32KHz request in low power mode
- Support OSC enable/disable request in low power mode
- Support to clamp all VD_LOGIC output before power off it in low power mode
- Support wakeup reset control in power off mode
- Support DDR self-refresh in low power mode
- Support DDR IO retention in low power mode
- Support DDR IO power off in low power mode
- Support DDR controller clock auto gating in low power mode
- Support to send idle requests to all NIU in the SoC (details will be described later)
- A group of configurable counter in PMU for HW control (such as PLL, PMIC, DDRIO and so on)
- Support varies configurable wakeup source for low power mode

6.2 Block Diagram

6.2.1 power domain partition



Note:
VD_* : voltage domain
PD_* : power domain

Fig. 6-1 Power Domain Partition

The above diagram describes the power domain and voltage domain partition, and the following table lists all the power domains.

Table 6-1 RK3288 Power Domain and Voltage Domain Summary

Voltage Domain	**Power Domain**	**Description**
VD_CORE (PD_CORE system)	PD_A17_0	A17 primary core logic，L1C and neon
	PD_A17_1	A17 slave core 1 logic，L1C and neon
	PD_A17_2	A17 slave core 2 logic，L1C and neon
	PD_A17_3	A17 slave core 3 logic，L1C and neon
	PD_SCU	SCU RAM，SCU，GIC，Periphral，L2 controller
	PD_DEBUG	A17 Debug
	PD_MEM	L2 Cache
VD_LOGIC	PD_BUS	Soc architecture subsystem, include soc architecture (NOC) ，eFuse，TZPC，ROM，DMAC_BUS，Crypto，Host，Timer(6ch), PWM(0~3), UART_DBG, I2C, DDR_PCTL, I2S, Spdif, Internal Memory(96K)
	PD_PERI	Peripheral subsystem , include DMAC_PERI, GMAC, NANDC0/1, USB Host0/USB Host1/ USB OTG, SDMMC/SDIO0/SDIO1/eMMC，HSADC, PS2C, TSADC, UART, I2C, SPI, GPS, TSP
	PD_VIO	Video input/output system, include VOPBIG, VOPLIT, ISP, IEP, RGA, MIPI-CSI, MIPI-DSI, LVDS, HDMI, eDP

Voltage Domain	Power Domain	Description
	PD_ALIVE	CRU，GRF， GPIO 1~8， TIMER, WDT
	PD_HEVC	HEVC
	PD_VIDEO	Video Encode&Decode , include VEPU，VDPU
VD_GPU	PD_GPU	GPU
VD_PMU	PD_PMU	PMU, SRAM(4K), Secure GRF, GPIO0

Notes: "Always on" means that their power supply can be switched off only by external PMIC module. Only one "always on" power domain is in a voltage domain.

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

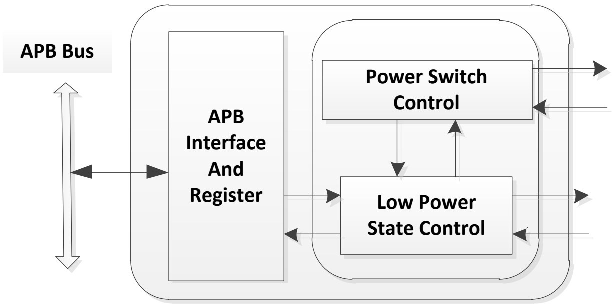


Fig. 6-2 PMU Bock Diagram

6.3 Power Switch Timing Requirement

The following table describe the switch time for power down and power up progress of each power domain. This table gives the time range, and each power domain switch time will be more than the min time and less than the max time.

Table 6-2 Power Switch Timing

Power domain	type	Power down Switch Timing① (ns)	Power up Switch Timing① (ns)
PD_A17_0	min	170.3	132.4
	max	306.7	237.5
PD_A17_1	min	170.3	132.4
	max	306.7	237.5
PD_A17_2	min	181.5	140.7
	max	326.2	251.9
PD_A17_3	min	181.5	140.7
	max	326.2	251.9
PD_DEBUG	min	169.4	131.7
	max	77.3	60
PD_BUS	min	169.4	131.7
	max	313.6	247
PD_PERI	min	103.7	80.5
	max	199.0	156.1
PD_VIO	min	280.6	217.5
	max	518.5	407.8

Power domain	type	Power down Switch Timing① (ns)	Power up Switch Timing① (ns)
PD_VIDEO	min	315.4	244.2
	max	586.2	460.4
PD_GPU	min	470.2	364
	max	871.4	684.1
PD_HEVC	min	33.4	25.9
	max	62.4	49
	max	65.6	51.5

Notes: the power switch timing is just the chip power electrical parameter, this is not the parameter for the software to determine the power domain status. The software need to check each power domain status register to determine the power status.

6.4 Function Description

6.4.1 Normal Mode

First of all, we define two modes of power for chip, normal mode and low power mode.
In normal mode, the PMU can power off/on all power domain (except pd_A17_0 and pd_bus) by setting PMU_PWRDN_CON register. At same, pmu can send idle request for every power domain by setting PMU_IDLE_REQ register.
Don’t set pd_A17_0 and pd_bus power off or send idle_req_core and idle_req_bus in normal mode. This will cause the system to not work properly.
Basically, there are 2 configurations that software can do in normal mode to save power.

- Configure DDR to self-refresh, DDR IO retention and DDR IO power off
- Power down power domains

The first one will save power consumption of using DDR controller and DDR IO. For avoiding confliction, the software must make sure the execution code of this step is not in DDR.
The second one will save power of the power domain which software is shutting down.

6.4.2 Low Power Mode

PMU can work in the Low Power Mode by setting bit[0] of PMU_PWRMODE_CON register. After setting the register, PMU would enter the Low Power mode. In the low power mode, pmu will auto power on/off the specified power domain, send idle req to specified power domain, shut down/up pll and so on. All of above are configurable by setting corresponding registers.

Table 6-3 Low Power State

Num	Hardware Flow	Description of Flow	Corresponding Register
0	NORMAL	in normal	
1	L2FLUSH_REQ	send L2 cache flush request	bit[3] of PMU_PWRMODE_CON
2	STANDBYL2	wait L2 cache standy	
3	A17_CLK_DIS	close A17 clock	bit[1] of PMU_PWRMODE_CON
4	TRANS_NO_FIN	wait the corresponding noc interface end the transaction	PMU_PMRMODE_CON1
5	SREF_ENTER	enter DDR self-refresh	bit[16:15] of PMU_PWRMODE_CON
6	DDR_IO_RET	ddr io retention	bit[18:17] of PMU_PWRMODE_CON
7	DDR_IO_PWROFF	ddr io power off	bit[20:19] of PMU_PWRMODE_CON
8	BUS_PWRDN	pd_bus power down	bit[4] of PMU_PWRMODE_CON
9	A17_0_PWRDN	pd_a17_0 power down	bit[5] of PMU_PWRMODE_CON
10	L2MEM_PWRDN	pd_l2mem power down (vd_core power down)	bit[6] of PMU_PWRMODE_CON
11	ALIVE_PMU_LF	pd_alive& pd_pmu switch to 32KHz clock	bit[11:10] of PMU_PWRMODE_CON
12	PLL_PWRDN	pll power down	bit[7] of PMU_PWRMODE_CON
13	INPUT_CLAMP	isolation cell input clamp	bit[13] of PMU_PWRMODE_CON
14	POWEROFF	chip power off	bit[8] of PMU_PWRMODE_CON
15	24M_OSC_DIS	close 24MHz OSC	bit[12] of PMU_PWRMODE_CON
16	WAIT_WAKEUP	wait wakeup source	PMU_WAKEUP_CFG0/1
17	WAKEUP_RESET	send reset after wakeup	
18	EXT_PWRUP	pmic power up whole chip	
19	RELEASE_CLAMP	release isolation cell clamp	
20	24M_OSC_EN	open 24MHz OSC	
21	ALIVE_PMU_HF	switch pd_alive and pd_pmu back to 24MHz	
22	WAKEUP_RESET_CLR	release wakeup reset	
23	PLL_PWRUP	Pll power up	
24	BUS_PWRUP	PD_BUS power up	
25	DDR_IO_PWRUP	ddr io power up	
26	SREF_EXIT	exit ddr self-refresh	
27	L2MEM_PWRUP	pd_l2mem power up	
28	A17_0_PWRUP	pd_a17_0 power up	
29	TRANS_RESTORE	restore the transaction	
30	A17_CLK_EN	enable a17 clock	

The Low Power mode have three steps:

Enter Low Power mode, there are some sub-steps in the enter step, every sub-step can be enable/disable by setting the corresponding register.

Wait wakeup, you can select the wakeup source by setting PMU_WAKEUP_CFG0/1 register

Exit Low Power mode, the sub-step are executed depend on whether they were executed in enter low power step.

6.4.3 Wakeup source of AP

The wakeup source is a group of signals which can trigger PMU from power mode to normal mode, such as SDMMC detect, core interrupt, GPIO0, and gpio interrupt.

Table 6-4 Wakeup Source

Num	Wakeup Source	Description
1	software control	software control (in normal mode)
2	arm interrupt	a17 interrupt
3	pmu_gpio	gpio0, in pd_pmu
4	sdmmc0	detect_n of sdmmc0
5	gpio int	gpio interrupt outside of pd_pmu

If software expect PMU be woken up from power mode by a wake up source, it should be enabled by write 1 to the corresponding bit of PMU_WAKEUP_CFG0/1 register before entering into power mode.

Technically, wakeup source can be used in every power mode if only the path from wakeup source to PMU is not shut down.

6.5 Register Description

6.5.1 Register Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG0	0x0000	W	0x00000000	PMU wake-up source configuration register0
PMU_WAKEUP_CFG1	0x0004	W	0x00000000	PMU wake-up source configuration register1
PMU_PWRDN_CON	0x0008	W	0x00000000	System power gating configuration register
PMU_PWRDN_ST	0x000c	W	0x00000000	System power gating status register
PMU_IDLE_REQ	0x0010	W	0x00000000	PMU Noc idle req control
PMU_IDLE_ST	0x0014	W	0x00000000	PMU Noc idle status
PMU_PWRMODE_CON	0x0018	W	0x00000000	PMU configuration register in power mode flow
PMU_PWR_STATE	0x001c	W	0x00000000	PMU Low power mode state
PMU_OSC_CNT	0x0020	W	0x00005dc0	24MHz OSC stabilization counter threshold
PMU_PLL_CNT	0x0024	W	0x10004000	PLL lock counter threshold
PMU_STABL_CNT	0x0028	W	0x00005dc0	External PMU stabilization counter threshold
PMU_DDR0IO_PWRON_CNT	0x002c	W	0x00005dc0	DDR0 IO power on counter threshold
PMU_DDR1IO_PWRON_CNT	0x0030	W	0x00005dc0	DDR1 IO power on counter threshold
PMU_CORE_PWRDWN_CNT	0x0034	W	0x00005dc0	CORE domain power down waiting counter in sleep mode
PMU_CORE_PWRUP_CNT	0x0038	W	0x00005dc0	CORE domain power up waiting counter in sleep mode
PMU_GPU_PWRDWN_CNT	0x003c	W	0x00005dc0	GPU domain power down waiting counter in sleep mode
PMU_GPU_PWRUP_CNT	0x0040	W	0x00005dc0	GPU domain power up waiting counter in sleep mode
PMU_WAKEUP_RST_CLR_CNT	0x0044	W	0x00005dc0	Wakeup reset deassert state wait counter in power off mode
PMU_SFT_CON	0x0048	W	0x00000000	PMU Software control in normal mode
PMU_DDR_SREF_ST	0x004c	W	0x00000000	PMU DDR self refresh status
PMU_INT_CON	0x0050	W	0x00000000	PMU interrupt configuration register
PMU_INT_ST	0x0054	W	0x00000000	PMU interrupt status register
PMU_BOOT_ADDR_SEL	0x0058	W	0x00005dc0	boot_addr_sel in power mode
PMU_GRF_CON	0x005c	W	0x00000008	grf control register

Name	Offset	Size	Reset Value	Description
PMU_GPIO_SR	0x0060	W	0x00020000	GPIO slew rate control
PMU_GPIO0_A_PULL	0x0064	W	0x0000555a	GPIO0A input to PU/PD programmation section
PMU_GPIO0_B_PULL	0x0068	W	0x00005555	GPIO0B input to PU/PD programmation section
PMU_GPIO0_C_PULL	0x006c	W	0x00000015	GPIO0C input to PU/PD programmation section
PMU_GPIO0_A_DRV	0x0070	W	0x0000555a	GPIO0A Drive strength slector
PMU_GPIO0_B_DRV	0x0074	W	0x00005555	GPIO0B Drive strength slector
PMU_GPIO0_C_DRV	0x0078	W	0x00000015	GPIO0C Drive strength slector
PMU_GPIO_OP	0x007c	W	0x00000000	GPIO0 output value
PMU_GPIO0_SEL18	0x0080	W	0x00000006	gpio0 1.8v/3.3v sel
PMU_GPIO0_A_IOMUX	0x0084	W	0x00000000	GPIO0A iomux sel
PMU_GPIO0_B_IOMUX	0x0088	W	0x00000000	GPIO0B iomux sel
PMU_GPIO0_C_IOMUX	0x008c	W	0x00000000	GPIO0C iomux sel
PMU_PWRMODE_CON1	0x0090	W	0x00000000	PMU power mode controll1
PMU_SYS_REG0	0x0094	W	0x00000000	PMU system register0
PMU_SYS_REG1	0x0098	W	0x00000000	PMU system register1
PMU_SYS_REG2	0x009c	W	0x00000000	PMU system register2
PMU_SYS_REG3	0x00a0	W	0x00000000	PMU system register3

6.5.2 Detail Register Description

PMU_WAKEUP_CFG0

Address: Operational Base + offset (0x0000)

PMU wake-up source configuration register0

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RW	0x0	gpio0c_2_wakeup_en GPIO0c bit2 wakeup enable 1'b0: disable 1'b1: enable
17	RW	0x0	gpio0c_1_wakeup_en GPIO0c bit1 wakeup enable 1'b0: disable 1'b1: enable
16	RW	0x0	gpio0c_0_wakeup_en GPIO0c bit0 wakeup enable 1'b0: disable 1'b1: enable
15	RW	0x0	gpio0b_7_wakeup_en GPIO0b bit7 wakeup enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
14	RW	0x0	gpio0b_6_wakeup_en GPIO0b bit6 wakeup enable 1'b0: disable 1'b1: enable
13	RW	0x0	gpio0b_5_wakeup_en GPIO0b bit5 wakeup enable 1'b0: disable 1'b1: enable
12	RW	0x0	gpio0b_4_wakeup_en GPIO0 bit12 wakeup enable 1'b0: disable 1'b1: enable
11	RW	0x0	gpio0b_3_wakeup_en GPIO0b bit3 wakeup enable 1'b0: disable 1'b1: enable
10	RW	0x0	gpio0b_2_wakeup_en GPIO0b bit2 wakeup enable 1'b0: disable 1'b1: enable
9	RW	0x0	gpio0b_1_wakeup_en GPIO0b bit1 wakeup enable 1'b0: disable 1'b1: enable
8	RW	0x0	gpio0b_0_wakeup_en GPIO0b bit0 wakeup enable 1'b0: disable 1'b1: enable
7	RW	0x0	gpio0a_7_wakeup_en GPIO0a bit7 wakeup enable 1'b0: disable 1'b1: enable
6	RW	0x0	gpio0a_6_wakeup_en GPIO0a bit6 wakeup enable 1'b0: disable 1'b1: enable
5	RW	0x0	gpio0a_5_wakeup_en GPIO0a bit5 wakeup enable 1'b0: disable 1'b1: enable
4	RW	0x0	gpio0a_4_wakeup_en GPIO0a bit4 wakeup enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio0a_3_wakeup_en GPIO0a bit3 wakeup enable 1'b0: disable 1'b1: enable
2	RW	0x0	gpio0a_2_wakeup_en GPIO0a bit2 wakeup enable 1'b0: disable 1'b1: enable
1	RW	0x0	gpio0a_1_wakeup_en GPIO0a bit1 wakeup enable 1'b0: disable 1'b1: enable
0	RW	0x0	gpio0a_0_wakeup_en GPIO0a bit0 wakeup enable 1'b0: disable 1'b1: enable

PMU_WAKEUP_CFG1
Address: Operational Base + offset (0x0004)
PMU wake-up source configuration register1

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	gpioint_wakeup_en GPIO Interrupt wake enable 1'b0: disable 1'b1: enable
2	RW	0x0	sdmmc0_wakeup_en SDMMC0 wake-up enable 1'b0: disable 1'b1: enable
1	RW	0x0	pmu_gpio_wakeup_type GPIO in pmu wakeup type 1'b0: posedge 1'b1: negedge
0	RW	0x0	armint_wakeup_en ARM interrupt wake-up enable 1'b0: disable 1'b1: enable

PMU_PWRDN_CON
Address: Operational Base + offset (0x0008)
System power gating configuration register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
14	RW	0x0	PD_HEVC_DWN_EN Power domain HEVC power down enable 1'b0: power on 1'b1: power off
13	RW	0x0	CHIP_PWROFF_EN software conifg power off chip logic 1'b1: power off 1'b0: not power off
12	RW	0x0	CORE_PWROFF_EN software conifg power off pd_core 1'b1: power off 1'b0: not power off
11	RW	0x0	PD_SCU_DWN_EN Power domain SCU power down enable 1'b0: power on 1'b1: power off
10	RO	0x0	reserved
9	RW	0x0	PD_GPU_DWN_EN Power domain GPU power down enable 1'b0: power on 1'b1: power off
8	RW	0x0	PD_VIDEO_DWN_EN Power domain VIDEO power down enable 1'b0: power on 1'b1: power off
7	RW	0x0	PD_VIO_DWN_EN Power domain VIO power down enable 1'b0: power on 1'b1: power off
6	RW	0x0	PD_PERI_DWN_EN Power domain PERI power down enable 1'b0: power on 1'b1: power off
5	RW	0x0	PD_BUS_DWN_EN Power domain BUS power down enable 1'b0: power on 1'b1: power off
4	RO	0x0	reserved
3	RW	0x0	PD_A17_3_DWN_EN Power Domain A17 slave core 3 power down enable 1'b0: power on 1'b1: power off

Bit	Attr	Reset Value	Description
2	RW	0x0	PD_A17_2_DWN_EN Power Domain A17 slave core 2 power down enable 1'b0: power on 1'b1: power off
1	RW	0x0	PD_A17_1_DWN_EN Power domain A17 slave core 1 power down enable 1'b0: power on 1'b1: power off
0	RW	0x0	PD_A17_0_DWN_EN Power Domain A17 primary core power down enable 1'b0: power on 1'b1: power off

PMU_PWRDN_ST
Address: Operational Base + offset (0x000c)
System power gating status register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	l2_standywfi
12	RW	0x0	l2_flush_done
11	RO	0x0	pd_scu_pwr_st Power domain SCU power status 1'b0: power on 1'b1: power off
10	RO	0x0	pd_hevc_pwr_st Power domain HEVC power status 1'b0: power on 1'b1: power off
9	RO	0x0	pd_gpu_pwr_st Power domain GPU power status 1'b0: power on 1'b1: power off
8	RO	0x0	pd_video_pwr_st Power domain VIDEO power status 1'b0: power on 1'b1: power off
7	RO	0x0	pd_vio_pwr_st Power domain VIO power status 1'b0: power on 1'b1: power off

Bit	Attr	Reset Value	Description
6	RO	0x0	pd_peri_pwr_st Power domain PERI power status 1'b0: power on 1'b1: power off
5	RO	0x0	pd_bus_pwr_st Power domain BUS power status 1'b0: power on 1'b1: power off
4	RO	0x0	reserved
3	RW	0x0	pd_A17_3_pwr_st Power domain A17 slave core 3 power status 1'b0: power on 1'b1: power off
2	RW	0x0	pd_A17_2_pwr_st Power domain A17 slave core 2 power status 1'b0: power on 1'b1: power off
1	RW	0x0	pd_A17_1_pwr_st Power domain A17 slave core 1 power status 1'b0: power on 1'b1: power off
0	RW	0x0	pd_A17_0_pwr_st Power domain A17 primary core power status 1'b0: power on 1'b1: power off

PMU_IDLE_REQ
Address: Operational Base + offset (0x0010)
PMU Noc idle req control

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	idle_req_hevc_cfg software config HEVC domain flush trasaction request 1'b1: idle req 1'b0: not idle req
8	RW	0x0	idle_req_cpup_cfg software config CPUP domain flush trasaction request 1'b1: idle req 1'b0: not idle req
7	RW	0x0	idle_req_dma_cfg software config DMA domain flush trasaction request 1'b1: idle req 1'b0: not idle req

Bit	Attr	Reset Value	Description
6	RW	0x0	idle_req_alive_cfg software config ALIVE domain flush trasaction request 1'b1: idle req 1'b0: not idle req
5	RW	0x0	idle_req_core_cfg software config CORE domain flush trasaction request 1'b1: idle req 1'b0: not idle req
4	RW	0x0	idle_req_vio_cfg software config VIO domain flush trasaction request 1'b1: idle req 1'b0: not idle req
3	RW	0x0	idle_req_video_cfg software config VIDEO domain flush trasaction request 1'b1: idle req 1'b0: not idle req
2	RW	0x0	idle_req_gpu_cfg software config GPU domain flush trasaction request 1'b1: idle req 1'b0: not idle req
1	RW	0x0	idle_req_peri_cfg software config PERI domain flush trasaction request 1'b1: idle req 1'b0: not idle req
0	RW	0x0	idle_req_bus_cfg software config BUS domain flush trasaction request 1'b1: idle req 1'b0: not idle req

PMU_IDLE_ST
Address: Operational Base + offset (0x0014)
PMU Noc idle status

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	idle_ack_hevc hevc domain flush transaction acknowledge 1'b0: no ack 1'b1: ack

Bit	Attr	Reset Value	Description
24	RW	0x0	idle_ack_cpup cpup domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
23	RW	0x0	idle_ack_dma dma domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
22	RW	0x0	idle_ack_alive ALIVE domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
21	RW	0x0	idle_ack_core core domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
20	RW	0x0	idle_ack_vio VIO domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
19	RW	0x0	idle_ack_video VIDEO domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
18	RW	0x0	idle_ack_gpu GPU domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
17	RW	0x0	idle_ack_peri PERI domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
16	RW	0x0	idle_ack_bus BUS domain flush transaction acknowledge 1'b0: no ack 1'b1: ack
15:10	RO	0x0	reserved
9	RW	0x0	IDLE_HEVC HEVC domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
8	RW	0x0	IDLE_CPUP CPUP domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish

Bit	Attr	Reset Value	Description
7	RW	0x0	IDLE_DMA DMA domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
6	RW	0x0	IDLE_ALIVE ALIVE domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
5	RW	0x0	IDLE_CORE CORE domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
4	RW	0x0	IDLE_VIO VIO domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
3	RW	0x0	IDLE_VIDEO VIDEO domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
2	RW	0x0	IDLE_GPU GPU domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
1	RW	0x0	IDLE_PERI PERI domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish
0	RW	0x0	IDLE_BUS BUS domain flush transaction finish(idle) 1'b0: no finish 1'b1: finish

PMU_PWRMODE_CON
Address: Operational Base + offset (0x0018)
PMU configuration register in power mode flow

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	ddr1io_ret_de_req ddr1io retention de-assert request 1'b0: de-assert request 1'b1: not de-assert request
21	RW	0x0	ddr0io_ret_de_req ddr0io retention de-assert request 1'b0: de-assert request 1'b1: not de-assert request

Bit	Attr	Reset Value	Description
20	RW	0x0	ddrc1_gating_en ddrc1 clock auto gating after self-refresh in low power mode 1'b1: auto gating 1'b0: not auto gating
19	RW	0x0	ddr0_gating_en ddrc0 auto gating in low power mode 1'b0: disable 1'b1: enable
18	RW	0x0	ddr1io_ret_en ddr1 io ret enable in low power mode 1'b1: enable 1'b0: disable
17	RW	0x0	ddr0io_ret_en DDR0 IO retention function enable or not 1'b0: DDR0 IO retention disable 1'b1: DDR0 IO retention enable
16	RW	0x0	sref1_enter_en ddr1 enter self-refresh in low power mode 1'b1: enable 1'b0: disable
15	RW	0x0	sref0_enter_en DDR0 enter self-refresh enable in low power mode 1'b0: disable DDR0 enter self-refresh 1'b1: enable DDR0 enter self-refresh
14	RW	0x0	wakeup_reset_en wakeup reset enable if power up 1'b0: diable 1'b1: enable
13	RW	0x0	input_clamp_en input clamp enable if power off input clamp for PD_PMU enable if power off 1'b0: disable 1'b1: enable
12	RW	0x0	osc_24m_dis 24MHz OSC disable in low power mode 1'b0: 24MHz OSC enable 1'b1: 24MHz OSC disable
11	RW	0x0	pmu_use_lf pmu domain clock switch to 32.768kHz enable 1'b0: not switch to 32.768kHz 1'b1: switch to 32.768kHz

Bit	Attr	Reset Value	Description
10	RW	0x0	alive_use_lf ALIVE domain clock switch to 32.768kHz enable 1'b0: not switch to 32.768kHz 1'b1: switch to 32.768kHz
9	RW	0x0	pwroff_comb three power off signal combination 1'b0: not combine 1'b1: combine enable
8	RW	0x0	chip_pd_en chip power down enable in power mode flow 1'b0: chip power on 1'b1: chip power off
7	RW	0x0	pll_pd_en pll power down enable in power mode flow 1'b0: pll power on 1'b1: pll power off
6	RW	0x0	scu_en scu power down enable in power mode flow 1'b0: scu power on 1'b1: scu power off
5	RW	0x0	A17_0_pd_en A17_0 power down in power mode flow 1'b0: A17_0 power on 1'b1: A17_0 power off
4	RW	0x0	bus_pd_en bus power off enable in low power mode 1'b0: bus power on 1'b1: bus power off
3	RW	0x0	l2flush_en l2 flush enable 1'b1: l2 flush enable 1'b0: l2 flush disable
2	RW	0x0	global_int_disable Global interrupt disable 1'b0: enable global interrupt 1'b1: disable global interrupt
1	RW	0x0	clk_core_src_gate_en A17 core clock source gating enable in idle mode 1'b0: enable 1'b1: disable

Bit	Attr	Reset Value	Description
0	RW	0x0	power_mode_en power mode flow enable 1'b0: disable 1'b1: enable

PMU_PWR_STATE
Address: Operational Base + offset (0x001c)
PMU Low power mode state

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	A17_CLK_EN A17 source clock enable 1'b0: state not happened 1'b1: state happened
29	RW	0x0	TRANS_RESTORE noc trans restore 1'b0: state not happened 1'b1: state happened
28	RW	0x0	A17_0_PWRUP A17 core0 power up state 1'b0: state not happened 1'b1: state happened
27	RW	0x0	L2MEM_PWRUP pd_l2mem powerup state 1'b0: state not happened 1'b1: state happened
26	RW	0x0	SREF_EXIT ddr exit self-refresh 1'b0: state not happened 1'b1: state happened
25	RW	0x0	DDR_IO_PWRUP ddr io powerup state 1'b0: state not happened 1'b1: state happened
24	RW	0x0	BUS_PWRUP pd_bus powerup state 1'b0: state not happened 1'b1: state happened
23	RW	0x0	PLL_PWRUP pll power up state 1'b0: state not happened 1'b1: state happened
22	RW	0x0	WAKEUP_RESET_CLR deassert wakeup reset 1'b0: state not happened 1'b1: state happened

Bit	Attr	Reset Value	Description
21	RW	0x0	ALIVE_PMU_HF pd_alive & pd_pmu switch to normal clock 1'b0: state not happened 1'b1: state happened
20	RW	0x0	X24M_OSC_EN 24MHz OSC enable 1'b0: state not happened 1'b1: state happened
19	RW	0x0	RELEASE_CLAMP release pd_pmu input clamp 1'b0: state not happened 1'b1: state happened
18	RW	0x0	EXT_PWRUP ext pmic power up 1'b0: state not happened 1'b1: state happened
17	RW	0x0	WAKEUP_RESET wakeup reset 1'b0: state not happened 1'b1: state happened
16	RW	0x0	WAIT_WAKEUP wati wakeup state 1'b0: state not happened 1'b1: state happened
15	RW	0x0	X24M_OSC_DIS 24MHz soc diable state 1'b0: state not happened 1'b1: state happened
14	RW	0x0	POWEROFF chip power off state 1'b0: state not happened 1'b1: state happened
13	RW	0x0	INPUT_CLAMP pd_pmu input clamp 1'b0: state not happened 1'b1: state happened
12	RW	0x0	PLL_PWRDN pll power down state 1'b0: state not happened 1'b1: state happened
11	RW	0x0	ALIVE_PMU_LF pd_alive&pd_pmu switch to 32khz 1'b0: state not happened 1'b1: state happened

Bit	Attr	Reset Value	Description
10	RW	0x0	L2MEM_PWRDN l2 mem power down state 1'b0: state not happened 1'b1: state happened
9	RW	0x0	A17_0_PWRDN A17 core0 power down state 1'b0: state not happened 1'b1: state happened
8	RW	0x0	BUS_PWRDN pd_bus power down state 1'b0: A17_0 power on 1'b1: A17_0 power off
7	RW	0x0	DDR_IO_PWROFF ddr io power off 1'b0: state not happened 1'b1: state happened
6	RW	0x0	DDR_IO_RET DDR io retention 1'b0: state not happened 1'b1: state happened
5	RW	0x0	SREF_ENTER ddr selfrefresh enter 1'b0: state not happened 1'b1: state happened
4	RW	0x0	TRANS_NO_FIN transfer no finish 1'b0: state not happened 1'b1: state happened
3	RW	0x0	A17_CLK_DIS A17 clock disable 1'b0: state not happened 1'b1: state happened
2	RW	0x0	STANDBYL2 L2 Standby 1'b0: state not happened 1'b1: state happened
1	RW	0x0	L2FLUSH_REQ L2 Flush req 1'b0: state not happened 1'b1: state happened
0	RW	0x0	NORMAL normal state 1'b0: state not happened 1'b1: state happened

PMU_OSC_CNT
Address: Operational Base + offset (0x0020)
24MHz OSC stabilization counter threshold

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	osc_stabl_cnt_thresh 24MHz OSC stabilization counter threshold

PMU_PLL_CNT
Address: Operational Base + offset (0x0024)
PLL lock counter threshold

Bit	Attr	Reset Value	Description
31:20	RW	0x100	pllrst_cnt_thresh PLL reset wait counter threshold
19:0	RW	0x04000	plllock_cnt_thresh PLL lock wait counter threshold

PMU_STABL_CNT
Address: Operational Base + offset (0x0028)
External PMU stabilization counter threshold

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	pmu_stabl_cnt_thresh External PMU stabilization counter threshold

PMU_DDR0IO_PWRON_CNT
Address: Operational Base + offset (0x002c)
DDR0 IO power on counter threshold

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	ddr0io_pwron_cnt_thresh DDR0 IO power on counter threshold

PMU_DDR1IO_PWRON_CNT
Address: Operational Base + offset (0x0030)
DDR1 IO power on counter threshold

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	ddr1io_pwron_cnt_thresh DDR1 IO power on counter threshold

PMU_CORE_PWRDWN_CNT
Address: Operational Base + offset (0x0034)
CORE domain power down waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	core_pwrdwn_cnt_thresh CORE domain power down waiting counter threshold

PMU_CORE_PWRUP_CNT
Address: Operational Base + offset (0x0038)
CORE domain power up waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	core_pwrup_cnt_thresh CORE domain power up waiting counter threshold

PMU_GPU_PWRDWN_CNT
Address: Operational Base + offset (0x003c)
GPU domain power down waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	gpu_pwrdwn_cnt_thresh GPU domain power down waiting counter threshold

PMU_GPU_PWRUP_CNT
Address: Operational Base + offset (0x0040)
GPU domain power up waiting counter in sleep mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	gpu_pwrup_cnt_thresh GPU domain power up waiting counter threshold

PMU_WAKEUP_RST_CLR_CNT
Address: Operational Base + offset (0x0044)
Wakeup reset deassert state wait counter in power off mode

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x05dc0	wakeup_rst_clr_cnt_thresh Power off mode wakeup reset clear counter threshold

PMU_SFT_CON
Address: Operational Base + offset (0x0048)
PMU Software control in normal mode

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	l2flush_cfg l2 flush config in normal mode 1'b1: l2 flush req 1'b0: l2 flush disable
14	RW	0x0	osc_disable_cfg software config OSC disable 1'b1: OSC disable 1'b0: OSC enable

Bit	Attr	Reset Value	Description
13	RW	0x0	osc_bypass osc bypass control 1'b0: disable 1'b1: enable
12	RW	0x0	alive_lf_ena_cfg software config ALIVE domain clock switch to 32.768kHz 1'b1: switch to 32.768kHz 1'b0: not switch
11	RW	0x0	pmu_lf_ena_cfg software config PMU domain clock switch to 32.768kHz 1'b1: switch to 32.768kHz 1'b0: not switch
10	RW	0x0	power_off_ddr1io_cfg software conifg power off DDR1 IO 1'b1: power off 1'b0: not power off
9	RW	0x0	ddr1_io_ret_cfg software config DDR1 IO retention 1'b1: retention 1'b0: not retention
8	RW	0x0	upctl1_c_sysreq_cfg software config enter DDR1 self-refresh by lowpower interface 1'b1: request enter self-refresh 1'b0: not enter self-refresh
7	RW	0x0	power_off_ddr0io_cfg software conifg power off DDR0 IO 1'b1: power off 1'b0: not power off
6	RW	0x0	ddr0_io_ret_cfg software config DDR0 IO retention 1'b1: retention 1'b0: not retention
5	RW	0x0	upctl0_c_sysreq_cfg software config enter DDR0 self-refresh by lowpower interface 1'b1: request enter self-refresh 1'b0: not enter self-refresh
4	RW	0x0	clk_core_src_gating_cfg software config A17 core clock source gating 1'b1: gating 1'b0: not gating

Bit	Attr	Reset Value	Description
3	RW	0x0	dbgnopwrdwn3_enable ARM CORE3 DBGNOPWRDWN function support enable 1'b0: not support 1'b1: support
2	RW	0x0	dbgnopwrdwn2_enable ARM CORE2 DBGNOPWRDWN function support enable 1'b0: not support 1'b1: support
1	RW	0x0	dbgnopwrdwn1_enable ARM CORE1 DBGNOPWRDWN function support enable 1'b0: not support 1'b1: support
0	RW	0x0	dbgnopwrdwn0_enable ARM CORE0 DBGNOPWRDWN function support enable 1'b0: not support 1'b1: support

PMU_DDR_SREF_ST
Address: Operational Base + offset (0x004c)
PMU DDR self refresh status

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	upctl0_c_sysack DDR0 enter self-refresh acknowledge 1'b0: no ack 1'b1: ack
2	RW	0x0	upctl0_c_active DDR0 enter self-refresh 1'b0: no active 1'b1: active
1	RW	0x0	upctl1_c_sysack DDR1 enter self-refresh acknowledge 1'b0: no ack 1'b1: ack
0	RW	0x0	upctl1_c_active DDR1 enter self-refresh 1'b0: no active 1'b1: active

PMU_INT_CON
Address: Operational Base + offset (0x0050)
PMU interrupt configuration register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	pd_mem_int_en Power domain L2 MEM power switch interrupt enable 1'b0: disable 1'b1: enable
26	RW	0x0	pd_hevc_int_en Power domain hevc power switch interrupt enable 1'b0: disable 1'b1: enable
25	RW	0x0	pd_gpu_int_en Power domain GPU power switch interrupt enable 1'b0: disable 1'b1: enable
24	RW	0x0	pd_video_int_en Power domain VIDEO power switch interrupt enable 1'b0: disable 1'b1: enable
23	RW	0x0	pd_vio_int_en Power domain VIO power switch interrupt enable 1'b0: disable 1'b1: enable
22	RW	0x0	pd_peri_int_en Power domain PERI power switch interrupt enable 1'b0: disable 1'b1: enable
21	RW	0x0	pd_bus_int_en Power domain BUS power switch interrupt enable 1'b0: disable 1'b1: enable
20	RO	0x0	reserved
19	RW	0x0	pd_a2_3_int_en Power domain A17 slave core 3 power switch interrupt enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
18	RW	0x0	pd_A17_2_int_en Power domain A17 slave core 2 power switch interrupt enable 1'b0: disable 1'b1: enable
17	RW	0x0	pd_A17_1_int_en Power domain A17 slave core 1 power switch interrupt enable 1'b0: disable 1'b1: enable
16	RW	0x0	pd_A17_0_int_en Power domain A17 primary core power switch interrupt enable 1'b0: disable 1'b1: enable
15:6	RO	0x0	reserved
5	RW	0x0	pwrmode_wakeup_int_en wakeup interrupt enable in power mode 1'b0: disable 1'b1: enable
4	RW	0x0	gpioint_wakeup_int_en gpio interrupt wakeup interrupt enable 1'b0: disable 1'b1: enable
3	RW	0x0	sdmmc0_wakeup_int_en SDMMC0 wakeup status interrupt enable 1'b0: disable 1'b1: enable
2	RW	0x0	gpio_wakeup_int_en GPIO0 wakeup status interrupt enable 1'b0: disable 1'b1: enable
1	RW	0x0	armint_wakeup_int_en ARM interrupt wakeup status interrupt enable 1'b0: disable 1'b1: enable
0	RW	0x0	pmu_int_en PMU interrupt enable 1'b0: disable 1'b1: enable

PMU_INT_ST
Address: Operational Base + offset (0x0054)
PMU interrupt status register

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

Bit	Attr	Reset Value	Description
27	RW	0x0	pd_mem_int_st Power domain l2 mem power switch status 1'b0: no power switch happen 1'b1: power switch happen
26	RW	0x0	pd_hevc_int_st Power domain HEVC power switch status 1'b0: no power switch happen 1'b1: power switch happen
25	W1C	0x0	pd_gpu_int_st Power domain GPU power switch status 1'b0: no power switch happen 1'b1: power switch happen
24	W1C	0x0	pd_video_int_st Power domain VIDEO power switch status 1'b0: no power switch happen 1'b1: power switch happen
23	W1C	0x0	pd_vio_int_st Power domain VIO power switch status 1'b0: no power switch happen 1'b1: power switch happen
22	W1C	0x0	pd_peri_int_st Power domain PERI power switch status 1'b0: no power switch happen 1'b1: power switch happen
21	W1C	0x0	pd_bus_int_st Power domain BUS power switch status 1'b0: no power switch happen 1'b1: power switch happen
20	RO	0x0	reserved
19	RW	0x0	pd_A17_3_int_st Power domain A17 slave core 3 power switch status 1'b0: no power switch happen 1'b1: power switch happen
18	RW	0x0	pd_A17_2_int_st Power domain A17 slave core 2 power switch status 1'b0: no power switch happen 1'b1: power switch happen
17	RW	0x0	pd_A17_1_int_st Power domain A17 slave core 1 power switch status 1'b0: no power switch happen 1'b1: power switch happen

Bit	Attr	Reset Value	Description
16	RW	0x0	pd_A17_0_int_st Power domain A17 primary core power switch status 1'b0: no power switch happen 1'b1: power switch happen
15:5	RO	0x0	reserved
4	RW	0x0	pwrmode_wakeup_event_trig power mode flow wakeup 1'b0: no wakeup 1'b1: wakeup
3	RW	0x0	gpioint_wakeup_event_trig ARM interrupt wake-up enent trigger 1'b0: no wakeup 1'b1: wakeup
2	W1C	0x0	sdmmc0_wakeup_event_trig SDMMC0 wake-up enent trigger 1'b0: no wakeup 1'b1: wakeup
1	W1C	0x0	gpio_wakeup_event_trig GPIO0 wake-up enent trigger 1'b0: no wakeup 1'b1: wakeup
0	RW	0x0	armint_wakeup_event_trig ARM interrupt wake-up enent trigger 1'b0: no wakeup 1'b1: wakeup

PMU_BOOT_ADDR_SEL
Address: Operational Base + offset (0x0058)
boot_addr_sel in power mode

Bit	Attr	Reset Value	Description
31:0	RW	0x00005dc0	boot_addr_sel boot addr sel when wakeup from power mode

PMU_GRF_CON
Address: Operational Base + offset (0x005c)
grf control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:10	RW	0x00	GRF_NPOR_CRNT_CTRL Npor signal crnt control register
9:4	RW	0x00	GRF_TEST_CRNT_CTRL Test signal crnt control register
3:2	RW	0x2	GRF_X32K_CRNT_CTRL X32K signal crnt control register

Bit	Attr	Reset Value	Description
1:0	RW	0x0	GRF_X24M_CRNT_CTRL X24M signal crnt control register

PMU_GPIO_SR
Address: Operational Base + offset (0x0060)
GPIO slew rate control

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RW	0x0	gpio0_c2_sr gpio0_c2 slew rate control 1'b0: slow (half frequency) 1'b1: fast
17	RW	0x1	gpio0_c1_sr gpio0_c1 slew rate control 1'b0: slow (half frequency) 1'b1: fast
16	RW	0x0	gpio0_c0_sr gpio0_c0 slew rate control 1'b0: slow (half frequency) 1'b1: fastl
15	RW	0x0	gpio0_b7_sr gpio0_b7 slew rate control 1'b0: slow (half frequency) 1'b1: fast
14	RW	0x0	gpio0_b6_sr gpio0_b6 slew rate control 1'b0: slow (half frequency) 1'b1: fast
13	RW	0x0	gpio0_b5_sr gpio0_b5 slew rate control 1'b0: slow (half frequency) 1'b1: fast
12	RW	0x0	gpio0_b4_sr gpio0_b4 slew rate control 1'b0: slow (half frequency) 1'b1: fast
11	RW	0x0	gpio0_b3_sr gpio0_b3 slew rate control 1'b0: slow (half frequency) 1'b1: fast
10	RW	0x0	gpio0_b2_sr gpio0_b2 slew rate control 1'b0: slow (half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
9	RW	0x0	gpio0_b1_sr gpio0_b1 slew rate control 1'b0: slow (half frequency) 1'b1: fast
8	RW	0x0	gpio0_b0_sr gpio0_b0 slew rate control 1'b0: slow (half frequency) 1'b1: fast
7	RW	0x0	gpio0_a7_sr gpio0_a7 slew rate control 1'b0: slow (half frequency) 1'b1: fastll
6	RW	0x0	gpio0_a6_sr gpio0_a6 slew rate control 1'b0: slow (half frequency) 1'b1: fast
5	RW	0x0	gpio0_a5_sr gpio0_a5 slew rate control 1'b0: slow (half frequency) 1'b1: fast
4	RW	0x0	gpio0_a4_sr gpio0_a4 slew rate control 1'b0: slow (half frequency) 1'b1: fast
3	RW	0x0	gpio0_a3_sr gpio0_a3 slew rate control 1'b0: slow (half frequency) 1'b1: fast
2	RW	0x0	gpio0_a2_sr gpio0_a2 slew rate control 1'b0: slow (half frequency) 1'b1: fast
1	RW	0x0	gpio0_a1_sr gpio0_a1 slew rate control 1'b0: slow (half frequency) 1'b1: fast
0	RW	0x0	gpio0_a0_sr gpio0_a0 slew rate control 1'b0: slow (half frequency) 1'b1: fast

PMU_GPIO0_A_PULL
Address: Operational Base + offset (0x0064)
GPIO0A input to PU/PD programmation section

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:14	RW	0x1	gpio0_a7_pull gpio0_a7 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
13:12	RW	0x1	gpio0_a6_pull gpio0_a6 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
11:10	RW	0x1	gpio0_a5_pull gpio0_a5 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
9:8	RW	0x1	gpio0_a4_pull gpio0_a4 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
7:6	RW	0x1	gpio0_a3_pull gpio0_a3 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
5:4	RW	0x1	gpio0_a2_pull gpio0_a2 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
3:2	RW	0x2	gpio0_a1_pull gpio0_a1 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
1:0	RW	0x2	gpio0_a0_pull gpio0_a0 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

PMU_GPIO0_B_PULL
Address: Operational Base + offset (0x0068)
GPIO0B input to PU/PD programmation section

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	RW	0x1	gpio0_b7_pull gpio0_b7 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x1	gpio0_b6_pull gpio0_b6 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x1	gpio0_b5_pull gpio0_b5 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
9:8	RW	0x1	gpio0_b4_pull gpio0_b4 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
7:6	RW	0x1	gpio0_b3_pull gpio0_b3 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
5:4	RW	0x1	gpio0_b2_pull gpio0_b2 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
3:2	RW	0x1	gpio0_b1_pull gpio0_b1 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)
1:0	RW	0x1	gpio0_b0_pull gpio0_b0 pu/pd programmation section [p2:p1] 2’b00: Z(Normal operation) 2’b01: weak 1 (pull-up) 2’b10: weak 0 (pull-down) 2’b11: repeater (bus keeper)

PMU_GPIO0_C_PULL
Address: Operational Base + offset (0x006c)
GPIO0C input to PU/PD programmation section

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio0_c2_pull gpio0_c2 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x1	gpio0_c1_pull gpio0_c1 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
1:0	RW	0x1	gpio0_c0_pull gpio0_c0 pu/pd programmation section [p2:p1] 2'b00: Z(Normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

PMU_GPIO0_A_DRV
Address: Operational Base + offset (0x0070)
GPIO0A Drive strength slector

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	RW	0x1	gpio0_a7_e gpio0_a7 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x1	gpio0_a6_e gpio0_a6 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
11:10	RW	0x1	gpio0_a5_e gpio0_a5 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x1	gpio0_a4_e gpio0_a4 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x1	gpio0_a3_e gpio0_a3 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x1	gpio0_a2_e gpio0_a2 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio0_a1_e gpio0_a1 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x2	gpio0_a0_e gpio0_a0 drive strength slector [e2:e1] 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

PMU_GPIO0_B_DRV
Address: Operational Base + offset (0x0074)
GPIO0B Drive strength slector

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	RW	0x1	gpio0_b7_e gpio0_b7 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
13:12	RW	0x1	gpio0_b6_e gpio0_b6 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
11:10	RW	0x1	gpio0_b5_e gpio0_b5 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
9:8	RW	0x1	gpio0_b4_e gpio0_b4 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
7:6	RW	0x1	gpio0_b3_e gpio0_b3 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
5:4	RW	0x1	gpio0_b2_e gpio0_b2 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA

Bit	Attr	Reset Value	Description
3:2	RW	0x1	gpio0_b1_e gpio0_b1 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
1:0	RW	0x1	gpio0_b0_e gpio0_b0 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA

PMU_GPIO0_C_DRV
Address: Operational Base + offset (0x0078)
GPIO0C Drive strength slector

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x1	gpio0_c2_e gpio0_c2 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
3:2	RW	0x1	gpio0_c1_e gpio0_c1 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA
1:0	RW	0x1	gpio0_c0_e gpio0_c0 drive strength slector [e2:e1] 2’b00: 2mA 2’b01: 4mA 2’b10: 8mA 2’b11: 12mA

PMU_GPIO_OP
Address: Operational Base + offset (0x007c)
GPIO0 output value

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RW	0x0	gpio0_c2_op gpio0_c2 output value
17	RW	0x0	gpio0_c1_op gpio0_c1 output value
16	RW	0x0	gpio0_c0_op gpio0_c0 output value
15	RW	0x0	gpio0_b7_op gpio0_b7 output value
14	RW	0x0	gpio0_b6_op gpio0_b6 output value
13	RW	0x0	gpio0_b5_op gpio0_b5 output value
12	RW	0x0	gpio0_b4_op gpio0_b4 output value
11	RW	0x0	gpio0_b3_op gpio0_b3 output value
10	RW	0x0	gpio0_b2_op gpio0_b2 output value
9	RW	0x0	gpio0_b1_op gpio0_b1 output value
8	RW	0x0	gpio0_b0_op gpio0_b0 output value
7	RW	0x0	gpio0_a7_op gpio0_a7 output value
6	RW	0x0	gpio0_a6_op gpio0_a6 output value
5	RW	0x0	gpio0_a5_op gpio0_a5 output value
4	RW	0x0	gpio0_a4_op gpio0_a4 output value
3	RW	0x0	gpio0_a3_op gpio0_a3 output value
2	RW	0x0	gpio0_a2_op gpio0_a2 output value
1	RW	0x0	gpio0_a1_op gpio0_a1 output value
0	RW	0x0	gpio0_a0_op gpio0_a0 output value

PMU_GPIO0_SEL18
Address: Operational Base + offset (0x0080)
gpio0 1.8v/3.3v sel

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

Bit	Attr	Reset Value	Description
2	RW	0x1	gpio0_c0_smt
1	RW	0x1	gpio0_b7_smt gpio0_a0 output value
0	RW	0x0	gpio0_a0_op gpio0_a0 output value 1'b0: >=2.5v 1'b1: <=1.8v

PMU_GPIO0_A_IOMUX
Address: Operational Base + offset (0x0084)
GPIO0A iomux sel

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RW	0x0	gpio0_a3 iomux 1'b0: gpioa3 1'b1: ddr1_retention
5	RO	0x0	reserved
4	RW	0x0	gpio0_a2 iomux 1'b0: gpioa2 1'b1: ddr0_retention
3	RO	0x0	reserved
2	RW	0x0	gpio0_a1 iomux 1'b0: gpioa1 1'b1: ddrio_pwroff
1	RO	0x0	reserved
0	RW	0x0	gpio0_a0 iomux 1'b0: gpioa0 1'b1: global_pwroff

PMU_GPIO0_B_IOMUX
Address: Operational Base + offset (0x0088)
GPIO0B iomux sel

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	gpio0_b7 iomux 1'b0: gpiob7 1'b1: i2c0pmu_sda
13:11	RO	0x0	reserved
10	RW	0x0	gpio0_b5 iomux 1'b0: gpiob5 1'b1: CLK_27M
9:5	RO	0x0	reserved
4	RW	0x0	gpio0_b2 iomux 1'b0: gpiob2 1'b1: tsadc_int
3:0	RO	0x0	reserved

PMU_GPIO0_C_IOMUX
Address: Operational Base + offset (0x008c)
GPIO0C iomux sel

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x0	gpio0_c1 iomux 2'b00: gpioc1 2'b01: test_clkout 2'b10: clkt1_27m 2'b11: reserved
1	RO	0x0	reserved
0	RW	0x0	gpio0_c0 iomux 1'b0: gpioc0 1'b1: i2c0pmu_scl

PMU_PWRMODE_CON1
Address: Operational Base + offset (0x0090)
PMU PowerMode CON1

Bit	Attr	Reset Value	Description
31:10	-	0x0	reserved
9	RW	0x0	clr_vio issue idle_req_vio in low power mode 1'b0: not issue 1'b1: issue
8	RW	0x0	clr_hevc issue idle_req_hevc in low power mode 1'b0: not issue 1'b1: issue
7	RW	0x0	clr_video issue idle_req_video in low power mode 1'b0: not issue 1'b1: issue
6	RW	0x0	clr_gpu issue idle_req_gpu in low power mode 1'b0: not issue 1'b1: issue
5	RW	0x0	clr_peri issue idle_req_peri in low power mode 1'b0: not issue 1'b1: issue
4	RW	0x0	clr_dma issue idle_req_dma in low power mode 1'b0: not issue 1'b1: issue

Bit	Attr	Reset Value	Description
3	RW	0x0	clr_alive issue idle_req_alive in low power mode 1'b0: not issue 1'b1: issue
2	RW	0x0	clr_cpup issue idle_req_cpup in low power mode 1'b0: not issue 1'b1: issue
1	RW	0x0	clr_core issue idle_req_core in low power mode 1'b0: not issue 1'b1: issue
0	RW	0x0	clr_bus issue idle_req_bus in low power mode 1'b0: not issue 1'b1: issue

PMU_SYS_REG0
Address: Operational Base + offset (0x0094)
PMU system register0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg0 PMU system register0

PMU_SYS_REG1
Address: Operational Base + offset (0x0098)
PMU system register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg1 PMU system register1

PMU_SYS_REG2
Address: Operational Base + offset (0x009c)
PMU system register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg2 PMU system register2

PMU_SYS_REG3
Address: Operational Base + offset (0x00a0)
PMU system register3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pmu_sys_reg3 PMU system register3

6.6 Timing Diagram

6.6.1 Each domain power switch timing

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

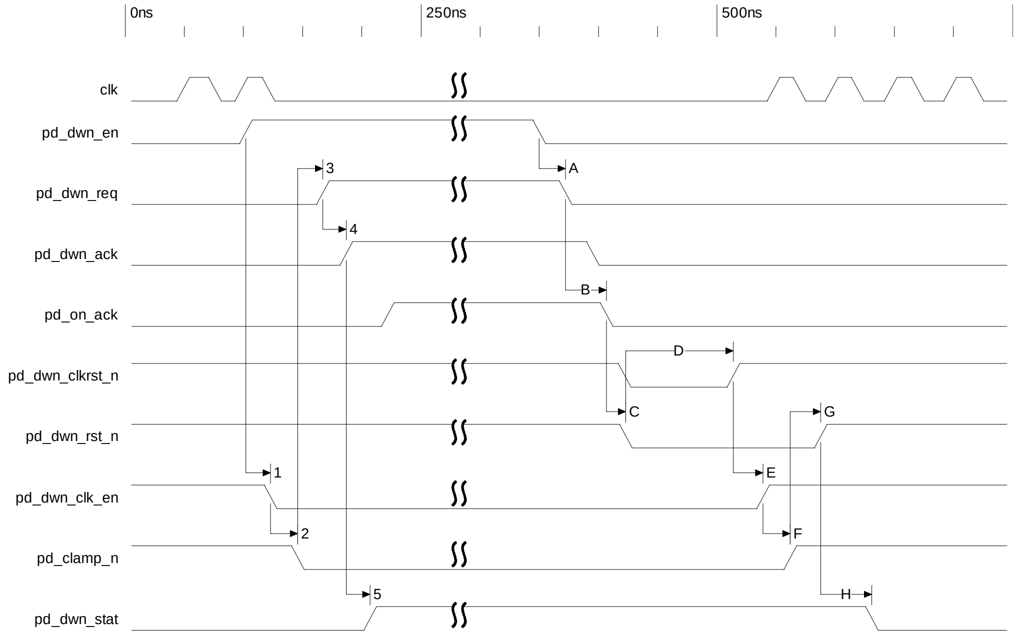


Fig. 6-3 Each Domain Power Switch Timing

6.6.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV, SIM0/1 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.

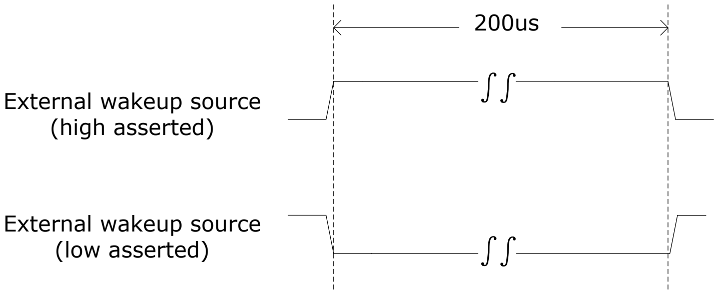


Fig. 6-4 External Wakeup Source PAD Timing

6.7 Application Notes

6.7.1 Recommend configurations for power mode.

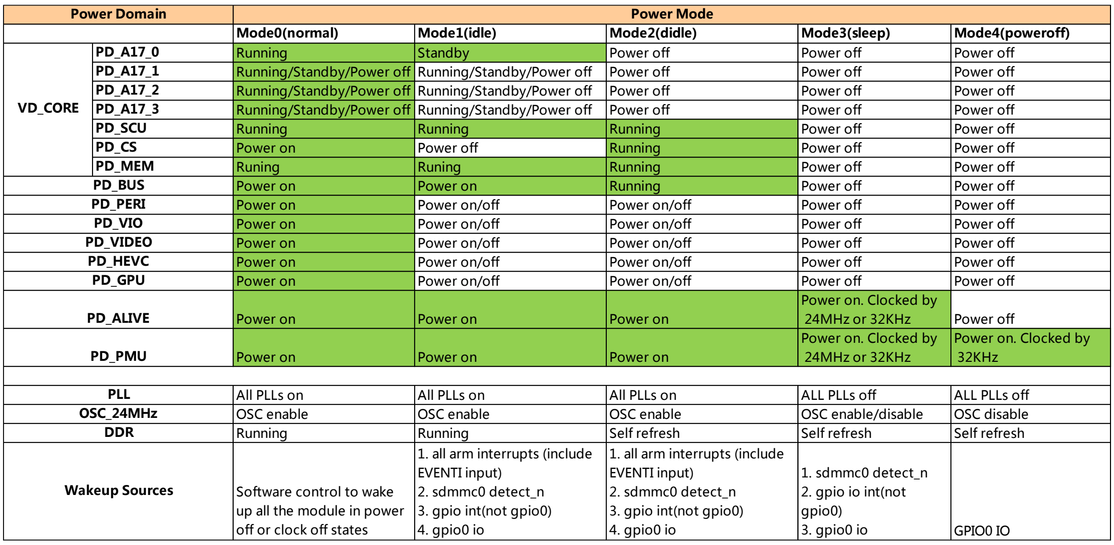
The PMU is a design with great flexibilities, but just for facilities and inheritances, a group of recommend configurations will be shown below for software. And for convenience, we will define several modes.

The RK3288 can support following 5 recommended power modes:

- normal
- idle mode
- deep idle mode
- sleep mode
- power off mode

The following table lists the detailed description of the modes.

Table 6-5 Power Domain Status Summary in all Work Mode



Power Domain		Power Mode				
		Mode0(normal)	Mode1(idle)	Mode2(didle)	Mode3(sleep)	Mode4(poweroff)
VD_CORE	PD_A17_0	Running	Standby	Power off	Power off	Power off
	PD_A17_1	Running/Standby/Power off	Running/Standby/Power off	Power off	Power off	Power off
	PD_A17_2	Running/Standby/Power off	Running/Standby/Power off	Power off	Power off	Power off
	PD_A17_3	Running/Standby/Power off	Running/Standby/Power off	Power off	Power off	Power off
	PD_SCU	Running	Running	Running	Power off	Power off
	PD_CS	Power on	Power off	Running	Power off	Power off
	PD_MEM	Runing	Runing	Running	Power off	Power off
PD_BUS		Power on	Power on	Running	Power off	Power off
PD_PERI		Power on	Power on/off	Power on/off	Power off	Power off
PD_VIO		Power on	Power on/off	Power on/off	Power off	Power off
PD_VIDEO		Power on	Power on/off	Power on/off	Power off	Power off
PD_HEVC		Power on	Power on/off	Power on/off	Power off	Power off
PD_GPU		Power on	Power on/off	Power on/off	Power off	Power off
PD_ALIVE		Power on	Power on	Power on	Power on. Clocked by 24MHz or 32KHz	Power off
PD_PMU		Power on	Power on	Power on	Power on. Clocked by 24MHz or 32KHz	Power on. Clocked by 32KHz
PLL		All PLLs on	All PLLs on	All PLLs on	ALL PLLs off	ALL PLLs off
OSC_24MHz		OSC enable	OSC enable	OSC enable	OSC enable/disable	OSC disable
DDR		Running	Running	Self refresh	Self refresh	Self refresh
Wakeup Sources		Software control to wake up all the module in power off or clock off states	1. all arm interrupts (include EVENTI input) 2. sdmmc0 detect_n 3. gpio int(not gpio0) 4. gpio0 io	1. all arm interrupts (include EVENTI input) 2. sdmmc0 detect_n 3. gpio int(not gpio0) 4. gpio0 io	1. sdmmc0 detect_n 2. gpio io int(not gpio0) 3. gpio0 io	GPIO0 IO

Normal mode

In this mode, you can power off/on or enable/disable the following power domain to save power: PD_PERI/PD_VIO/PD_VIDEO/PD_HEVC/PD_GPU

Idle mode

This mode is used when the core do not have load for a shot while such as waiting for interrupt and the software want to save power by gating Cortex-A17 source clock.

In idle mode, core1/2/3 of Cortex-A17 should be either power off or in WFI/WFE state. The core0 of A17 should be in WFI/WFE state. The configurations of core clock source gating and disable global interrupt are presented. The Cortex-A17 can waked up by an interrupt.

Deep idle mode

Deep idle mode is used in the scenario of audio player. In deep idle mode, powering off Cortex-A17 cores or VD_CORE voltage domain is operational, and others are same as normal mode.

In deep idle mode, you can set ddr enter the self-refresh, and power off DDRIO and enable DDR retention function in this mode, but it will takes a longer time for the recovery of DDR IO.

Sleep mode

The sleep mode can power off all power domains except PD_ALIVE. The VD_CORE is turned off externally, PD_BUS power off by hardware, and other domains power off by software.

In sleep mode the clock of PD_ALIVE can be switched from 24MHz to 32.768kHz optionally by hardware.

In sleep mode all PLLs power down mandatorily to save power by hardware.

In sleep mode OSC can be disabled optionally by hardware.

In sleep mode DDR self-refresh can be issued by hardware mandatorily.

In sleep mode DDR IO can power off and enter retention optionally by hardware.

Power off mode

The power off mode turns off the power of all VD_LOGIC externally.

In power off mode all PLLs power down mandatorily to save power by hardware.

In power off mode OSC disable request should be send by hardware.

In power off mode DDR self-refresh should be issued mandatorily by hardware.

In power off mode DDR IO can power off and enter retention optionally by hardware.

6.7.2 System Register

PMU support 4 words register: PMU_SYS_REG0, PMU_SYS_REG1, PMU_SYS_REG2, PMU_SYS_REG3. These registers are always on no matter what low power mode. So software can use these registers to retain some information which is useful after wakeup from any mode.

6.7.3 Configuration Constraint

In order to shut down the power domains correctly, the software must obey the rules bellow:

Send NIU request to the NIU in power domain that you want to shut down.

Querying PMU_NOC_ST register to get the information until the pacific NIU is in idle state.

Send power request to the power domain through PMU_PWRDN_CON register.

Querying PMU_PWRDN_ST register to make sure the pacific power domain is power down.

The power domains controlled only by software are showing below:
PD_VIO, PD_PERI, PD_GPU, PD_VIDEO, PD_HEVC, and PD_A17_1 and PD_A17_2 to PD_A17_3.
So you must power off these power domains before enter low power mode if you need.

6.7.4 Poweroff Request Combine

There is only two poweroff request, one is for VD_CORE and VD_LOGIC (power_off_req), another is for DDRIO (power_off_ddrio).

POWER_OFF_REQ

In normal mode, If you set the chip power down (bit[13] of PMU_PWRDN_CON), or set core power down (bit[12] of PMU_PWRDN_CON), the power_off_req will be set to 1 immediate.
In Low Power mode, if you set the chip power down (bit[8] of PMU_PWRMODE_CON) or set core power down (bit[6] of PMU_PWRMODE_CON), the power_off_req will be set to 1 at the sub-step POWEROFF.

POWER_OFF_DDRIO

In normal mode, If you set the power_off_ddr0io_cfg (bit[7] of PMU_SFT_CON), or set power_off_ddr1io_cfg (bit[10] of PMU_SFT_CON), the power_off_ddrio will be set to 1 immediate.
In Low Power mode, if you set the ddr0io_ret_en (bit[17] of PMU_PWRMODE_CON) or set ddr1io_ret_en (bit[18] of PMU_PWRMODE_CON), the power_off_ddrio will be set to 1 at the sub-step DDR_IO_POWEROFF.
In Low Power mode, if you set pwroff_comb (bit[9] of PMU_PWRMODE_CON) at same time, the power_off_ddrio would not be set to 1, and the power_off_req would be set to 1 at the sub-step DDR_IO_POWEROFF.

Chapter 7 Memory Management Unit (MMU)

7.1 Overview

An MMU controls address translation, access permissions, memory attribute.determination, and checking at a memory system level.
The MMU used in other modules will be introduced below.

7.2 Block Diagram

The MMU divides memory into 4KB pages, where each page can be individually configured.
The MMU uses a 2-level page table structure:
1. The first level, the Page Directory consists of 1024 Directory Table Entries (DTEs), each pointing to a Page Table.
2. The second level, the Page Table consists of 1024 Page Table Entries (PTEs), each pointing to a page in memory
Fig. 11-1 shows the structure of the two-level page table.

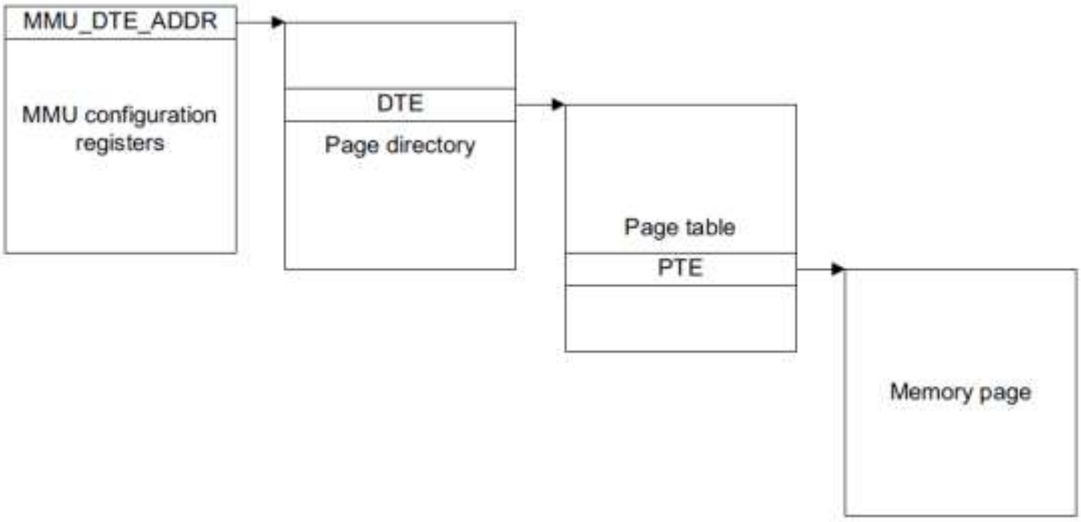


Fig. 7-1 Power Domain Partition

Fig. 11-2 shows the arrangement of the MMU address bits.

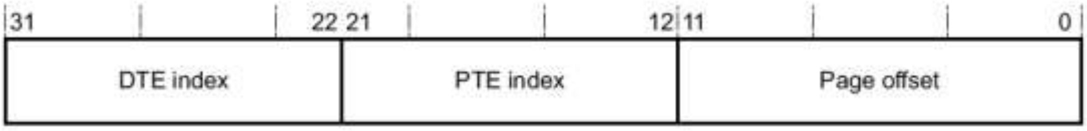


Fig. 7-2 Power Domain Partition

The MMU uses the following algorithm to translate an address:

1. Find the DTE at address given by:
MMU_DTE_ADDR + (4 * DTE index)
2. Find the PTE at address given by:
Page table address from DTE + (4 * PTE index)
3. Calculate effective address as follows:
Page address from PTE + Page offset.

The page directory is a 4KB data structure that contains 1024 32-bit DTEs. The page directory must align at a 4KB boundary in memory.

Each DTE contains the address of a page table and a page table present bit. The system:

- initializes the entire page directory before use
- clears the page table present bit for any DTE that does not point to a valid page table.

Fig. 11-3 shows the page bit assignments.

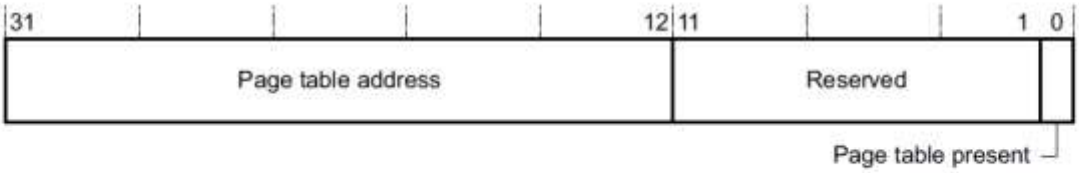


Fig. 7-3 Page directory entry bit assignments

Bits	Name	Function
[31:12]	Page table address	This field stores bits [31:12] of the address for a page table
[11:1]	Reserved	Reserved, write as zero
[0]	Page table present	This bit indicates when the page table address points to a valid page table. 0 = page table not valid 1= page table valid.

The page table is a 4KB data structure containing 1024 32-bit PTEs. The page table must be aligned at a 4KB boundary in memory.

Each PTE contains the address of a page of memory, a Page Table present bit, and Read/Write Permission bits. The entire Page Table must be initialized before use, and any PTE not pointing to a valid page must clear the Page Present bit.

Fig.11-3 shows the page table entry bit assignments.

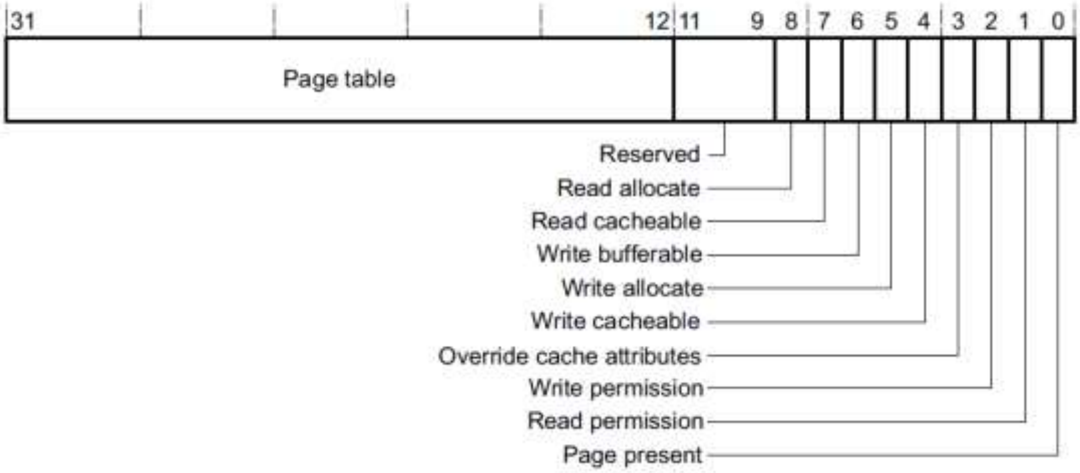


Fig. 7-4 Page directory entry bit assignments

Bits	Name	Function
[31:12]	Page table address	This field stores bits [31:12] of the address for a page table
[11:9]	Reserved	Reserved, write as zero
[8]	Read allocate	If set, allocate cache space on read misses. Must not be set if the Read cacheable bit is not set. Only used for reads, if the Override cache attributes bit is set.
[7]	Read cacheable	If set, enable caching or prefetching of data. Only used for reads, if the Override cache attributes bit is set.
[6]	Write bufferable	If set, enable write to be delayed on their way to memory. Only used for writes, if the Override cache attributes bit is set.
[5]	Write allocate	If set, allocate cache space on write misses. Must not be set if the Write cachecable bit is not set. Only used for writes, if the Override cache attributes is set.
[4]	Write cachecable	If set, enable different writes to be merged together. Only used for writes, if the Override cache attributes bit is set.
[3]	Override cache attributes	If set, the cacheability attributes specified in bits [8:4] are used to control the cache attribytes used on the memory bus. If cleared, the default cacheability attributes from the specific processors are used on the system bus.
[2]	Write permission	Enable write accesses to the page, if present.
[1]	Read permission	Enable read accesses from the page, if present.
[0]	Page present	This bit indicates when the page table field points to a valid page. 0 = page not valid 1 = page valid.

7.3 Register Description

7.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
MMU_DTE_ADDR	0x0000	W	0x00000000	MMU current page table address
MMU_STATUS	0x0004	W	0x00000018	MMU status register
MMU_CMD	0x0008	W	0x00000000	MMU command register
MMU_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logic address of last page fault register
MMU_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU zap cache line register
MMU_INT_RAWSTAT	0x0014	W	0x00000000	MMU raw interrupt status register
MMU_INT_CLEAR	0x0018	W	0x00000000	MMU interrupt clear register
MMU_INT_MASK	0x001c	W	0x00000000	MMU interrupt mask register
MMU_INT_STATUS	0x0020	W	0x00000000	MMU interrupt status register

Name	Offset	Size	Reset Value	Description
MMU_AUTO_GATING	0x0024	W	0x00000000	clock atuo gating register

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

7.3.2 Detail Register Description

MMU_DTE_ADDR

Address: Operational Base + offset (0x0000)

MMU current page table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_dte_addr page table address

MMU_STATUS

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	mmu_page_fault_bus_id Index of master responsible for the last page fault
5	RO	0x0	mmu_page_fault_is_write The direction of access for last page fault: 0: read 1:write
4	RO	0x1	mmu_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 access. The MMU_IDLE signal only reports idle when the MMU processor is idle and accesses are active on the external bus. Note: the MMU can be idle in page fault mode.
2	RO	0x0	mmu_stall_active MMU stall mode currently enabled. The mode is enabled by command.
1	RO	0x0	mmu_page_fault_active MMU page fault mode currently enabled.The mode is enabled by command
0	RO	0x0	mmu_paging_enabled mmu paging is enabled.

MMU_CMD

Address: Operational Base + offset (0x0008)

MMU command register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x0	mmu_cmd 0: MMU_ENABLE_PAGING. enable paging. 1: MMU_DISABLE_PAGING. disable paging. 2: MMU_ENABLE_STALL. turn on stall mode. 3: MMU_DISABLE_STALL. turn off stall mode. 4: MMU_ZAP_CACHE. zap the entire page table cache. 5: MMU_PAGE_FAULT_DONE. leave page fault mode. 6: MMU_FORCE_RESET. reset the mmu. The MMU_ENABLE_STALL command can always be issued. Other commands are ignored unless the MMU is idle or stalled.

MMU_PAGE_FAULT_ADDR
Address: Operational Base + offset (0x000c)
MMU logic address of last page fault register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	mmu_page_fault_addr address of last page fault

MMU_ZAP_ONE_LINE
Address: Operational Base + offset (0x0010)
MMU zap cache line register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_zap_one_line address to be invalidated from the page table cache.

MMU_INT_RAWSTAT
Address: Operational Base + offset (0x0014)
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

MMU_INT_CLEAR
Address: Operational Base + offset (0x0018)
MMU interrupt clear register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_clear read bus error interrupt clear. write 1 to this register can clear read bus error interrupt.

Bit	Attr	Reset Value	Description
0	RW	0x0	page_fault_clear page fault interrupt clear, write 1 to this register can clear page fault interrupt.

MMU_INT_MASK
Address: Operational Base + offset (0x001c)
MMU interrupt mask register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_int_en read bus error interrupt enable
0	RW	0x0	page_fault_int_en page fault interrupt enable

MMU_INT_STATUS
Address: Operational Base + offset (0x0020)
MMU interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error interrupt
0	RO	0x0	page_fault page fault interrupt

MMU_AUTO_GATING
Address: Operational Base + offset (0x0024)
clock atuo gating register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	mmu_atuo_gating mmu clock auto gating when it is 1, the mmu will auto gating itself

7.4 MMU Base address

The table below shows the MMU base address in different modules.

ISP_MMU0_BASE	ISP_BASEADDR + 0x4000
ISP_MMU1_BASE	ISP_BASEADDR + 0x5000
VOP_BIG_MMU_BASE	VOP_BIG_BASEADDR + 0x300
VOP_LIT_MMU_BASE	VOP_LIT_BASEADDR + 0x300
IEP_MMU_BASE	IEP_BASE + 0x800

VIP_MMU_BASE	VIP_BASE + 0x800

Chapter 8 Timer

8.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device.
Timer0~4 and Timer6 count down from a programmed value and generate an interrupt when the count reaches zero.
Timer5 and Timer7 count up from zero to a programmed value and generate an interrupt when the count reaches the programmed value.
Timer supports the following features:

- Two APB timers in the soc system. One is in the alive subsystem, include two programmable 64 bits timer channel, acts as TIMER6 and TIMER7; The other is in the cpu subsystem, include six programmable 64 bits timer channel, acts as TIMER0, TIMER1, TIMER2, TIMER3, TIMER4, and TIMER5 respectively.
- Two operation modes: free-running and user-defined count.
- Maskable for each individual interrupt.
- TIMER5 is used for gpu ; TIMER7 is used for core.

8.2 Block Diagram

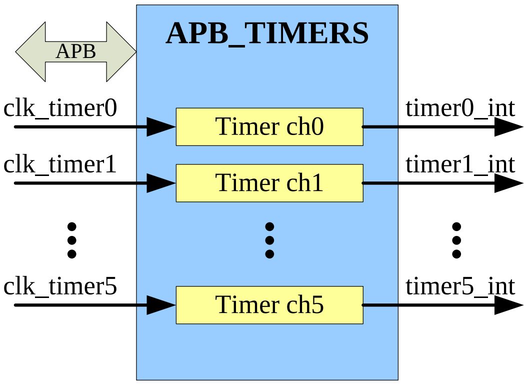


Fig. 8-1 Timers Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channel) that in the cpu subsystem. The other APB timers that in the alive subsystem only include one programmable timer channel.

8.3 Function description

8.3.1 Timer clock

TIMER0, TIMER1, TIMER2, Timer3, TIMER4 and TIMER5 are in the CPU subsystem, using timer ch0 ~ ch5 respectively. The timer clock is 24MHz OSC.
TIMER6 and TIMER7 are in the ALIVE subsystem, using timer ch0 ~ ch1. The timer clock is 24MHz OSC.

8.3.2 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).

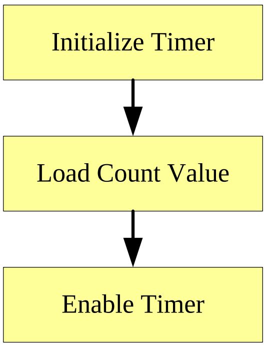


Fig. 8-2 Timer Usage Flow

8.3.3 Loading a timer count value

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 (0≤n≤5) and TIMERn_ LOAD_COUNT0 (0≤n≤5)). 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.

8.3.4 Timer mode selection

- User-defined count mode – Timer loads TIMERn_LOAD_COUNT1 (0≤n≤5) and TIMERn_ LOAD_ COUNT0 (0≤n≤5) register as initial value. Timer will not automatically load the count register, when timer counts down to 0. 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 (0≤n≤5) and TIMERn_ LOAD_ COUNT0 (0≤n≤5) register as initial value. Timer will automatically load the count register, when timer counts down to 0.

8.4 Register Description

This section describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

8.4.1 Registers Summary

Timer0 - Timer5
Base = 0xff6b_0000

Name	Offset	Size	Reset Value	Description
TIMER0_LOAD_COUNT0	0x0000	W	0x00000000	Timer0 Load Count Register
TIMER0_LOAD_COUNT1	0x0004	W	0x00000000	Timer0 Load Count Register
TIMER0_CURRENT_VALUE0	0x0008	W	0x00000000	Timer0 Current Value Register
TIMER0_CURRENT_VALUE1	0x000C	W	0x00000000	Timer0 Current Value Register
TIMER0_CONTROLREG	0x0010	W	0x00000000	Timer0 Control Register
TIMER0_INTSTATUS	0x0018	W	0x00000000	Timer0 Interrupt Status Register
TIMER1_LOAD_COUNT0	0x0020	W	0x00000000	Timer1 Load Count Register
TIMER1_LOAD_COUNT1	0x0024	W	0x00000000	Timer1 Load Count Register
TIMER1_CURRENT_VALUE0	0x0028	W	0x00000000	Timer1 Current Value Register
TIMER1_CURRENT_VALUE1	0x002c	W	0x00000000	Timer1 Current Value Register
TIMER1_CONTROLREG	0x0030	W	0x00000000	Timer1 Control Register
TIMER1_INTSTATUS	0x0038	W	0x00000000	Timer1 Interrupt Status Register
TIMER2_LOAD_COUNT0	0x0040	W	0x00000000	Timer2 Load Count Register
TIMER2_LOAD_COUNT1	0x0044	W	0x00000000	Timer2 Load Count Register
TIMER2_CURRENT_VALUE0	0x0048	W	0x00000000	Timer2 Current Value Register
TIMER2_CURRENT_VALUE1	0x004c	W	0x00000000	Timer2 Current Value Register

Name	Offset	Size	Reset Value	Description
TIMER2_CONTROLREG	0x0050	W	0x00000000	Timer2 Control Register
TIMER2_INTSTATUS	0x0058	W	0x00000000	Timer2 Interrupt Status Register
TIMER3_LOAD_COUNT0	0x0060	W	0x00000000	Timer3 Load Count Register
TIMER3_LOAD_COUNT1	0x0064	W	0x00000000	Timer3 Load Count Register
TIMER3_CURRENT_VALUE0	0x0068	W	0x00000000	Timer3 Current Value Register
TIMER3_CURRENT_VALUE1	0x006c	W	0x00000000	Timer3 Current Value Register
TIMER3_CONTROLREG	0x0070	W	0x00000000	Timer3 Control Register
TIMER3_INTSTATUS	0x0078	W	0x00000000	Timer3 Interrupt Status Register
TIMER4_LOAD_COUNT0	0x0080	W	0x00000000	Timer4 Load Count Register
TIMER4_LOAD_COUNT1	0x0084	W	0x00000000	Timer4 Load Count Register
TIMER4_CURRENT_VALUE0	0x0088	W	0x00000000	Timer4 Current Value Register
TIMER4_CURRENT_VALUE1	0x008c	W	0x00000000	Timer4 Current Value Register
TIMER4_CONTROLREG	0x0090	W	0x00000000	Timer4 Control Register
TIMER4_INTSTATUS	0x0098	W	0x00000000	Timer4 Interrupt Status Register
TIMER5_LOAD_COUNT0	0x00a0	W	0x00000000	Timer5 Load Count Register
TIMER5_LOAD_COUNT1	0x00a4	W	0x00000000	Timer5 Load Count Register
TIMER5_CURRENT_VALUE0	0x00a8	W	0x00000000	Timer5 Current Value Register
TIMER5_CURRENT_VALUE1	0x00ac	W	0x00000000	Timer5 Current Value Register
TIMER5_CONTROLREG	0x00b0	W	0x00000000	Timer5 Control Register
TIMER5_INTSTATUS	0x00b8	W	0x00000000	Timer5 Interrupt Status Register

Timer6-7
Base: 0xff81_0000

Name	Offset	Size	Reset Value	Description
TIMER6_LOAD_COUNT0	0x0000	W	0x00000000	Timer6 Load Count Register
TIMER6_LOAD_COUNT1	0x0004	W	0x00000000	Timer6 Load Count Register
TIMER6_CURRENT_VALUE0	0x0008	W	0x00000000	Timer6 Current Value Register
TIMER6_CURRENT_VALUE1	0x000C	W	0x00000000	Timer6 Current Value Register
TIMER6_CONTROLREG	0x0010	W	0x00000000	Timer6 Control Register
TIMER6_INTSTATUS	0x0018	W	0x00000000	Timer6 Interrupt Status Register
TIMER7_LOAD_COUNT0	0x0020	W	0x00000000	Timer7 Load Count Register
TIMER7_LOAD_COUNT1	0x0024	W	0x00000000	Timer7 Load Count Register
TIMER7_CURRENT_VALUE0	0x0028	W	0x00000000	Timer7 Current Value Register
TIMER7_CURRENT_VALUE1	0x002c	W	0x00000000	Timer7 Current Value Register
TIMER7_CONTROLREG	0x0030	W	0x00000000	Timer7 Control Register
TIMER7_INTSTATUS	0x0038	W	0x00000000	Timer7 Interrupt Status Register

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

8.4.2 Detail Register Description

TIMERn_LOAD_COUNT0
Address: Operational Base + offset(0x00+n*0x20)
Timer n Load Count Register (0≤n≤5)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Low 32 bits Value to be loaded into Timer n. This is the value from which counting commences.

TIMERn_LOAD_COUNT1
Address: Operational Base + offset(0x04+n*0x20)
Timer n Load Count Register (0≤n≤5)

Bit	Attr	Reset Value	Description
31:0	RW	0x0	High 32 bits Value to be loaded into Timer n. This is the value from which counting commences.

TIMERn_CURRENT_VALUE0
Address: Operational Base + offset(0x08+n*0x20)
Timer n Current Value Register (0≤n≤5)

Bit	Attr	Reset Value	Description
31:0	R	0x0	Low 32 bits of Current Value of Timer n.

TIMERn_CURRENT_VALUE1
Address: Operational Base + offset(0x0c+n*0x20)
Timer n Current Value Register (0≤n≤5)

Bit	Attr	Reset Value	Description
31:0	R	0x0	High 32 bits of Current Value of Timer n.

TIMERn_CONTROLREG
Address: Operational Base + offset(0x10+n*0x20)
Timer n Control Register (0≤n≤5)

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	RW	0x0	Timer interrupt mask. 1’b0: mask 1’b1: not mask
1	RW	0x0	Timer mode. 1’b0: free-running mode 1’b1: user-defined count mode
0	RW	0x0	Timer enable. 1’b0: disable 1’b1: enable

TIMERn_INTSTATUS
Address: Operational Base + offset(0x18+n*0x20)
Timer n Interrupt Status Register (0≤n≤5)

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	This register contains the interrupt status for timer n Write 1 to this register will clear the interrupt

Notes: Attr: ***RW*** *– Read/writable,* ***R*** *– read only,* ***W*** *– write only*

8.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disable the timer enable bit (bit 0 of TIMERn_CONTROLREG (0≤n≤5)), the timer_en output signal is de-asserted, and timer_clk will stop. When user enable 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.

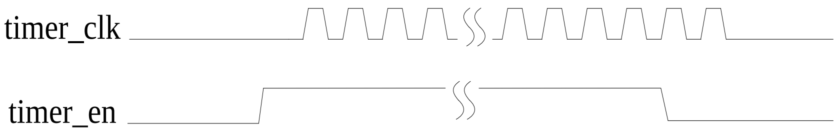


Fig. 8-3 Timing between timer_en and timer_clk

Please refer to funciton description section for the timer usage flow.

Chapter 9 Interrupt Controller

9.1 Overview

The generic interrupt controller (GIC400) in this device has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A17.

It supports the following features:

Supports 160 hardware interrupt inputs
Masking of any interrupts
Prioritization of interrupts
Distribution of the interrupts to the target Cortex-A17 processor(s)
Generation of interrupts by software
Supports Security Extensions

9.2 Block Diagram

The generic interrupt controller comprises with:

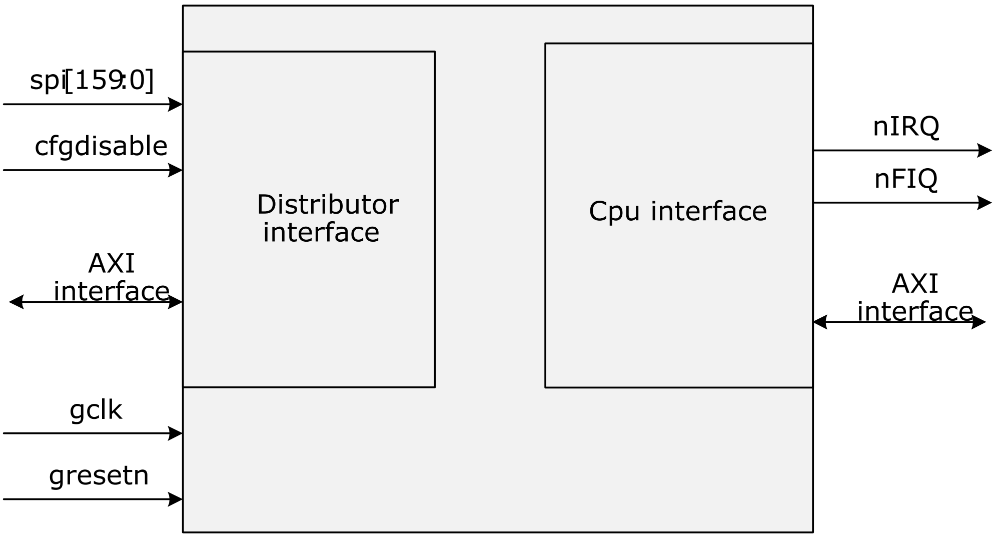


Fig. 9-1 Block Diagram

9.3 Function Description

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

Chapter 10 DMA Controller for Bus System (DMAC_BUS)

10.1 Overview

This device supports 2 Direct Memory Access (DMA) tops, one for cpu system (DMAC_BUS), and the other one for Peripheral system (PERI_DMAC).Both of these two dma support transfers between memory and memory, peripheral and memory.

DMAC_BUS supports TrustZone technology and is under secure state after reset. The secure state can be changed by configuring TZPC module.

DMAC_BUS is mainly used for data transfer of the following slaves: I2S0/I2S1/ SPDIF/UART0/Embedded SRAM and transfer data from/to external DDR SDRAM.

Following table shows the DMAC_BUS peripheral request mapping scheme.

Table 10-1 DMAC_BUS Request Mapping Table

Req number	Source	Polarity
0	I2S tx	High level
1	I2S rx	High level
2	SPDIF	High level
3	SPDIF (8ch)	High level
4	UART DBG tx	High level
5	UART DBG rx	High level

DMAC_BUS supports the following features:

- Supports Trustzone technology
- Supports 6 peripheral request
- Up to 64bits data size
- 5 channel at the same time
- Up to burst 16
- 10 interrupt output and 1 abort output
- Supports 32 MFIFO depth

10.2 Block Diagram

Figure 14-1 shows the block diagram of DMAC_BUS

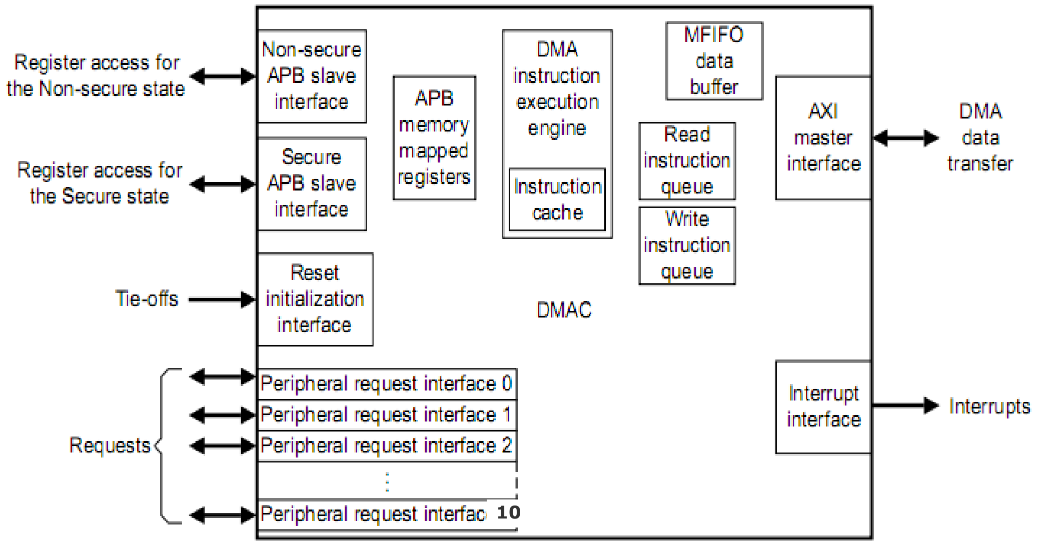

Register access for the Non-secure state
Register access for the Secure state
Tie-offs
Requests
Non-secure APB slave interface
Secure APB slave interface
Reset initialization interface
APB memory mapped registers
DMA instruction execution engine
Instruction cache
MFIFO data buffer
Read instruction queue
Write instruction queue
AXI master interface
DMA data transfer
DMAC
Peripheral request interface 0
Peripheral request interface 1
Peripheral request interface 2
Peripheral request interfac 10
Interrupt interface
Interrupts

Fig. 10-1 Block diagram of DMAC_BUS

As the DMAC_BUS supports Trustzone technology, so dual APB interfaces enable the operation of the DMAC_BUS 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_BUS.The default interface after reset is secure apb interface.

10.3 Function Description

10.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache.

DMAC_BUS supports 6 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.

When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete.

When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

10.3.2 Operating states

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

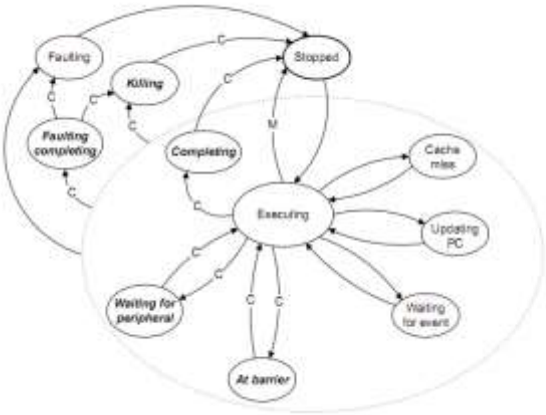


Fig. 10-2 DMAC_BUS operation states

Note:
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 the status of boot_from_pc(tie-off interface of dmac) controls the DMA manager thread state:

boot_from_pc is LOW :DMA manager thread moves to the Stopped state.
boot_from_pc is HIGH :DMA manager thread moves to the Executing state.

10.4 Register Description

10.4.1 Register summary

Name	Offset	Size	Reset Value	Description
DMAC_BUS_DSR	0x0000	W	0x0	DMA Status Register.
DMAC_BUS_DPC	0x0004	W	0x0	DMA Program Counter Register.
-	-	-	-	reserved
DMAC_BUS_INTEN	0x0020	W	0x0	Interrupt Enable Register
DMAC_BUS_EVENT_RIS	0x0024	W	0x0	Event Status Register.
DMAC_BUS_INTMIS	0x0028	W	0x0	Interrupt Status Register
DMAC_BUS_INTCLR	0x002C	W	0x0	Interrupt Clear Register
DMAC_BUS_FSRD	0x0030	W	0x0	Fault Status DMA Manager Register.
DMAC_BUS_FSRC	0x0034	W	0x0	Fault Status DMA Channel Register.
DMAC_BUS_FTRD	0x0038	W	0x0	Fault Type DMA Manager Register.
-	-	-	-	reserved
DMAC_BUS_FTR0	0x0040	W	0x0	Fault type for DMA Channel 0
DMAC_BUS_FTR1	0x0044	W	0x0	Fault type for DMA Channel 1
DMAC_BUS_FTR2	0x0048	W	0x0	Fault type for DMA Channel 2
DMAC_BUS_FTR3	0x004C	W	0x0	Fault type for DMA Channel 3
DMAC_BUS_FTR4	0x0050	W	0x0	Fault type for DMA Channel 4
DMAC_BUS_FTR5	0x0054	W	0x0	Fault type for DMA Channel 5
-	-	-	-	reserved
DMAC_BUS_CSR0	0x0100	W	0x0	Channel Status for DMA Channel 0

Name	Offset	Size	Reset Value	Description
DMAC_BUS_CSR1	0x0108	W	0x0	Channel Status for DMA Channel 1
DMAC_BUS_CSR2	0x0110	W	0x0	Channel Status for DMA Channel 2
DMAC_BUS_CSR3	0x0118	W	0x0	Channel Status for DMA Channel 3
DMAC_BUS_CSR4	0x0120	W	0x0	Channel Status for DMA Channel 4
DMAC_BUS_CSR5	0x0128	W	0x0	Channel Status for DMA Channel 5
DMAC_BUS_CPC0	0x0104	W	0x0	Channel PC for DMA Channel 0
DMAC_BUS_CPC1	0x010c	W	0x0	Channel PC for DMA Channel 1
DMAC_BUS_CPC2	0x0114	W	0x0	Channel PC for DMA Channel 2
DMAC_BUS_CPC3	0x011c	W	0x0	Channel PC for DMA Channel 3
DMAC_BUS_CPC4	0x0124	W	0x0	Channel PC for DMA Channel 4
DMAC_BUS_CPC5	0x012c	W	0x0	Channel PC for DMA Channel 5
DMAC_BUS_SAR0	0x0400	W	0x0	Source Address for DMA Channel 0
DMAC_BUS_SAR1	0x0420	W	0x0	Source Address for DMA Channel 1
DMAC_BUS_SAR2	0x0440	W	0x0	Source Address for DMA Channel 2
DMAC_BUS_SAR3	0x0460	W	0x0	Source Address for DMA Channel 3
DMAC_BUS_SAR4	0x0480	W	0x0	Source Address for DMA Channel 4
DMAC_BUS_SAR5	0x04a0	W	0x0	Source Address for DMA Channel 5
DMAC_BUS_DAR0	0x0404	W	0x0	Dest Address for DMAChannel 0
DMAC_BUS_DAR1	0x0424	W	0x0	Dest Address for DMAChannel 1
DMAC_BUS_DAR2	0x0444	W	0x0	Dest Address for DMAChannel 2
DMAC_BUS_DAR3	0x0464	W	0x0	Dest Address for DMAChannel 3
DMAC_BUS_DAR4	0x0484	W	0x0	Dest Address for DMAChannel 4
DMAC_BUS_DAR5	0x04a4	W	0x0	Dest Address for DMAChannel 5
DMAC_BUS_CCR0	0x0408	W	0x0	Channel Control for DMA Channel 0
DMAC_BUS_CCR1	0x0428	W	0x0	Channel Control for DMA Channel 1
DMAC_BUS_CCR2	0x0448	W	0x0	Channel Control for DMA Channel 2
DMAC_BUS_CCR3	0x0468	W	0x0	Channel Control for DMA Channel 3
DMAC_BUS_CCR4	0x0488	W	0x0	Channel Control for DMA Channel 4
DMAC_BUS_CCR5	0x04a8	W	0x0	Channel Control for DMA Channel 5
DMAC_BUS_LC0_0	0x040C	W	0x0	Loop Counter 0 for DMA Channel 0
DMAC_BUS_LC0_1	0x042C	W	0x0	Loop Counter 0 for DMA Channel 1
DMAC_BUS_LC0_2	0x044C	W	0x0	Loop Counter 0 for DMA Channel 2
DMAC_BUS_LC0_3	0x046C	W	0x0	Loop Counter 0 for DMA Channel 3
DMAC_BUS_LC0_4	0x048C	W	0x0	Loop Counter 0 for DMA Channel 4
DMAC_BUS_LC0_5	0x04aC	W	0x0	Loop Counter 0 for DMA Channel 5
DMAC_BUS_LC1_0	0x0410	W	0x0	Loop Counter 1 for DMA Channel 0
DMAC_BUS_LC1_1	0x0430	W	0x0	Loop Counter 1 for DMA Channel 1
DMAC_BUS_LC1_2	0x0450	W	0x0	Loop Counter 1 for DMA Channel 2
DMAC_BUS_LC1_3	0x0470	W	0x0	Loop Counter 1 for DMA Channel 3
DMAC_BUS_LC1_4	0x0490	W	0x0	Loop Counter 1 for DMA Channel 4
DMAC_BUS_LC1_5	0x04b0	W	0x0	Loop Counter 1 for DMA Channel 5
-	-	-	-	reserved
DMAC_BUS_DBGST DMAC_BUS_ATUS	0x0D00	W	0x0	Debug Status Register.
DMAC_BUS_DBGCMD	0x0D04	W	0x0	Debug Command Register.
DMAC_BUS_DBGINST0	0x0D08	W	0x0	Debug Instruction-0 Register.
DMAC_BUS_DBGINST1	0x0D0C	W	0x0	Debug Instruction-1 Register.
DMAC_BUS_CR0	0x0E00	W		Configuration Register 0.
DMAC_BUS_CR1	0x0E04	W		Configuration Register 1.
DMAC_BUS_CR2	0x0E08	W		Configuration Register 2.
DMAC_BUS_CR3	0x0E0C	W		Configuration Register 3.
DMAC_BUS_CR4	0x0E10	W		Configuration Register 4.
DMAC_BUS_CRDn	0x0E14	W		Configuration Register Dn.

Name	Offset	Size	Reset Value	Description
DMAC_BUS_WD	0x0E80	W	0x0	Watchdog Register.

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

10.4.2 Detail Register Description

DMAC_BUS_DSR
Address:Operational Base+0x0
DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	R	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	R	0x0	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	R	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_BUS_DPC
Address:Operational Base+0x4
DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA manager thread

DMAC_BUS_INTEN
Address:Operational Base+0x20
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	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 interrupt request.

DMAC_BUS_EVENT_RIS
Address:Operational Base+0x24
Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_BUS_INTMIS
Address:Operational Base+0x28
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_BUS_INTCLR
Address:Operational Base+0x2c
Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	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_BUS_FSRD
Address:Operational Base+0x30
Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_BUS_FSRC
Address:Operational Base+0x34
Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_BUS_FTRD
Address:Operational Base+0x38
Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	-	-	reserved
30	R	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

Bit	Attr	Reset Value	Description
			system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	-	-	reserved
16	R	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	-	-	reserved
5	R	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	R	0x0	ndicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.
3:2	-	-	reserved
1	R	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	R	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC_BUS_FTR0~DMAC_BUS_FTR5

Address:Operational Base+0x40
Operational Base+0x44
Operational Base+0x48
Operational Base+0x4c
Operational Base+0x50
Operational Base+0x54

Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	R	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	R	0x0	If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface:

Bit	Attr	Reset Value	Description
			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	-	-	reserved
18	R	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	R	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	R	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	-	-	reserved
13	R	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	R	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	-	-	reserved
7	R	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	R	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

Bit	Attr	Reset Value	Description
			violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: • DMAWFP to wait for a secure peripheral • DMALDP or DMASTP to notify a secure peripheral • DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	R	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	-	-	reserved
1	R	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	R	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_BUS_CSR0~DMAC_BUS_CSR5

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

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21	R	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	-	-	reserved
15	R	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	R	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	-	-	reserved

Bit	Attr	Reset Value	Description
8:4	R	0x0	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	R	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_BUS_CPC0~DMAC_BUS_CPC5

Address:Operational Base+0x104
Operational Base+0x10c
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12c

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA channel n thread

DMAC_BUS_SAR0~DMAC_BUS_SAR5

Address:Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4a0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the source data for DMA channel n

DMAC_BUS_DAR0~DMAC_BUS_DAR5

Address:Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4a4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the Destinationdata for DMA channel n

DMAC_BUS_CCR0~DMAC_BUS_CCR5
Address:Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4a8
Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	-	-	reserved
27:25	R	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	R	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

21:18	R	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	R	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	R	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	R	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data.

Bit	Attr	Reset Value	Description
			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	R	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.

7:4	R	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	R	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	R	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_BUS_LC0_0~DMAC_BUS_LC0_5

Address:Operational Base+0x40c
Operational Base+0x42c
Operational Base+0x44c
Operational Base+0x46c
Operational Base+0x48c
Operational Base+0x4ac

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 0 iterations

DMAC_BUS_LC1_0~DMAC_BUS_LC1_5

Address:Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470
Operational Base+0x490
Operational Base+0x4b0
Loop Counter 1 Registers

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

DMAC_BUS_DBGSTATUS
Address:Operational Base+0xd00
Debug Status Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved

1:0	R	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_BUS_DBGCMD
Address:Operational Base+0xd04
Debug Command Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved

1:0	W	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_BUS_DBGINST0
Address:Operational Base+0xd08
Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 1
23:16	W	0x0	Instruction byte 0
15:11	-	-	reserved

10:8	W	0x0	DMA channel number:
			b000 = DMA channel 0
			b001 = DMA channel 1
			b010 = DMA channel 2
			…
			b111 = DMA channel 7

7:1	-	-	reserved
0	W	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC_BUS_DBGINST1
Address:Operational Base+0xd0c
Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 5
23:16	W	0x0	Instruction byte 4
15:8	W	0x0	Instruction byte 3
7:0	W	0x0	Instruction byte 2

DMAC_BUS_CR0
Address:Operational Base+0xe00
Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	-	-	reserved

21:17	R	0x2	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	R	0x7	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	-	-	reserved

6:4	R	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	-	-	reserved
2	R	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	R	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	R	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_BUS_CR1
Address:Operational Base+0xe04
Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	-	-	reserved

7:4	R	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
			…

Bit	Attr	Reset Value	Description
			b1111 = 16 i-cache lines.
3	-	-	reserved

2:0	R	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_BUS_CR2
Address:Operational Base+0xe08
Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_BUS_CR3
Address:Operational Base+0xe0c
Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_BUS_CR4
Address:Operational Base+0xe10
Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	R	0x6	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_BUS_CRDn
Address:Operational Base+0xe14
DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	-	-	reserved

29:20	R	0x20	The number of lines that the data buffer contains:
			b000000000 = 1 line
			b000000001 = 2 lines
			…
			b111111111 = 1024 lines

19:16	R	0x9	The depth of the read queue:
			b0000 = 1 line
			b0001 = 2 lines
			...
			b1111 = 16 lines.

15	-	-	reserved

Bit	Attr	Reset Value	Description

14:12	R	0x4	Read issuing capability that programs the number of outstanding read transactions:
			b000 = 1
			b001 = 2
			…
			b111 = 8

11:8	R	0x7	The depth of the write queue:
			b0000 = 1 line
			b0001 = 2 lines
			…
			b1111 = 16 lines.

7	-	-	reserved

6:4	R	0x3	Write issuing capability that programs the number of outstanding write transactions:
			b000 = 1
			b001 = 2
			…
			b111 = 8

3	-	-	reserved

2:0		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_BUS_WD
Address:Operational Base+0xe80
DMA Watchdog Register

Bit	Attr	Reset Value	Description
-	-	-	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.

10.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

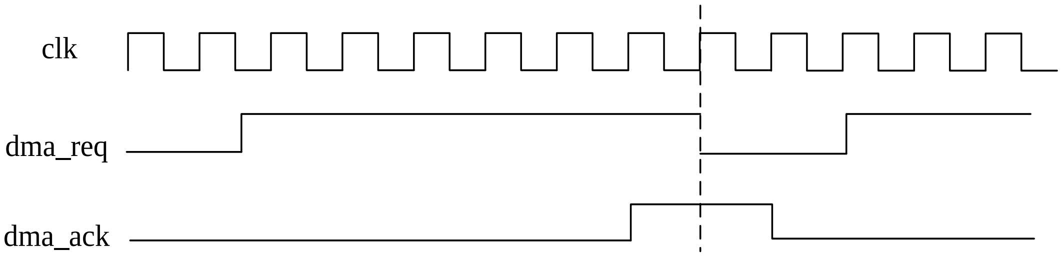


Fig. 10-3 DMAC_BUS request and acknowledge timing

10.6 Interface Description

DMAC_BUS has the following tie-off signals. It can be configured by GRF register or TZPC register. (Please refer to these two chapters to find how to configure)

interface	Reset value	Control source
boot_addr	0x0	Secure GRF
boot_from_pc	0x0	Secure GRF
boot_manager_ns	0x0	Secure GRF / TZPC
boot_irq_ns	0x0	Secure GRF / TZPC
boot_periph_ns	0x0	TZPC

boot_addr
Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.
boot_from_pc
Controls the location in which the DMAC executes its initial instruction, after it exits from reset:
 0 = DMAC waits for an instruction from either APB interface
 1 = DMA manager thread executes the instruction that is located at the address that
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.

10.7 Application Notes

10.7.1 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.

10.7.2 Security usage

When the DMAC exits from reset, the status of the configuration signals that tie-off signals which descripted in chapter 10.6.

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 Registe.
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 Registe.
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.

10.7.3 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 cyclerestrictions

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 ofconcurrently running DMA channel programs with a resource requirement that exceeds theconfigured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

10.7.4 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 split across two lines in the data buffer (see Splitting data, below).
3. There is one idle cycle between the two read data beats .
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data

Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface mneed to be split across two lines in the internal data buffer. This occurs when the read data beat contains datbytes 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 thisdefect.

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

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

10.7.5 Interrupt shares between channel.

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

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

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

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

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

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

10.7.6 Instruction sets

Table 10-2 DMAC Instruction sets

Mnemonic	**Instruction**	**Thread usage:** **• M = DMA manager** **• C = DMA channel**
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	Wai t For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

10.7.7 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 adifferent 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.

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 11 DMA Controller for Peripheral System (DMAC_PERI)

11.1 Overview

DMAC_PERI does not support TrustZone technology and work under non-secure state only.

DMAC_PERI is mainly used for data transfer of the following slaves: HSADC, UART BT, UART BB, UART GPS, UART EXP, SPI0, SPI1,SPI2.

Following table shows the DMAC_PERI request mapping scheme.

Table 11-1 DMAC_PERI Request Mapping Table

Req number	Source	Polarity
0	HSADC/TSI	High level
1	UART BT tx	High level
2	UART BT rx	High level
3	UART BB tx	High level
4	UART BB rx	High level
5	Reserved	
6	Reserved	
7	UART GPS tx	High level
8	UART GPS rx	High level
9	UART EXP tx	High level
10	UART EXP rx	High level
11	SPI0 tx	High level
12	SPI0 rx	High level
13	SPI1 tx	High level
14	SPI1 rx	High level
15	SPI2 tx	High level
16	SPI2 rx	High level
17	Reserved	
18	Reserved	
19	Reserved	

DMAC_PERI supports the following features:

- Supports 20 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 64 MFIFO depth.

11.2 Block Diagram

Figure 11-1 shows the block diagram of DMAC_PERI.

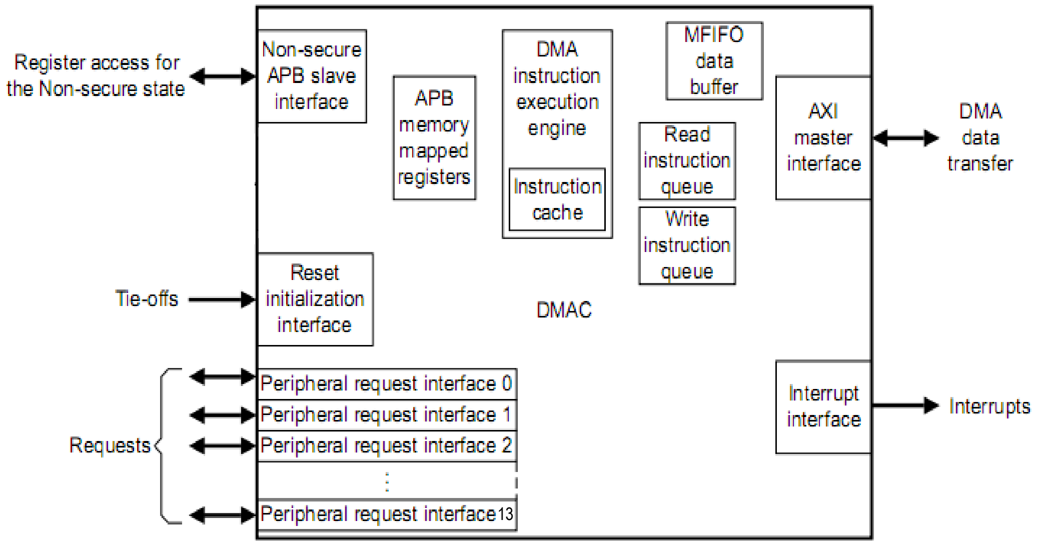

Register access for the Non-secure state
Non-secure APB slave interface
APB memory mapped registers
DMA instruction execution engine
Instruction cache
MFIFO data buffer
Read instruction queue
Write instruction queue
AXI master interface
DMA data transfer
Tie-offs
Reset initialization interface
DMAC
Requests
Peripheral request interface 0
Peripheral request interface 1
Peripheral request interface 2
Peripheral request interface13
Interrupt interface
Interrupts

Fig. 11-1 Block diagram of DMAC_PERI

11.3 Function Description

Please refer to chapter 14.3 for the similar description.

11.4 Register Description

11.4.1 Register summary

Name	Offset	Size	Reset Value	Description
DMAC_PERI_DSR	0x0000	W	0x0	DMA Status Register.
DMAC_PERI_DPC	0x0004	W	0x0	DMA Program Counter Register.
-	-	-	-	reserved
DMAC_PERI_INTEN	0x0020	W	0x0	Interrupt Enable Register
DMAC_PERI_EVENT_RIS	0x0024	W	0x0	Event Status Register.
DMAC_PERI_INTMIS	0x0028	W	0x0	Interrupt Status Register
DMAC_PERI_INTCLR	0x002C	W	0x0	Interrupt Clear Register
DMAC_PERI_FSRD	0x0030	W	0x0	Fault Status DMA Manager Register.
DMAC_PERI_FSRC	0x0034	W	0x0	Fault Status DMA Channel Register.
DMAC_PERI_FTRD	0x0038	W	0x0	Fault Type DMA Manager Register.
-	-	-	-	reserved
DMAC_PERI_FTR0	0x0040	W	0x0	Fault type for DMA Channel 0
DMAC_PERI_FTR1	0x0044	W	0x0	Fault type for DMA Channel 1
DMAC_PERI_FTR2	0x0048	W	0x0	Fault type for DMA Channel 2
DMAC_PERI_FTR3	0x004C	W	0x0	Fault type for DMA Channel 3
DMAC_PERI_FTR4	0x0050	W	0x0	Fault type for DMA Channel 4
DMAC_PERI_FTR5	0x0054	W	0x0	Fault type for DMA Channel 5
DMAC_PERI_FTR6	0x0058	W	0x0	Fault type for DMA Channel 6
-	-	-	-	reserved
DMAC_PERI_CSR0	0x0100	W	0x0	Channel Status for DMA Channel 0
DMAC_PERI_CSR1	0x0108	W	0x0	Channel Status for DMA Channel 1
DMAC_PERI_CSR2	0x0110	W	0x0	Channel Status for DMA Channel 2
DMAC_PERI_CSR3	0x0118	W	0x0	Channel Status for DMA Channel 3
DMAC_PERI_CSR4	0x0120	W	0x0	Channel Status for DMA Channel 4
DMAC_PERI_CSR5	0x0128	W	0x0	Channel Status for DMA Channel 5

Name	Offset	Size	Reset Value	Description
DMAC_PERI_CSR6	0x0130	W	0x0	Channel Status for DMA Channel 6
DMAC_PERI_CPC0	0x0104	W	0x0	Channel PC for DMA Channel 0
DMAC_PERI_CPC1	0x010c	W	0x0	Channel PC for DMA Channel 1
DMAC_PERI_CPC2	0x0114	W	0x0	Channel PC for DMA Channel 2
DMAC_PERI_CPC3	0x011c	W	0x0	Channel PC for DMA Channel 3
DMAC_PERI_CPC4	0x0124	W	0x0	Channel PC for DMA Channel 4
DMAC_PERI_CPC5	0x012c	W	0x0	Channel PC for DMA Channel 5
DMAC_PERI_CPC6	0x0134	W	0x0	Channel PC for DMA Channel 6
DMAC_PERI_SAR0	0x0400	W	0x0	Source Address for DMA Channel 0
DMAC_PERI_SAR1	0x0420	W	0x0	Source Address for DMA Channel 1
DMAC_PERI_SAR2	0x0440	W	0x0	Source Address for DMA Channel 2
DMAC_PERI_SAR3	0x0460	W	0x0	Source Address for DMA Channel 3
DMAC_PERI_SAR4	0x0480	W	0x0	Source Address for DMA Channel 4
DMAC_PERI_SAR5	0x04a0	W	0x0	Source Address for DMA Channel 5
DMAC_PERI_SAR6	0x04c0	W	0x0	Source Address for DMA Channel 6
DMAC_PERI_DAR0	0x0404	W	0x0	Dest Address for DMAChannel 0
DMAC_PERI_DAR1	0x0424	W	0x0	Dest Address for DMAChannel 1
DMAC_PERI_DAR2	0x0444	W	0x0	Dest Address for DMAChannel 2
DMAC_PERI_DAR3	0x0464	W	0x0	Dest Address for DMAChannel 3
DMAC_PERI_DAR4	0x0484	W	0x0	Dest Address for DMAChannel 4
DMAC_PERI_DAR5	0x04a4	W	0x0	Dest Address for DMAChannel 5
DMAC_PERI_DAR6	0x04c4	W	0x0	Dest Address for DMAChannel 6
DMAC_PERI_CCR0	0x0408	W	0x0	Channel Control for DMA Channel 0
DMAC_PERI_CCR1	0x0428	W	0x0	Channel Control for DMA Channel 1
DMAC_PERI_CCR2	0x0448	W	0x0	Channel Control for DMA Channel 2
DMAC_PERI_CCR3	0x0468	W	0x0	Channel Control for DMA Channel 3
DMAC_PERI_CCR4	0x0488	W	0x0	Channel Control for DMA Channel 4
DMAC_PERI_CCR5	0x04a8	W	0x0	Channel Control for DMA Channel 5
DMAC_PERI_CCR6	0x04c8	W	0x0	Channel Control for DMA Channel 6
DMAC_PERI_LC0_0	0x040C	W	0x0	Loop Counter 0 for DMA Channel 0
DMAC_PERI_LC0_1	0x042C	W	0x0	Loop Counter 0 for DMA Channel 1
DMAC_PERI_LC0_2	0x044C	W	0x0	Loop Counter 0 for DMA Channel 2
DMAC_PERI_LC0_3	0x046C	W	0x0	Loop Counter 0 for DMA Channel 3
DMAC_PERI_LC0_4	0x048C	W	0x0	Loop Counter 0 for DMA Channel 4
DMAC_PERI_LC0_5	0x04aC	W	0x0	Loop Counter 0 for DMA Channel 5
DMAC_PERI_LC0_6	0x04cC	W	0x0	Loop Counter 0 for DMA Channel 6
DMAC_PERI_LC1_0	0x0410	W	0x0	Loop Counter 1 for DMA Channel 0
DMAC_PERI_LC1_1	0x0430	W	0x0	Loop Counter 1 for DMA Channel 1
DMAC_PERI_LC1_2	0x0450	W	0x0	Loop Counter 1 for DMA Channel 2
DMAC_PERI_LC1_3	0x0470	W	0x0	Loop Counter 1 for DMA Channel 3
DMAC_PERI_LC1_4	0x0490	W	0x0	Loop Counter 1 for DMA Channel 4
DMAC_PERI_LC1_5	0x04b0	W	0x0	Loop Counter 1 for DMA Channel 5
DMAC_PERI_LC1_6	0x04d0	W	0x0	Loop Counter 1 for DMA Channel 6
-	-	-	-	reserved
DMAC_PERI_DBGST DMAC_PERI_ATUS	0x0D00	W	0x0	Debug Status Register.
DMAC_PERI_DBGCMD	0x0D04	W	0x0	Debug Command Register.
DMAC_PERI_DBGINST0	0x0D08	W	0x0	Debug Instruction-0 Register.
DMAC_PERI_DBGINST1	0x0D0C	W	0x0	Debug Instruction-1 Register.
DMAC_PERI_CR0	0x0E00	W		Configuration Register 0.
DMAC_PERI_CR1	0x0E04	W		Configuration Register 1.
DMAC_PERI_CR2	0x0E08	W		Configuration Register 2.
DMAC_PERI_CR3	0x0E0C	W		Configuration Register 3.

Name	Offset	Size	Reset Value	Description
DMAC_PERI_CR4	0x0E10	W		Configuration Register 4.
DMAC_PERI_CRDn	0x0E14	W		Configuration Register Dn.
DMAC_PERI_WD	0X0E80	W		Watchdog Register

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

11.4.2 Detail Register Description

DMAC_PERI_DSR
Address:Operational Base+0x0
DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	R	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	R	0x0	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	R	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_PERI_DPC
Address:Operational Base+0x4
DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA manager thread

DMAC_PERI_INTEN
Address:Operational Base+0x20
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	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 interrupt request.

DMAC_PERI_EVENT_RIS
Address:Operational Base+0x24
Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_PERI_INTMIS
Address:Operational Base+0x28
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_PERI_INTCLR
Address:Operational Base+0x2c
Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	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_PERI_FSRD
Address:Operational Base+0x30
Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_PERI_FSRC
Address:Operational Base+0x34
Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_PERI_FTRD
Address:Operational Base+0x38
Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	-	-	reserved
30	R	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

Bit	Attr	Reset Value	Description
			system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	-	-	reserved
16	R	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	-	-	reserved
5	R	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	R	0x0	ndicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.
3:2	-	-	reserved
1	R	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	R	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC_PERI_FTR0~DMAC_PERI_FTR6
Address:Operational Base+0x40
Operational Base+0x44
Operational Base+0x48
Operational Base+0x4c
Operational Base+0x50
Operational Base+0x54
Operational Base+0x58
Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	R	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	R	0x0	If the DMA channel aborts, this bit indicates if the

Bit	Attr	Reset Value	Description
			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	-	-	reserved
18	R	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	R	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	R	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	-	-	reserved
13	R	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	R	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	-	-	reserved
7	R	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	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP,

Bit	Attr	Reset Value	Description
			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: • DMAWFP to wait for a secure peripheral • DMALDP or DMASTP to notify a secure peripheral • DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	R	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	-	-	reserved
1	R	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	R	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_PERI_CSR0~DMAC_PERI_CSR6

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

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21	R	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	-	-	reserved
15	R	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	R	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	-	-	reserved
8:4	R	0x0	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	R	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_PERI_CPC0~DMAC_PERI_CPC6

Address:Operational Base+0x104
Operational Base+0x10c
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12c
Operational Base+0x134

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA channel n thread

DMAC_PERI_SAR0~DMAC_PERI_SAR6

Address:Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4a0
Operational Base+0x4c0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the source data for DMA channel n

DMAC_PERI_DAR0~DMAC_PERI_DAR5

Address:Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4a4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the Destinationdata for DMA channel n

DMAC_PERI_CCR0~DMAC_PERI_CCR6

Address:Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4a8
Operational Base+0x4c8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	-	-	reserved
27:25	R	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	R	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

21:18	R	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	R	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	R	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.

Bit	Attr	Reset Value	Description
			1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	R	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	R	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.

7:4	R	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	R	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	R	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_PERI_LC0_0~DMAC_PERI_LC0_6

Address:Operational Base+0x40c
Operational Base+0x42c
Operational Base+0x44c
Operational Base+0x46c
Operational Base+0x48c
Operational Base+0x4ac
Operational Base+0x4cc

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 0 iterations

DMAC_PERI_LC1_0~DMAC_PERI_LC1_6
Address:Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470
Operational Base+0x490
Operational Base+0x4b0
Operational Base+0x4e0

Loop Counter 1 Registers

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

DMAC_PERI_DBGSTATUS
Address:Operational Base+0xd00
Debug Status Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved

1:0	R	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_PERI_DBGCMD
Address:Operational Base+0xd04
Debug Command Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved

1:0	W	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_PERI_DBGINST0
Address:Operational Base+0xd08
Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 1
23:16	W	0x0	Instruction byte 0
17:11	-	-	reserved

10:8	W	0x0	DMA channel number:
			b000 = DMA channel 0
			b001 = DMA channel 1
			b010 = DMA channel 2
			…
			b111 = DMA channel 7

Bit	Attr	Reset Value	Description
7:1	-	-	reserved
0	W	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC_PERI_DBGINST1
Address:Operational Base+0xd0c
Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 5
23:16	W	0x0	Instruction byte 4
15:8	W	0x0	Instruction byte 3
7:0	W	0x0	Instruction byte 2

DMAC_PERI_CR0
Address:Operational Base+0xe00
Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	-	-	reserved

21:17	R	0x2	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	R	0x7	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	-	-	reserved

6:4	R	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	-	-	reserved
2	R	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	R	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	R	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_PERI_CR1

Address:Operational Base+0xe04
Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	-	-	reserved

7:4	R	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	-	-	reserved

2:0	R	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_PERI_CR2
Address:Operational Base+0xe08
Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC_PERI_CR3
Address:Operational Base+0xe0c
Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	R	0x0	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_PERI_CR4
Address:Operational Base+0xe10
Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	R	0x6	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_PERI_CRDn
Address:Operational Base+0xe14
DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	-	-	reserved

29:20	R	0x20	The number of lines that the data buffer contains:
			b000000000 = 1 line
			b000000001 = 2 lines

Bit	Attr	Reset Value	Description
			… b111111111 = 1024 lines

19:16	R	0x9	The depth of the read queue:
			b0000 = 1 line
			b0001 = 2 lines
			...
			b1111 = 16 lines.

15	-	-	reserved

14:12	R	0x4	Read issuing capability that programs the number of outstanding read transactions:
			b000 = 1
			b001 = 2
			…
			b111 = 8

11:8	R	0x7	The depth of the write queue:
			b0000 = 1 line
			b0001 = 2 lines
			…
			b1111 = 16 lines.

7	-	-	reserved

6:4	R	0x3	Write issuing capability that programs the number of outstanding write transactions:
			b000 = 1
			b001 = 2
			…
			b111 = 8

3	-	-	reserved

2:0		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_PERI_WD
Address:Operational Base+0xe80
DMA Watchdog Register

Bit	Attr	Reset Value	Description
31:1	-	-	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.

11.5 Timing Diagram

Please refer to chapter 14.5 for the similar description.

11.6 Interface Description

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

DMAC_PERI

interface	**Reset value**	**Control source**
boot_addr	0x0	GRF
boot_from_pc	0x0	GRF
boot_manager_ns	0x0	GRF
boot_irq_ns	0xf	GRF
boot_periph_ns	0xfffff	GRF

boot_addr
Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

boot_from_pc
Controls the location in which the DMAC executes its initial instruction, after it exits from reset:
  0 = DMAC waits for an instruction from either APB interface
  1 = DMA manager thread executes the instruction that is located at the address that

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.

11.7 Application Notes

Please refer to chapter 14.7 for the similar description.

Chapter 12 Temperature Sensor ADC(TSADC)

12.1 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 interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 4 channel TS-ADC, the temperature criteria of each channel 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 debounce can be configurable

12.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

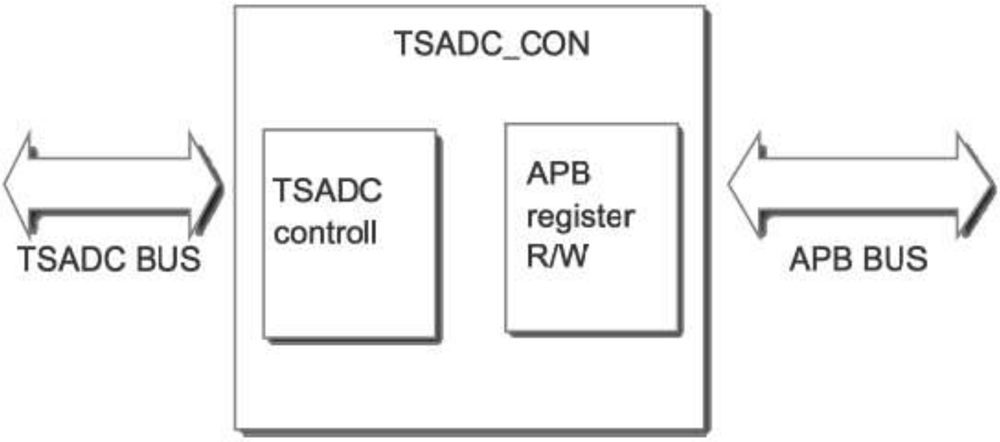


Fig. 12-1 TS-ADC Controller Block Diagram

12.3 Function Description

12.3.1 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.

12.3.2 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.

12.4 Register Description

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000208	The control register of A/D Converter.
TSADC_AUTO_CON	0x0004	W	0x00000000	TSADC auto mode control register
TSADC_INT_EN	0x0008	W	0x00000000	Interrupt enable
TSADC_INT_PD	0x000c	W	0x00000000	Interrupt Status
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_DATA2	0x0028	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_DATA3	0x002c	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_COMP2_INT	0x0038	W	0x00000000	TSADC high temperature level for source 2
TSADC_COMP3_INT	0x003c	W	0x00000000	TSADC high temperature level for source 3
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_COMP2_SHUT	0x0048	W	0x00000000	TSADC high temperature level for source 2
TSADC_COMP3_SHUT	0x004c	W	0x00000000	TSADC high temperature level for source 3
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

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

12.4.2 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.
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	0x1	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]). 111 : Input source 0 (SARADC_AIN[0]) 110 : Input source 1 (SARADC_AIN[1]) 101 : Input source 2 (SARADC_AIN[2]) 100 : Input source 3 (SARADC_AIN[3]) Others : Reserved

TSADC_AUTO_CON
Address: Operational Base + offset (0x0004)
TSADC auto mode control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	last_tshut 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

Bit	Attr	Reset Value	Description
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.
15:9	RO	0x0	reserved
8	RW	0x0	tshut polarity 0: low active 1: high active
7	RW	0x0	src3_en 0: do not care the temperature of source 3 1: if the temperature of source 3 is too high , TSHUT will be valid
6	RW	0x0	src2_en 0: do not care the temperature of source 2 1: if the temperature of source 2 is too high , TSHUT will be valid
5	RW	0x0	src1_en 0: do not care the temperature of source 1 1: if the temperature of source 1 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
3:1	RO	0x0	reserved
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
31:13	RO	0x0	reserved
12	RW	0x0	eoc_int_en eoc_Interrupt enable. eoc_interrupt enable in user defined mode 0: Disable; 1: Enable
11	RW	0x0	tshut_2cru_en_src3 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.

Bit	Attr	Reset Value	Description
10	RW	0x0	tshut_2cru_en_src2 0: TSHUT output to cru disabled. TSHUT output will always keep low . 1: TSHUT output works.
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	RW	0x0	tshut_2gpio_en_src3 0: TSHUT output to gpio0b2 disabled. TSHUT output will always keep low . 1: TSHUT output works.
6	RW	0x0	tshut_2gpio_en_src2 0: TSHUT output to gpio0b2 disabled. TSHUT output will always keep low . 1: TSHUT output works.
5	RW	0x0	tshut_2gpio_en_src1 0: TSHUT output to gpio0b2 disabled. TSHUT output will always keep low . 1: TSHUT output works.
4	RW	0x0	tshut_2gpio_en_src0 0: TSHUT output to gpio0b2 disabled. TSHUT output will always keep low . 1: TSHUT output works.
3	RW	0x0	ht_inten_src3 high temperature interrupt enable for src3 0: disable 1: enable
2	RW	0x0	ht_inten_src2 high temperature interrupt enable for src2 0: disable 1: enable
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
31:9	RO	0x0	reserved
8	RW	0x0	eoc_int_pd Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
7	RW	0x0	tshut_o_src3 TSHUT output status
6	RW	0x0	tshut_o_src2 TSHUT output status
5	RW	0x0	tshut_o_src1 TSHUT output status
4	RW	0x0	tshut_o_src0 TSHUT output status
3	RW	0x0	ht_irq_src3 When TSADC output is smaller 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.
2	RW	0x0	ht_irq_src2 When TSADC output is smaller 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.
1	RW	0x0	ht_irq_src1 When TSADC output is smaller 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 smaller 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

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 0 last conversion (DOUT[9:0]).

TSADC_DATA1
Address: Operational Base + offset (0x0024)
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 1 last conversion (DOUT[9:0]).

TSADC_DATA2
Address: Operational Base + offset (0x0028)
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 2 last conversion (DOUT[9:0]).

TSADC_DATA3
Address: Operational Base + offset (0x002c)
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 3 last conversion (DOUT[9:0]).

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 smaller 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

Bit	Attr	Reset Value	Description
11:0	RW	0x000	tsadc_comp_src1 TSADC high temperature level. TSADC output is smaller than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP2_INT
Address: Operational Base + offset (0x0038)
TSADC high temperature level for source 2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src2 TSADC high temperature level. TSADC output is smaller than tsadc_comp, means the temperature is high. TSADC_INT will be valid.

TSADC_COMP3_INT
Address: Operational Base + offset (0x003c)
TSADC high temperature level for source 3

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src3 TSADC high temperature level. TSADC output is smaller 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 smaller 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

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

TSADC_COMP2_SHUT
Address: Operational Base + offset (0x0048)
TSADC high temperature level for source 2

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src2 TSADC high temperature level. TSADC output is smaller than tsadc_comp, means the temperature is too high. TSHUT will be valid.

TSADC_COMP3_SHUT
Address: Operational Base + offset (0x004c)
TSADC high temperature level for source 3

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src3 TSADC high temperature level. TSADC output is smaller 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

12.5 Application Notes

12.5.1 Channel Select

The system has three Temperature Sensors, channel0 is reserve, and channel 1is for CPU, and channel 2 is for GPU.

12.5.2 Single-sample conversion

To saving power, the TS-ADC used single-sample conversion. The timing as flowing picture:

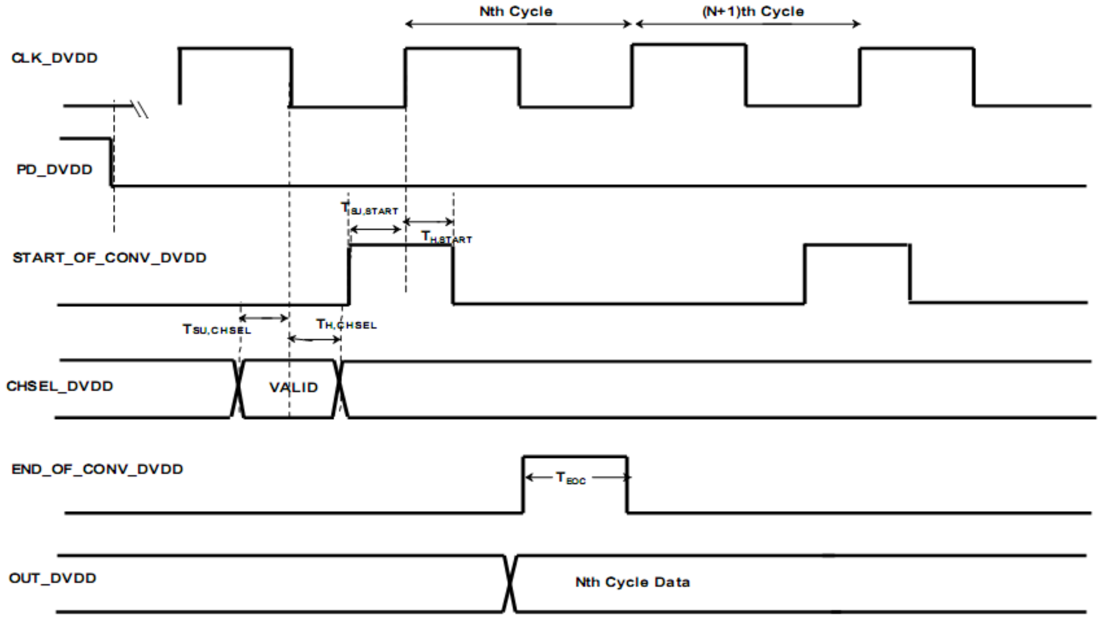


Fig. 12-2 Single-sample conversion

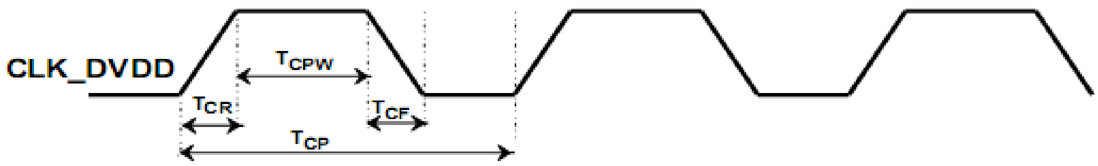


Fig. 12-3 Clock Timing Diagram

The below timing constraints are applicable for single-sample conversion mode.

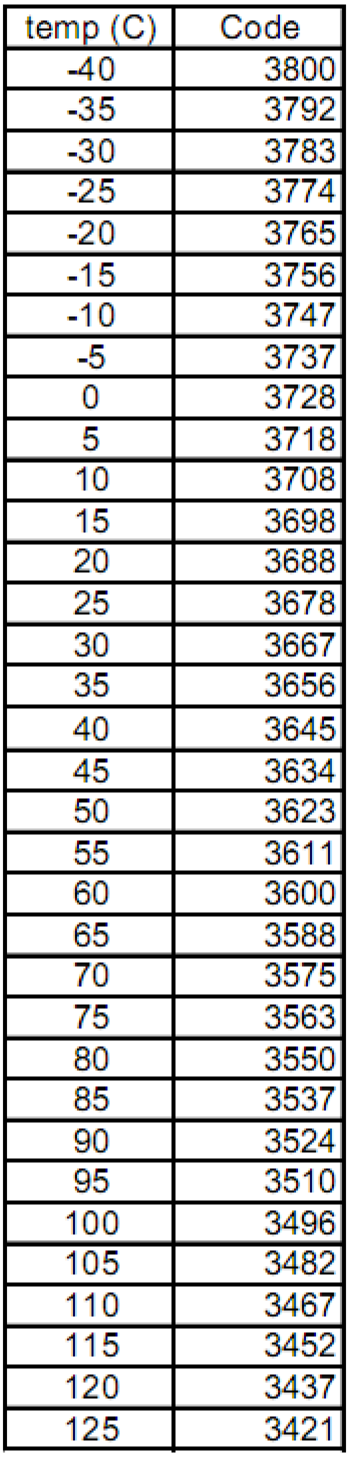
Table 12-1 Timing parameters

Timing	Symbol	Value			Unit	Description
		Min	Typ	Max		
START_OF_CONV Setup time	TSU,START	5			ns	Set Up time for START_OF_CONV w.r.t CLKIN rising edge
START_OF_CONV Hold time	TH,START	5			ns	Hold time for START_OF_CONV w.r.t CLKIN rising edge
CHSEL setup time	TSU,CHSEL	5			ns	Set Up time for CHSEL w.r.t CLKIN falling edge
CHSEL Hold time	TH,CHSEL	5			ns	Hold time for CHSEL w.r.t CLKIN falling edge
Data Setup	TSU,DATA	11		15	us	Set Up time for output data w.r.t either CLKIN rising edge or END_OF_CONV falling edge
Data Hold	TH,DATA	5		9	us	Hold time for output data w.r.t either CLKIN rising edge or END_OF_CONV falling edge
Data access time	TDAC	5		9	us	Valid data w.r.t CLKIN rising edge
Delay time	TDelay			5	ns	Delay between Valid data and EOC_DVDD rising edge
EOC Pulse Width (max frequency)	TEOC	11		15	us	Pulse width of EOC
CLKIN Rise Time	TCR			2	ns	CLKIN Rise Time
CLKIN Fall Time	TCF			2	ns	CLKIN Fall Time
CLK Pulse Width(Duty Cycle)	TCPW	45		55	%	CLKIN High/Low Time Period
CLK Period	TCP	20			us	CLKIN Time Period

Note: The time from negedge of PD_DVDD to posedge of START_OF_CONV_DVDD is more than 8 cycles of CLK_DVDD, and less than 500us at same time.

12.5.3 Temperature-to-code mapping

Table 12-2 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.

12.5.4 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).

12.5.5 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=0,1,2,3), 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=0,1,2,3), 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 13 eFuse

13.1 Overview

In RK3288, there are two eFuse. One is organized as 32bits by 8 one-time programmable electrical fuses with random access interface, and the other is organized as 32bits by 32 one-time programmable electrical fuses.
The 32x32 eFuse can only be accessed by APB bus when IO_SECURITYsel is high.
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

13.2 Block Diagram

In the following diagram, all the signals except power supply VDD_EFUSE and VQPS_EFUSE are controlled by registers. For detailed description, please refer to detailed register descriptions.

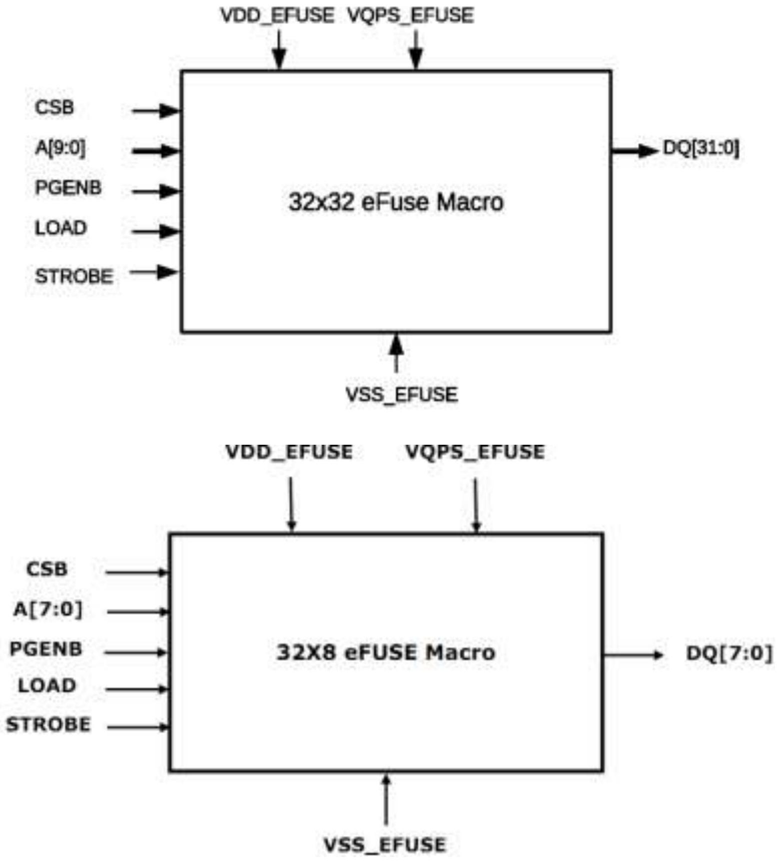


Fig. 13-1 RK3288 eFuse block diagram

13.3 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.

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
EFUSE_CTRL	0x0000	W	0x00000000	efuse control register
EFUSE_DOUT	0x0004	W	0x00000000	efuse data out 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

EFUSE_CTRL

Address: Operational Base + offset (0x0000)

eFuse control register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
15:6	RW	0x00	efuse_addr efuse address pins : A[7:0] / A[9:0]
5:4	RO	0x0	reserved
3	RW	0x0	efuse_pgenb efuse program enable (active low) :PGENB
2	RW	0x0	efuse_load efuse turn on sense amplifier and load data into latch (active high) : LOAD
1	RW	0x0	efuse_strobe efuse turn on the array for read or program access (active high) : STROBE
0	RW	0x0	efuse_csb efuse chip select enable signal, active low : CSB

EFUSE_DOUT

Address: Operational Base + offset (0x0004)

eFuse data out register

Bit	Attr	Reset Value	Description
31:0	RO	0x00	efuse_dout efuse data output

13.5 Timing Diagram

When efuse is in program(PGM) mode.

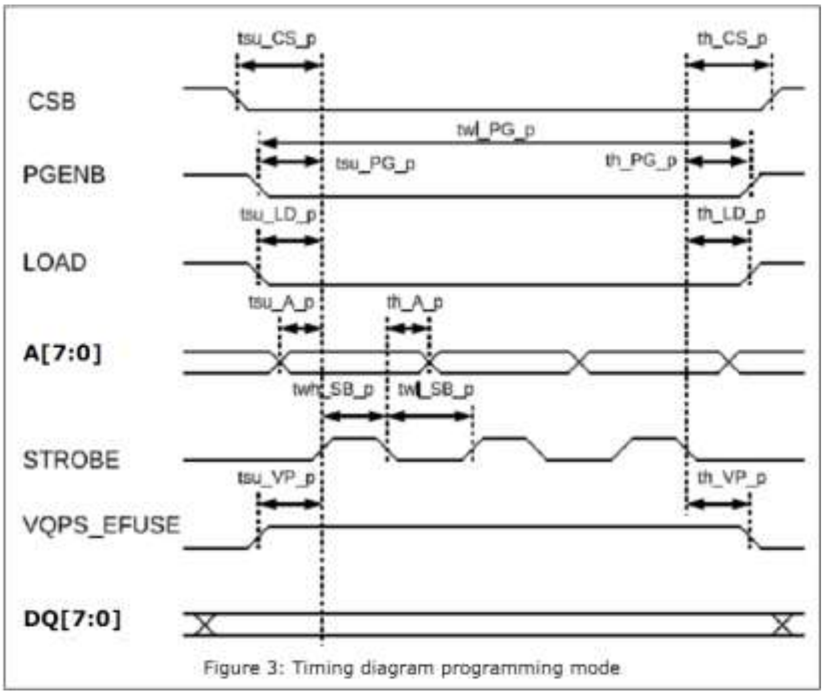


Fig. 13-2 RK3288 efuse timing diagram in program mode

When efuse is in read mode.

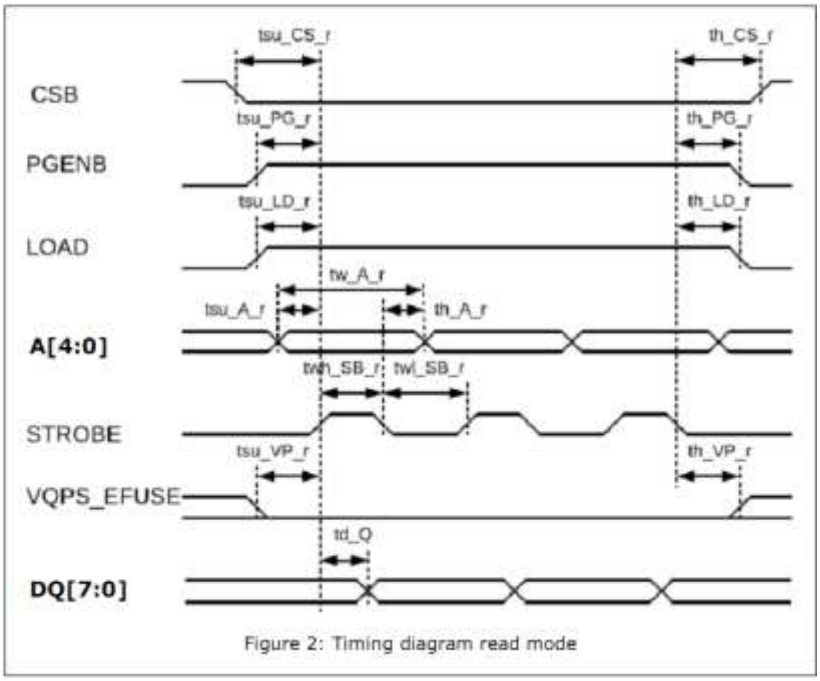
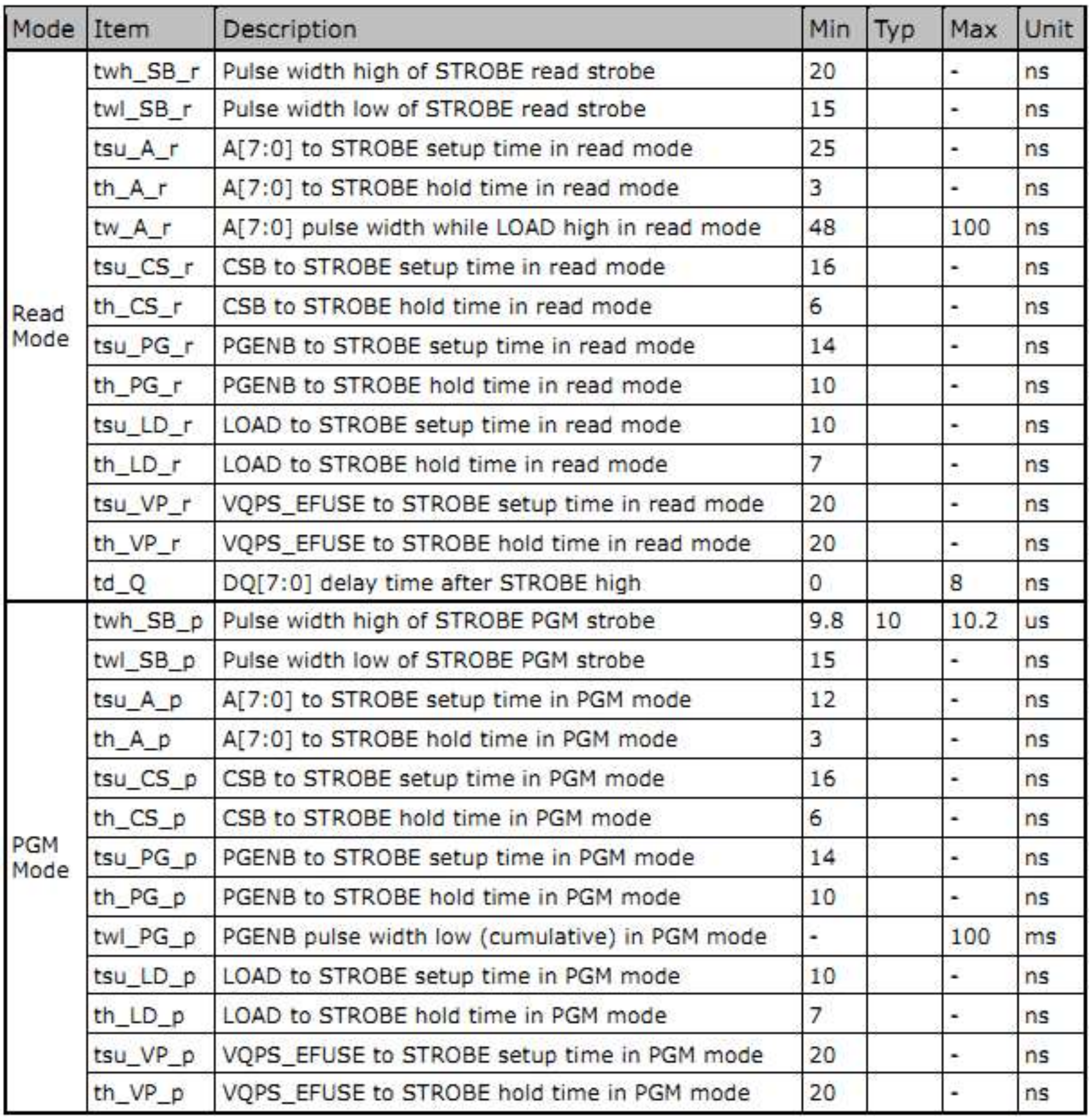


Fig. 13-3 RK3288 efuse timing diagram in read mode

The following table has shows the detailed value for timing parameters in the above diagram.

Table 13-1 RK3288 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[7:0] to STROBE setup time in read mode	25		-	ns
	th_A_r	A[7:0] to STROBE hold time in read mode	3		-	ns
	tw_A_r	A[7:0] 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[7:0] 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[7:0] to STROBE setup time in PGM mode	12		-	ns
	th_A_p	A[7:0] 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

13.6 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/Q31 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.

Chapter 14 WatchDog(WDT)

14.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may be caused by conflicting parts or programs in a SoC.The WDT would generated 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

14.2 Block Diagram

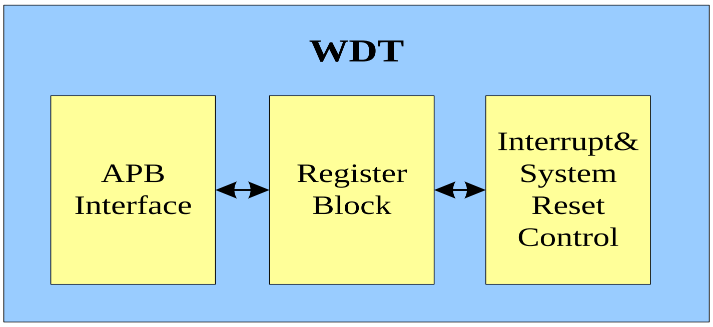


Fig. 14-1 WDT block diagram

Block Descriptions:

APB Interface

The APB Interface implements the APB slave operation. It's 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 comprising of a decrementing counter and control logic.

14.3 Function description

14.3.1 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

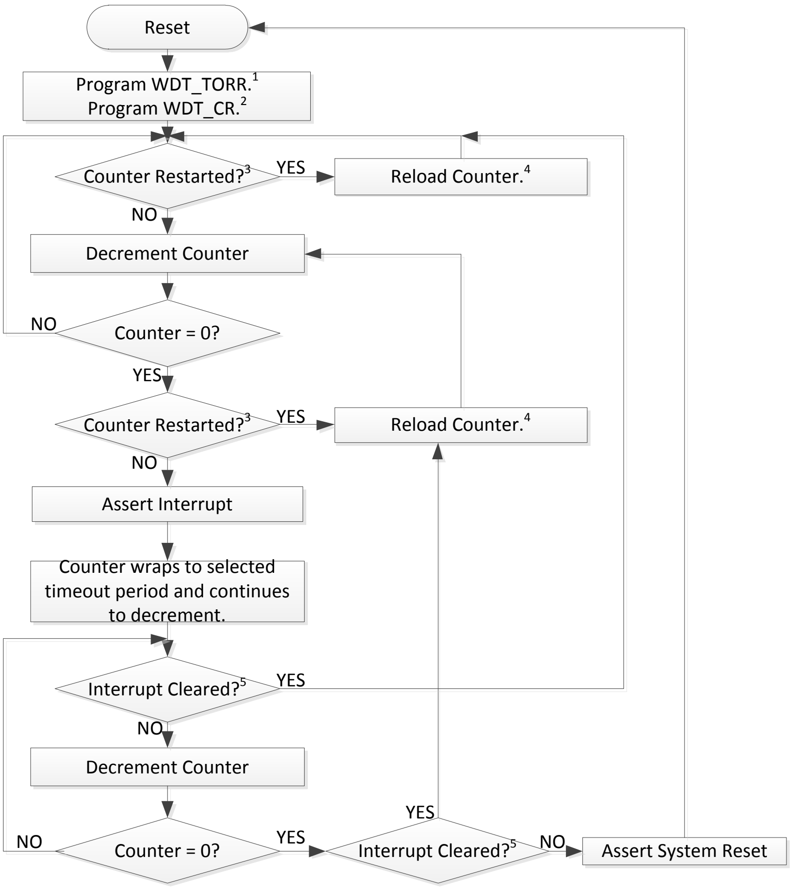
When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

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.

14.3.2 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. 14-2 WDT Operation Flow

14.4 Register Description

This section describes the control/status registers of the design.

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register

Name	Offset	Size	Reset Value	Description
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*

14.4.2 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. 3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 1'b0: Generate a system reset 1'b1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset
0	RW	0x0	wdt_en WDT enable 1'b0: WDT disabled 1'b1: WDT enabled

WDT_TORR
Address: Operational Base + offset (0x0004)
Timeout range Register

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

Bit	Attr	Reset Value	Description
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: 4’b0000: 0x0000ffff 4’b0001: 0x0001ffff 4’b0010: 0x0003ffff 4’b0011: 0x0007ffff 4’b0100: 0x000fffff 4’b0101: 0x001fffff 4’b0110: 0x003fffff 4’b0111: 0x007fffff 4’b1000: 0x00ffffff 4’b1001: 0x01ffffff 4’b1010: 0x03ffffff 4’b1011: 0x07ffffff 4’b1100: 0x0fffffff 4’b1101: 0x1fffffff 4’b1110: 0x3fffffff 4’b1111: 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 when ever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)
Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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'b1: Interrupt is active regardless of polarity. 1'b0: Interrupt is inactive.

WDT_EOI
Address: Operational Base + offset (0x0014)
Interrupt clear Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

14.5 Application Notes

Please refer to the function description section.

Chapter 15 Pulse Width Modulation (PWM)

15.1 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
- 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 do 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.

15.2 Block Diagram

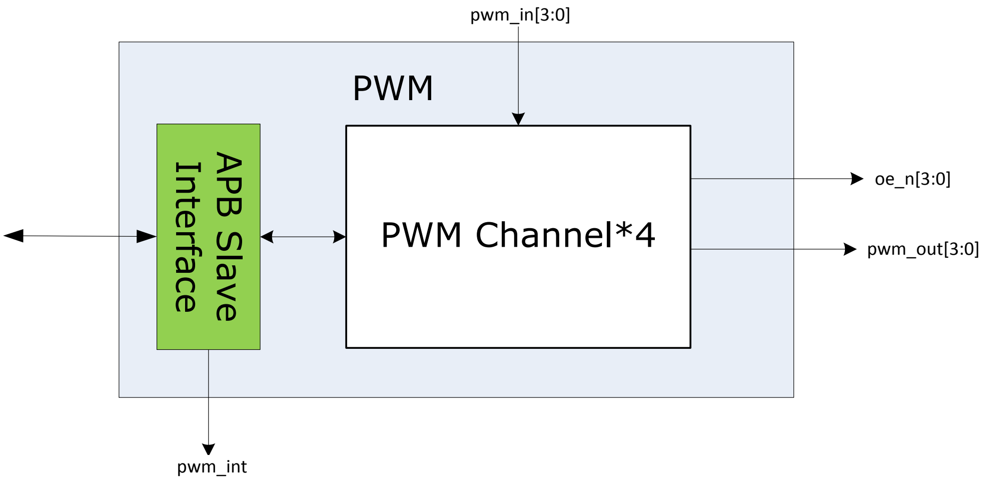


Fig. 15-1 PWM architecture

15.2.1 PWM APB Slave Interface

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.

15.2.2 PWM Channels

This is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

15.3 Functional description

The PWM supports three operation modes: reference 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.

15.3.1 Reference mode

The reference 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.
Note: 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.

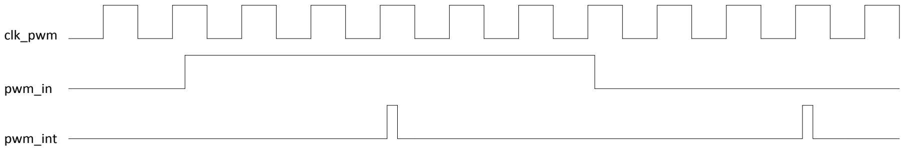


Fig. 15-2 PWM Reference Mode

15.3.2 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.

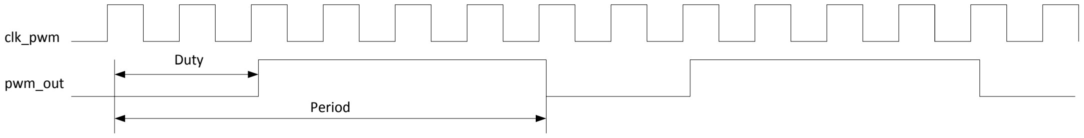


Fig. 15-3 PWM Left-aligned Output Mode

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.

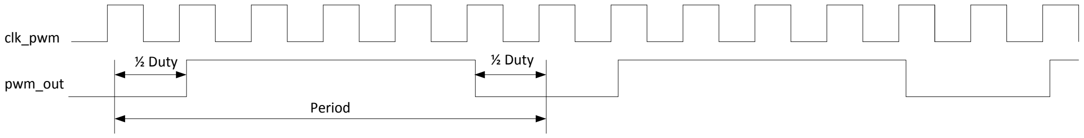


Fig. 15-4 PWM Center-aligned Output Mode

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).

15.3.3 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.

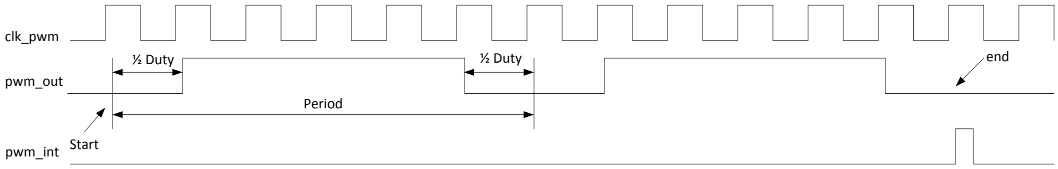


Fig. 15-5 PWM Center-aligned Output Mode

15.4 Register Description

15.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
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

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

15.4.2 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_LPR 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

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 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

Bit	Attr	Reset Value	Description
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.
7:6	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked 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_LPR 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

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 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

Bit	Attr	Reset Value	Description
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.
7:6	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked 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_LPR 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

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

Bit	Attr	Reset Value	Description
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.
7:6	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked 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_LPR 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

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 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

Bit	Attr	Reset Value	Description
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.
7:6	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en PWM channel enable 0: disabled 1: enabled. If the PWM is worked 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 0 input waveform. Otherwise, please refer to PWM3_PERIOD_HPR 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 0 input waveform. Otherwise, please refer to PWM2_PERIOD_HPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.
9	RO	0x0	CH1_Pol Channel 1 Interrupt Polarity Flag This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM1_PERIOD_HPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.

Bit	Attr	Reset Value	Description
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_HPR 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	RW	0x0	CH3_IntSts Channel 3 Interrupt Status 0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated
2	RW	0x0	CH2_IntSts Channel 2 Interrupt Status 0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated
1	RW	0x0	CH1_IntSts Channel 1 Interrupt Status 0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated
0	RW	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
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

Bit	Attr	Reset Value	Description
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

15.5 Interface Description

Table 15-1 PWM Interface Description

Module pin	Direction	Pad name	IOMUX
pwm3	O	GPIO7_C[7]	GPIO7CH_IOMUX[14:12]=011
pwm2	O	GPIO7_C[6]	GPIO7CH_IOMUX[9:8]=11
pwm1	O	GPIO7_A[1]	GPIO7A_IOMUX[2]=1
pwm0	O	GPIO7_A[0]	GPIO7A_IOMUX[1:0]=01

15.6 Application Notes

15.6.1 PWM Reference 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 reference 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.

15.6.2 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, write ‘0’ to PWMx_CTRL.pwm_en bit to disable the PWM channel.

15.6.3 Low-power mode

Setting PWMx_CTRL.lp_en to ‘1’ makes the channel enter the low-power mode. When then 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.

15.6.4 Other notes

When the channel is active to produce waveforms, it is free to programming 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.
In RK3288, user need to configure the SOC_CON2[0] to "1'b1" before using the PWM.

Chapter 16 UART Interface

16.1 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:

- AMBA APB interface – Allows for easy integration into a Synthesizable Components for AMBA 2 implementation
- Support interrupt interface to interrupt controller
- Contain two 64Bytes FIFOs for data receive and transmit
- Programmable serial data baud rate as calculated by the following: baud rate = (serial clock frequency)/(16×divisor)
- UART_BB/UART_BT/UART_GPS/UART_EXP support auto flow-control, UART_DBG do not support auto flow-control
- UART_DBG support IrDA 1.0 SIR mode with up to 115.2 Kbaud data rate
- UART_BB/UART_BT/UART_GPS/UART_EXP are in peripheral subsystem, UART_DBG is in bus subsystem

16.2 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

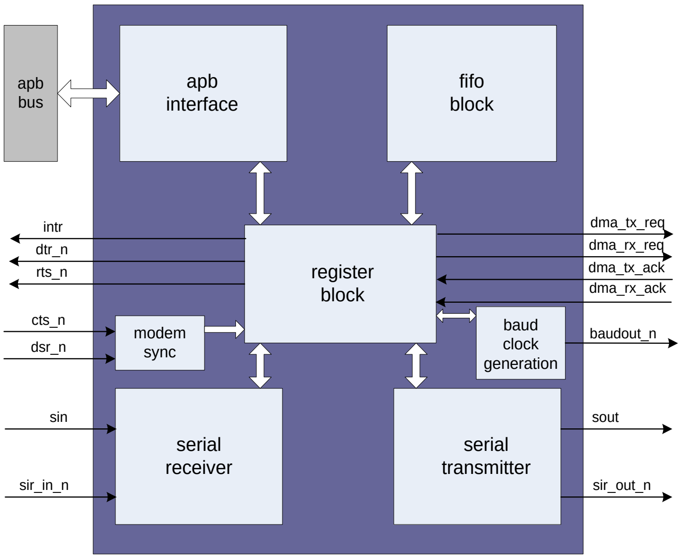


Fig. 16-1 UART Architecture

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
Generate the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).

Serial Transmitter
Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. 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.

16.3 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.

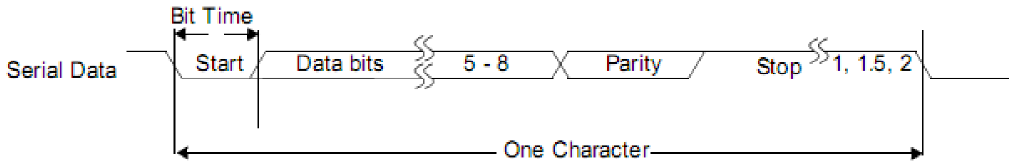


Fig. 16-2 UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications 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.

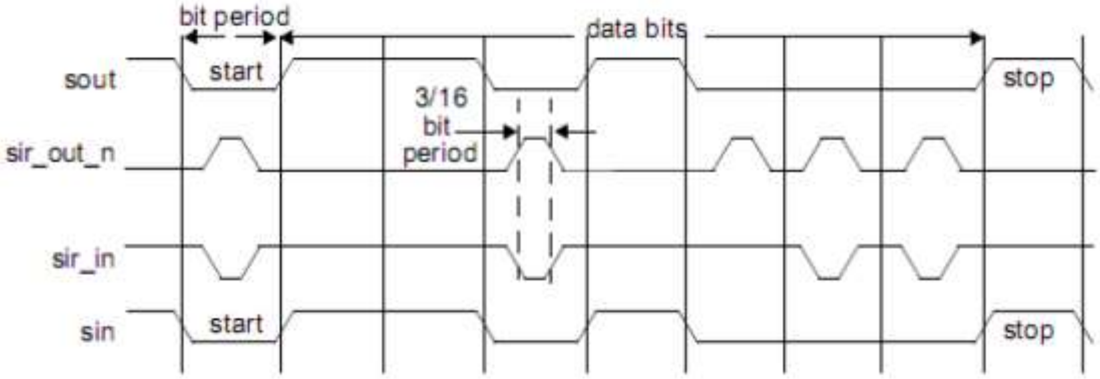


Fig. 16-3 IrDA 1.0

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.

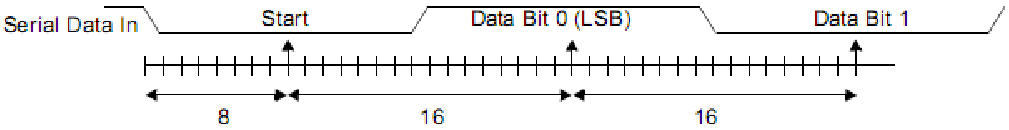


Fig. 16-4 UART baud rate

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 UART1/UART2/UART3 is 32bytes and the FIFO depth of UART0 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The uart supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.
The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

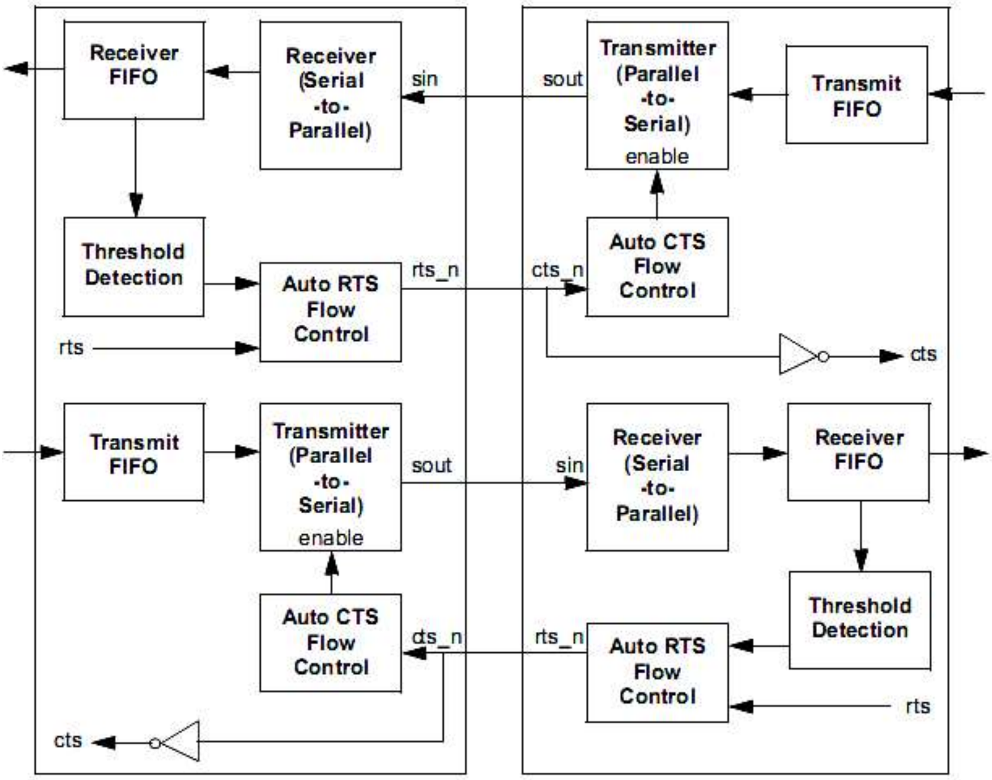


Fig. 16-5 UART Auto flow control block diagram

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)

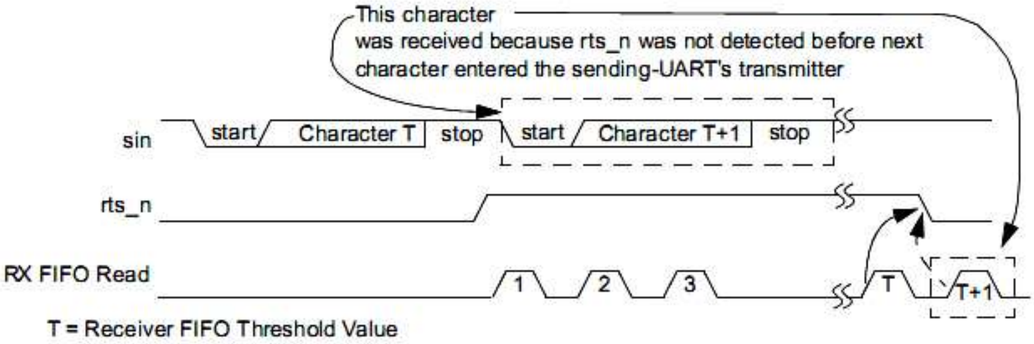

This character was received because rts_n was not detected before next character entered the sending-UART's transmitter
sin
start
Character T
stop
start
Character T+1
stop
rts_n
RX FIFO Read
1
2
3
T
T+1
T = Receiver FIFO Threshold Value

Fig. 16-6 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)

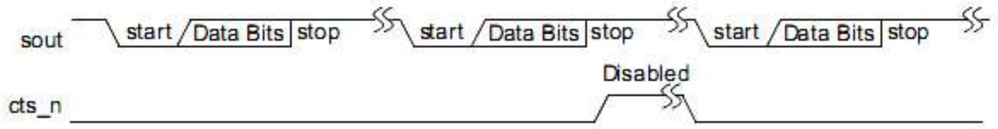


Fig. 16-7 UART AUTO CTS TIMING

16.4 Register Description

This section describes the control/status registers of the design. There are 5 UARTs in RK3288, and each one has its own base address.

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000001	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000060	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030~0x006c	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x0030~0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000006	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x3330382a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

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

16.4.2 Detail Register Description

UART_RBR
Address: Operational Base + offset (0x0000)
Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
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)
Divisor Latch (Low)

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 frequency) / (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. 1'b0: disabled 1'b1: enabled
6:4	RO	0x0	reserved
3	RW	0x0	modem_status_int_en Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 1'b0: disabled 1'b1: enabled
2	RW	0x0	receive_line_status_int_en Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 1'b0: disabled 1'b1: enabled
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt.

Bit	Attr	Reset Value	Description
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 1'b0: disabled 1'b1: enabled

UART_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. 2'b00: disabled 2'b11: enabled
5:4	RO	0x0	reserved
3:0	RO	0x1	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 4'b0000: modem status 4'b0001: no interrupt pending 4'b0010: THR empty 4'b0100: received data available 4'b0110: receiver line status 4'b0111: busy detect 4'b1100: character timeout

UART_FCR
Address: Operational Base + offset (0x0008)
FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	WO	0x0	rcvr_trigger RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. 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: 2'b00: FIFO empty 2'b01: 2 characters in the FIFO 2'b10: FIFO 1/4 full 2'b11: FIFO 1/2 full
3	WO	0x0	dma_mode DMA Mode This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 1'b0: mode 0 1'b1: 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 select. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

Bit	Attr	Reset Value	Description
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.
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

Bit	Attr	Reset Value	Description
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. 1'b0: parity disabled 1'b1: 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 select, the receiver checks only the first stop bit. 1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.

Bit	Attr	Reset Value	Description
1:0	RW	0x0	data_length_sel Data Length Select. 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: 2'b00: 5 bits 1'b01: 6 bits 1'b10: 7 bits 1'b11: 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. 1'b0: IrDA SIR Mode disabled 1'b1: IrDA SIR Mode enabled
5	RW	0x0	auto_flow_ctrl_en Auto Flow Control Enable. 1'b0: Auto Flow Control Mode disabled 1'b1: Auto Flow Control Mode enabled
4	RW	0x0	loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)
2	RW	0x0	out1 OUT1

Bit	Attr	Reset Value	Description
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: 1'b0: dtr_n de-asserted (logic 1) 1'b1: 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. 1'b0: no error in RX FIFO 1'b1: error in RX FIFO
6	RO	0x1	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	0x1	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 1’b0: no data ready 1’b1: 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
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 (0x0030~0x006c)
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 (0x0030~0x006c)
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. 1'b0: FIFO access mode disabled 1'b1: 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. 1'b0: Receive FIFO not full 1'b1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 1'b0: Receive FIFO is empty 1'b1: Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	RO	0x1	trasn_fifo_empty Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty

Bit	Attr	Reset Value	Description
1	RO	0x1	trans_fifo_not_full Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 1’b0: Transmit FIFO is full 1’b1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	RO	0x0	uart_busy UART Busy. UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the UART is idle or inactive. 1’b0: UART is idle or inactive 1’b1: 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]).

Bit	Attr	Reset Value	Description
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS

Address: Operational Base + offset (0x008c)

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
1:0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET
Address: Operational Base + offset (0x00a0)
Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
1:0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX
Address: Operational Base + offset (0x00a4)
Halt TX

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 1'b0: Halt TX disabled 1'b1: Halt TX enabled

UART_DMASA
Address: Operational Base + offset (0x00a8)
DMA Software Acknowledge

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

Bit	Attr	Reset Value	Description
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_UCV
Address: Operational Base + offset (0x00f8)
UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x3330382a	ver ASCII value for each number in the version

UART_CTR
Address: Operational Base + offset (0x00fc)
Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

16.5 Interface description

Table 16-1 UART Interface Description

Module pin	Direction	Pad name	IOMUX
UART_BT Interface			
uartbt_sin	I	GPIO4_C[0]	GPIO1A_IOMUX[0]=1
uartbt_sout	O	GPIO4_C[1]	GPIO1A_IOMUX[2]=1
uartbt_cts_n	I	GPIO4_C[2]	GPIO1A_IOMUX[4]=1
uartbt_rts_n	O	GPIO4_C[3]	GPIO1A_IOMUX[6]=1
UART_BB Interface			
uartbb_sin	I	GPIO5_B[0]	GPIO1A_IOMUX[1:0]=01
uartbb_sout	O	GPIO5_B[1]	GPIO1A_IOMUX[3:2]=01
uartbb_cts_n	I	GPIO5_B[2]	GPIO1A_IOMUX[5:4]=01
uartbb_rts_n	O	GPIO5_B[3]	GPIO1A_IOMUX[7:6]=01
UART_DBG Interface			
uartdbg_sin	I	GPIO7_C[6]	GPIO7CH_IOMUX[9:8]=01
uartdbg_sout	O	GPIO7_C[7]	GPIO7CH_IOMUX[14:12]=001
uartdbg_sirsin	I	GPIO7_C[6]	GPIO7CH_IOMUX[9:8]=10
Uartdbg_sirout	O	GPIO7_C[7]	GPIO7CH_IOMUX[14:12]=010
UART_GPS Interface			
uartgps_sin	I	GPIO7_A[7]	GPIO7A_IOMUX[15:14]=01
uartgps_sout	O	GPIO7_B[0]	GPIO7B_IOMUX[1:0]=01
uartgps_cts_n	I	GPIO7_B[1]	GPIO7B_IOMUX[3:2]=01
uartgps_rts_n	O	GPIO7_B[2]	GPIO7B_IOMUX[5:4]=01
UART_EXP Interface			
uartexp_sin	I	GPIO5_B[7]	GPIO5B_IOMUX[15:14]=11
uartexp_sout	O	GPIO5_B[6]	GPIO5B_IOMUX[13:12]=11
uartexp_cts_n	I	GPIO5_B[4]	GPIO5B_IOMUX[9:8]=11
uartexp_rts_n	O	GPIO5_B[5]	GPIO5B_IOMUX[11:10]=11

16.6 Application Notes

16.6.1 None FIFO Mode Transfer Flow

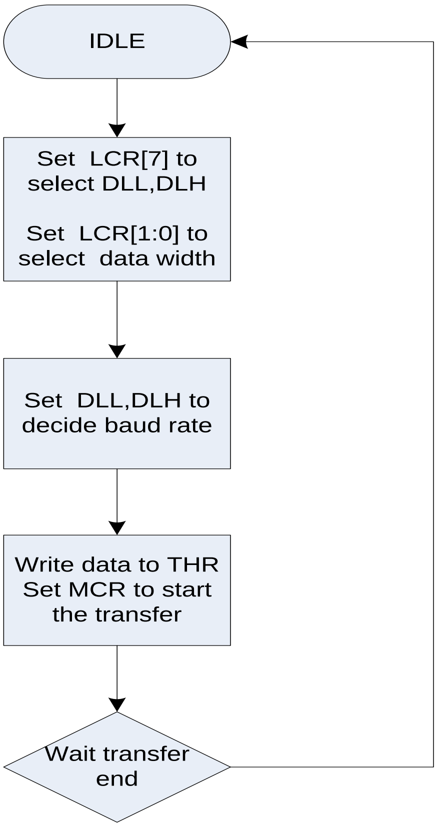

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set DLL,DLH to decide baud rate
Write data to THR Set MCR to start the transfer
Wait transfer end

Fig. 16-8 UART none fifo mode

16.6.2 FIFO Mode Transfer Flow

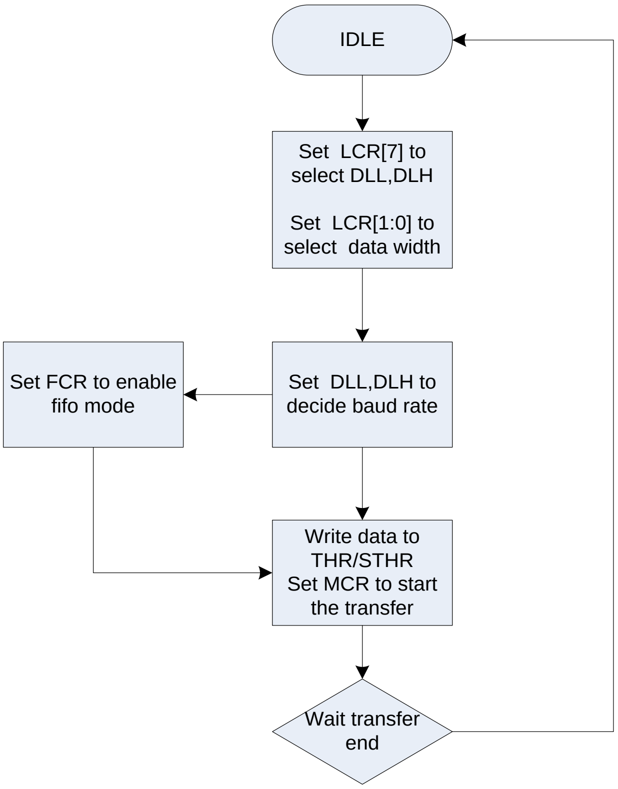

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set FCR to enable fifo mode
Set DLL,DLH to decide baud rate
Write data to THR/STHR
Set MCR to start the transfer
Wait transfer end

Fig. 16-9 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.

16.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.
UART source clocks can be selected from CODEC PLL and GENERAL PLL outputs. UART_BT source clocks can also be selected from NEW PLL and 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.

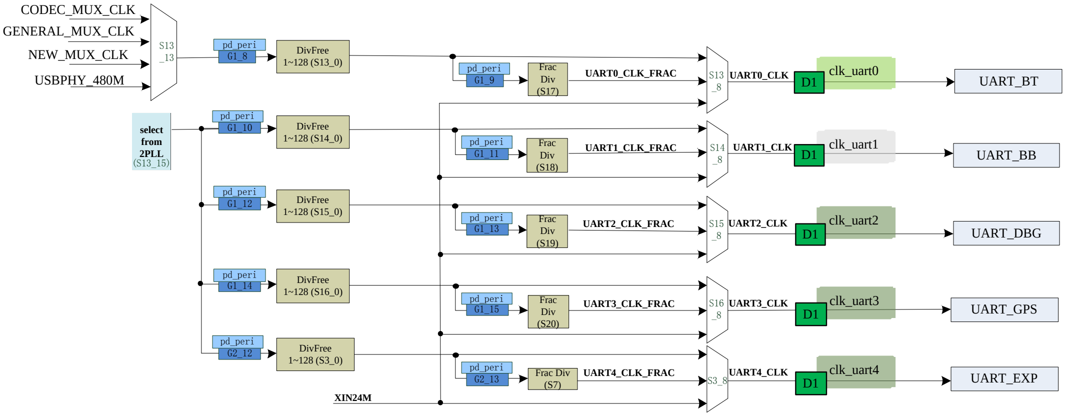


Fig. 16-10 UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 16-2 UART baud rate configuration

Baud Rate	Reference Configuration

115.2 Kbps	Configure GENERAL PLL to get 648MHz clock output;
	Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;
	Config UART_DLL to 8.

460.8 Kbps	Configure GENERAL PLL to get 648MHz clock output;
	Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;
	Configure UART_DLL to 2.

921.6 Kbps	Configure GENERAL PLL to get 648MHz clock output;
	Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock;
	Configure UART_DLL to 1.

1.5 Mbps	Choose GENERAL PLL to get 384MHz clock output;
	Divide 384MHz clock by 16 to get 24MHz clock;
	Configure UART_DLL to 1

3 Mbps	Choose GENERAL PLL to get 384MHz clock output;
	Divide 384MHz clock by 8 to get 48MHz clock;
	Configure UART_DLL to 1

4 Mbps	Configure GENERAL PLL to get 384MHz clock output;
	Divide 384MHz clock by 6 to get 64MHz clock;
	Configure UART_DLL to 1

16.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

GRF_SOC_CON13[4:0] (grf_uart_cts_sel[4:0]) used to configure the polarity of cts_n. Every bit for one UART, bit4 is for UART_EXP, bit3 is for UART_GPS, bit2 is for UART_DBG, bit1 is for UART_BB, bit0 is for UART_BT.
GRF_SOC_CON13[9:5] (grf_uart_rts_sel[4:0]) used to configure the polarity of rts_n. Every bit for one UART, bit4 is for UART_EXP, bit3 is for UART_GPS, bit2 is for UART_DBG, bit1 is for UART_BB, bit0 is for UART_BT.
When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, low active.
When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, low active.

Chapter 17 GPIO

17.1 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

17.2 Block Diagram

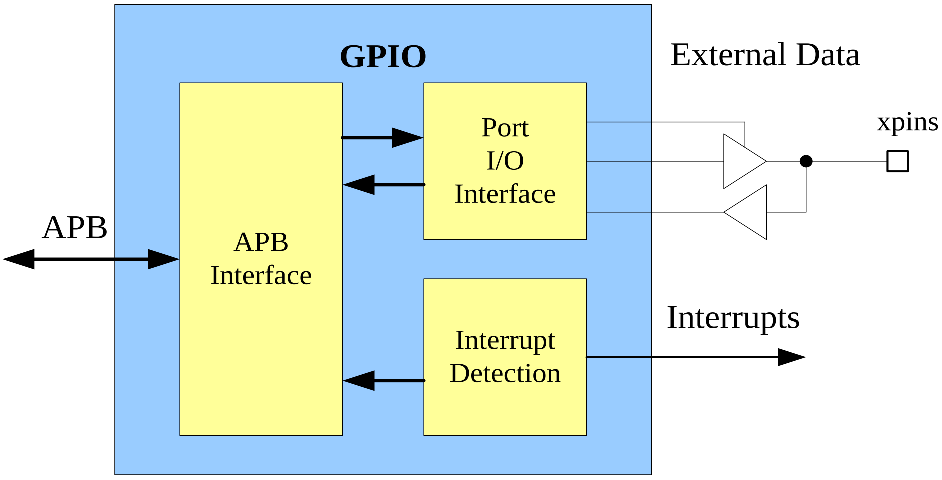


Fig. 17-1 GPIO block diagram

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.

17.3 Function description

17.3.1 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.

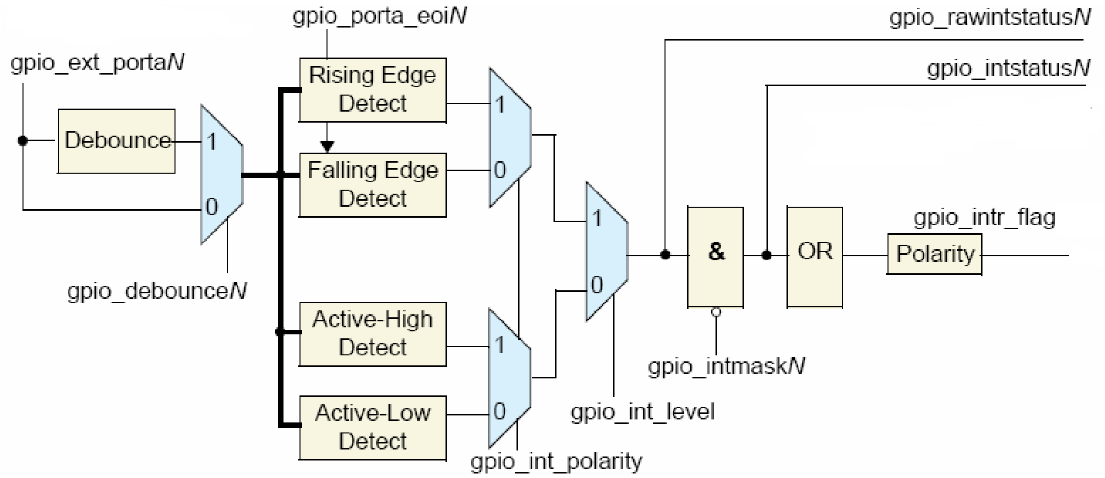


Fig. 17-2 GPIO Interrupt RTL Block Diagram

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).

17.3.2 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 alevel-sensitive interrupt.

9 GPIOs' hierarchy in the chip

GPIO0 is in PD_PMU subsystem, GPIO1~8 are in PD_ALIVE subsystem.

17.4 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 9 GPIOs (GPIO0 ~ GPIO8), and each of them has same register group. Therefore, 9 GPIOs' register groups have 9 different base address.

17.4.1 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*

17.4.2 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

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 1'b0: Input (default) 1'b1: Output

GPIO_INTEN
Address: Operational Base + offset (0x0030)
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. 1'b0: Configure Port A bit as normal GPIO signal (default) 1'b1: Configure Port A bit as interrupt

GPIO_INTMASK
Address: Operational Base + offset (0x0034)
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. 1'b0: Interrupt bits are unmasked (default) 1'b1: Mask interrupt

GPIO_INTTYPE_LEVEL
Address: Operational Base + offset (0x0038)
Interrupt level register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 1’b0: Level-sensitive (default) 1’b1: Edge-sensitive

GPIO_INT_POLARITY
Address: Operational Base + offset (0x003c)
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. 1’b0: Active-low (default) 1’b1: 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

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 1'b0: No debounce (default) 1'b1: Enable debounce

GPIO_PORTA_EOI
Address: Operational Base + offset (0x004c)
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. 1'b0: No interrupt clear (default) 1'b1: 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

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 1’b0: No synchronization to pclk_intr (default) 1’b1: Synchronize to pclk_intr

17.5 Interface description

Table 17-1 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’h0000
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0]=16’h0000
gpio0_porta[23:16]	I/O	GPIO0_C[7:0]	GRF_GPIO0C_IOMUX[15:0]=16’h0000
gpio0_porta[31:24]	I/O	GPIO0_D[7:0]	GRF_GPIO0D_IOMUX[15:0]=16’h0000
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0]=16’h0000
gpio1_porta[15:8]	I/O	GPIO1_B[7:0]	GRF_GPIO1B_IOMUX[15:0]=16’h0000
gpio1_porta[23:16]	I/O	GPIO1_C[7:0]	GRF_GPIO1C_IOMUX[15:0]=16’h0000
gpio1_porta[31:24]	I/O	GPIO1_D[7:0]	GRF_GPIO1D_IOMUX[15:0]=16’h0000
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0]=16’h0000
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2B_IOMUX[15:0]=16’h0000
gpio2_porta[23:16]	I/O	GPIO2_C[7:0]	GRF_GPIO2C_IOMUX[15:0]=16’h0000
gpio2_porta[31:24]	I/O	GPIO2_D[7:0]	GRF_GPIO2D_IOMUX[15:0]=16’h0000
GPIO3 Interface			
gpio3_porta[7:0]	I/O	GPIO3_A[7:0]	GRF_GPIO3A_IOMUX[15:0]=16’h0000
gpio3_porta[15:8]	I/O	GPIO3_B[7:0]	GRF_GPIO3B_IOMUX[15:0]=16’h0000
gpio3_porta[23:16]	I/O	GPIO3_C[7:0]	GRF_GPIO3C_IOMUX[15:0]=16’h0000
gpio3_porta[31:24]	I/O	GPIO3_D[7:0]	GRF_GPIO3D_IOMUX[15:0]=16’h0000

17.6 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 18 I2C Interface

18.1 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
- There are two I2C controller in bus sub-system:I2C PMU, and I2C AUDIO. There are four I2C controller in peripheral sub-system: I2C SENSOR, I2C CAM, I2C_TP, and I2C_HDMI.

18.2 Block Diagram

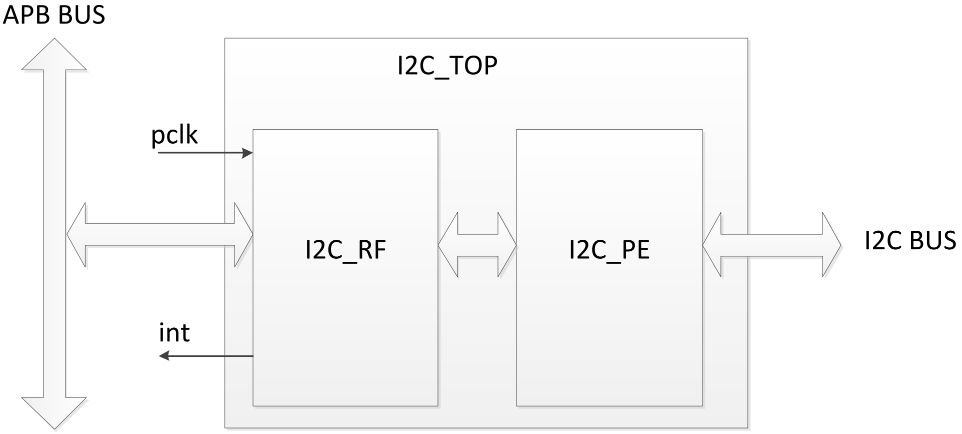


Fig. 18-1 I2C architecture

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.

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.

I2C_TOP

I2C_TOP module is the top module of the I2C controller.

18.3 Function description

This chapter provides a description about the functions and behavior under various conditions. The I2C controller supports only Master function. It supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

18.3.1 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 & 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 system clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

18.3.2 Master Mode Programming

1. SCL Clock: When the I2C controller is programmed in Master mode, the SCL frequency is determine by I2C_CLKDIV register. The SCL frequency is calculated by the following formula：
SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)
SCL = PCLK/ SCLK Divisor

2. Data Receiver Register Access
When the i2c controller received MRXCNT bytes data， CPU can get the datas through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 byte data in one transaction.
When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

3. Transmit Trasmitter Register
Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 byte data in one transaction. The lower byte will be transmitted first.
When MTXCNT register is written, the I2C controller will start to transmit data.

4. Start Command
Write 1 to I2C_CON[3], the controller will send I2C start command.

5. Stop Command
Write 1 to I2C_CON[4], the controller will send I2C stop command

6. 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 triggered 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.

7. 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].

8. 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 complete transmitting a byte.
Byte received finish interrupt (Bit 1): The bit is asserted when Master complete receiving a byte.
MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master complete transmitting MTXCNT bytes.
MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master complete receiving MRXCNT bytes.
Start interrupt (Bit 4): The bit is asserted when Master finish asserting start command to I2C bus.
Stop interrupt (Bit 5): The bit is asserted when Master finish asserting stop command to I2C

bus.

NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

9. 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.

10. 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.

11. I2C controller data transfer waveform

Bit transferring

(a) 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.

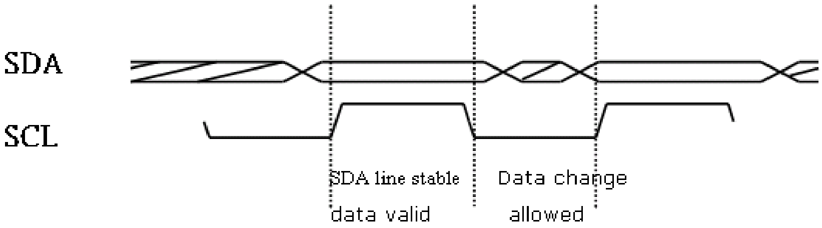


Fig. 18-2 I2C DATA Validity

(b) 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.

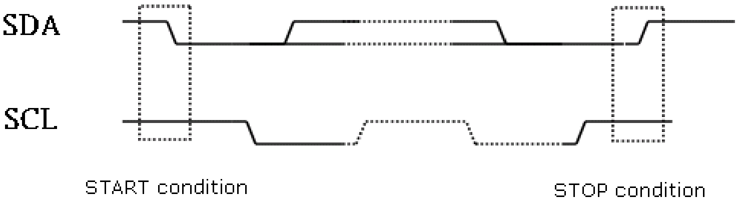


Fig. 18-3 I2C Start and stop conditions

Data transfer

(a) 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".

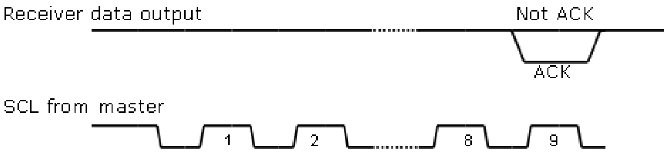


Fig. 18-4 I2C Acknowledge

(b) 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.

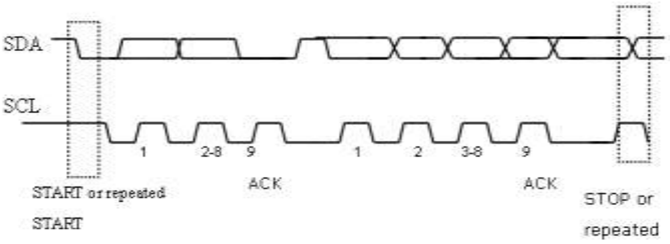


Fig. 18-5 I2C byte transfer

18.4 Register Description

This section describes the control/status registers of the design.

18.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2C_CON	0x0000	W	0x00000000	control register
I2C_CLKDIV	0x0004	W	0x00060006	Clock divisor register
I2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master receive mode
I2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master recevie mode
I2C_MTXCNT	0x0010	W	0x00000000	master transmit count
I2C_MRXCNT	0x0014	W	0x00000000	master receive count
I2C_IEN	0x0018	W	0x00000000	interrupt enable register
I2C_IPD	0x001c	W	0x00000000	interrupt pending register
I2C_FCNT	0x0020	W	0x00000000	finished count
I2C_TXDATA0	0x0100	W	0x00000000	I2C transmit data register 0
I2C_TXDATA1	0x0104	W	0x00000000	I2C transmit data register 1
I2C_TXDATA2	0x0108	W	0x00000000	I2C transmit data register 2
I2C_TXDATA3	0x010c	W	0x00000000	I2C transmit data register 3
I2C_TXDATA4	0x0110	W	0x00000000	I2C transmit data register 4
I2C_TXDATA5	0x0114	W	0x00000000	I2C transmit data register 5
I2C_TXDATA6	0x0118	W	0x00000000	I2C transmit data register 6
I2C_TXDATA7	0x011c	W	0x00000000	I2C transmit data register 7
I2C_RXDATA0	0x0200	W	0x00000000	I2C receive data register 0
I2C_RXDATA1	0x0204	W	0x00000000	I2C receive data register 1
I2C_RXDATA2	0x0208	W	0x00000000	I2C receive data register 2
I2C_RXDATA3	0x020c	W	0x00000000	I2C receive data register 3
I2C_RXDATA4	0x0210	W	0x00000000	I2C receive data register 4
I2C_RXDATA5	0x0214	W	0x00000000	I2C receive data register 5
I2C_RXDATA6	0x0218	W	0x00000000	I2C receive data register 6
I2C_RXDATA7	0x021c	W	0x00000000	I2C receive data register 7

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

18.4.2 Detail Register Description

I2C_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 last byte acknowledge control in master receive mode . 1'b0: ACK 1'b1: NAK
4	W1C	0x0	stop stop enable when this bit is written to 1，I2C will generate stop signal. It cleared itself when stop operation ends.
3	W1C	0x0	start start enable when this bit is written to 1，I2C will generate start signal. It cleared itself when start operation ends.
2:1	RW	0x0	i2c_mode 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

I2C_CLKDIV
Address: Operational Base + offset (0x0004)
Clock divisor register

Bit	Attr	Reset Value	Description
31:16	RW	0x0006	CLKDIVH SCL high level clock count T（SCL_HIGH）= T(PCLK) * (CLKDIVH +1)* 8
15:0	RW	0x0006	CLKDIVL SCL low level clock count T（SCL_LOW）= T(PCLK) * (CLKDIVL+1) * 8

I2C_MRXADDR
Address: Operational Base + offset (0x0008)
the slave address accessed for master receive mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid
25	RW	0x0	addmvld address middle byte valid
24	RW	0x0	addlvld address low byte valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read

I2C_MRXRADDR

Address: Operational Base + offset (0x000c)
the slave register address accessed   for master receive mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid
25	RW	0x0	sraddmvld address middle byte valid
24	RW	0x0	sraddlvld address low byte valid
23:0	RW	0x000000	sraddr slave register address accessed

I2C_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

I2C_MRXCNT

Address: Operational Base + offset (0x0014)
master receive count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master receive count

I2C_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
5	RW	0x0	stopien stop operation finished interrupt enable
4	RW	0x0	startien start operation finished interrupt enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable
2	RW	0x0	mbtfien MTXCNT data transmit finished interrupt enable
1	RW	0x0	brfien byte receive finished interrupt enable
0	RW	0x0	btfien byte transmit finished interrupt enable

I2C_IPD
Address: Operational Base + offset (0x001c)
interrupt pending register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	nakrcvipd NAK handshake received interrupt pending bit
5	RW	0x0	stopipd stop operation finished interrupt pending bit
4	RW	0x0	startipd start operation finished interrupt pending bit
3	RW	0x0	mbrfipd MRXCNT data received finished interrupt pending bit
2	RW	0x0	mbtfipd MTXCNT data transmit finished interrupt pending bit
1	RW	0x0	brfipd byte receive finished interrupt pending bit
0	RW	0x0	btfipd byte transmit finished interrupt pending bit

I2C_FCNT
Address: Operational Base + offset (0x0020)
finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

I2C_TXDATA0
Address: Operational Base + offset (0x0100)
I2C transmit data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0

I2C_TXDATA1
Address: Operational Base + offset (0x0104)
I2C transmit data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1

I2C_TXDATA2
Address: Operational Base + offset (0x0108)
I2C transmit data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2

I2C_TXDATA3
Address: Operational Base + offset (0x010c)
I2C transmit data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3

I2C_TXDATA4
Address: Operational Base + offset (0x0110)
I2C transmit data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4

I2C_TXDATA5
Address: Operational Base + offset (0x0114)
I2C transmit data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5

I2C_TXDATA6
Address: Operational Base + offset (0x0118)

I2C transmit data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6

I2C_TXDATA7
Address: Operational Base + offset (0x011c)
I2C transmit data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7

I2C_RXDATA0
Address: Operational Base + offset (0x0200)
I2C receive data register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0

I2C_RXDATA1
Address: Operational Base + offset (0x0204)
I2C receive data register 1

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1

I2C_RXDATA2
Address: Operational Base + offset (0x0208)
I2C receive data register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2

I2C_RXDATA3
Address: Operational Base + offset (0x020c)
I2C receive data register 3

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3

I2C_RXDATA4
Address: Operational Base + offset (0x0210)
I2C receive data register 4

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4

I2C_RXDATA5
Address: Operational Base + offset (0x0214)
I2C receive data register 5

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5

I2C_RXDATA6
Address: Operational Base + offset (0x0218)
I2C receive data register 6

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6

I2C_RXDATA7
Address: Operational Base + offset (0x021c)
I2C receive data register 7

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7

18.5 Interface description

Table 18-1 I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C PMU Interface			
i2c_pmu_sda	I/O	IO_I2C0PMUsda_PMUgpio0b7	GRF_GPIO0B_IOMUX[14]=1
i2c_pmu_scl	I/O	IO_I2C0PMUscl_PMUgpio0c0	GRF_GPIO0C_IOMUX[0]=1
I2C SENSOR Interface			
i2c_sensor_sda	I/O	IO_I2C1SENSORsda_SCrst_GPIO1830gpio8a4	GRF_GPIO8A_IOMUX[9:8]=01
i2c_sensor_scl	I/O	IO_I2C1SENSORscl_SCclk_GPIO1830gpio8a5	GRF_GPIO8A_IOMUX[11:10]=01
I2C AUDIO Interface			
i2c_audio_sda	I/O	IO_I2C2AUDIOsda_AUDIOgpio6b1	GRF_GPIO6B_IOMUX[2]=1
i2c_audio_scl	I/O	IO_I2C2AUDIOscl_AUDIOgpio6b2	GRF_GPIO6B_IOMUX[4]=1
I2C CAM Interface			
i2c_cam_sda	I/O	IO_I2C3CAMsda_DVPgpio2c1	GRF_GPIO2C_IOMUX[2]=1
i2c_cam_scl	I/O	IO_I2C3CAMscl_DVPgpio2c0	GRF_GPIO2C_IOMUX[0]=1
I2C TP Interface			
i2c_tp_sda	I/O	IO_I2C4TPsda_GPIO30gpio7c1	GRF_GPIO7CL_IOMUX[4]=1
i2c_tp_scl	I/O	IO_I2C4TPscl_GPIO30gpio7c2	GRF_GPIO7CL_IOMUX[8]=1
I2C HDMI Interface			
i2c_hdmi_sda	I/O	IO_I2C5HDMIsda_EDPHDMII2Csda_GPIO30gpio7c3	GRF_GPIO7CL_IOMUX[13:12]=01
i2c_hdmi_scl	I/O	IO_I2C5HDMIscl_EDPHDMII2Cscl_GPIO30gpio7c4	GRF_GPIO7CH_IOMUX[1:0]=01

18.6 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 following.

- Transmit only mode (I2C_CON[1:0]=2'b00)

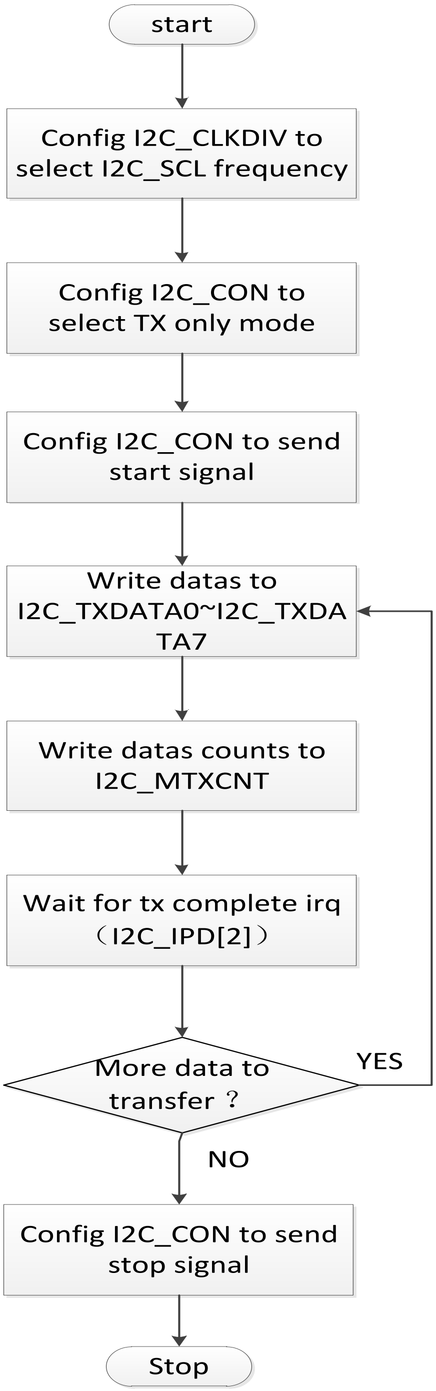

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select TX only mode
Config I2C_CON to send start signal
Write datas to I2C_TXDATA0~I2C_TXDATA7
Write datas counts to I2C_MTXCNT
Wait for tx complete irq （I2C_IPD[2]）
More data to transfer ？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig. 18-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

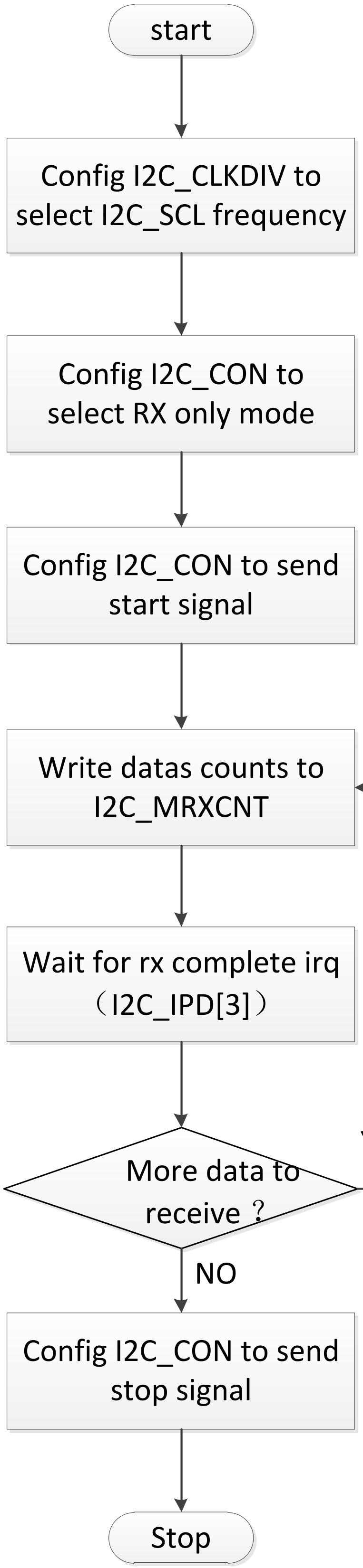

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select RX only mode
Config I2C_CON to send start signal
Write datas counts to I2C_MRXCNT
Wait for rx complete irq （I2C_IPD[3]）
More data to receive ？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig. 18-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2’b01 or I2C_CON[1:0]=2’b11)

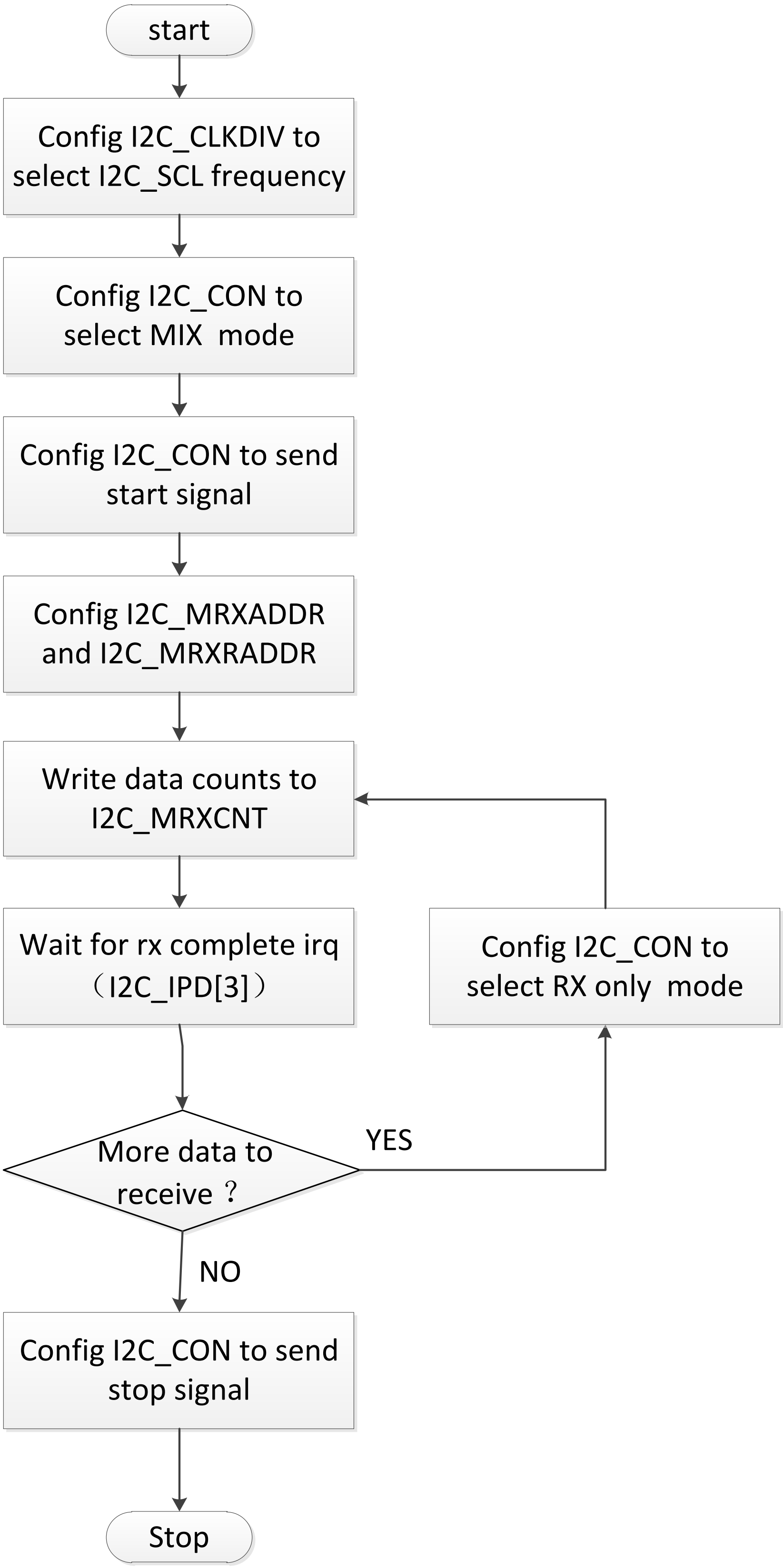

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select MIX mode
Config I2C_CON to send start signal
Config I2C_MRXADDR and I2C_MRXRADDR
Write data counts to I2C_MRXCNT
Wait for rx complete irq （I2C_IPD[3]）
More data to receive ？
YES
Config I2C_CON to select RX only mode
NO
Config I2C_CON to send stop signal
Stop

Fig. 18-8 I2C Flow chat for mix mode

Chapter 19 I2S/PCM Controller

19.1 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 (up to 7.1channel) and stereo input are supported in I2S/PCM controller.

- Support five internal 32-bit wide and 32-location deep FIFOs, four for transmitting and one for receiving audio data
- 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 2,4,6,8 channels audio transmitting in I2S and PCM mode
- Support 2 channels audio receiving in I2S and PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support 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 3 independent LRCK signals, one for receiving and two for transmitting audio data
- Support configurable SCLK and LRCK polarity
- Support SCLK is equivalent to MCLK divided by an even number range from 2 to 64 in master mode

19.2 Block Diagram

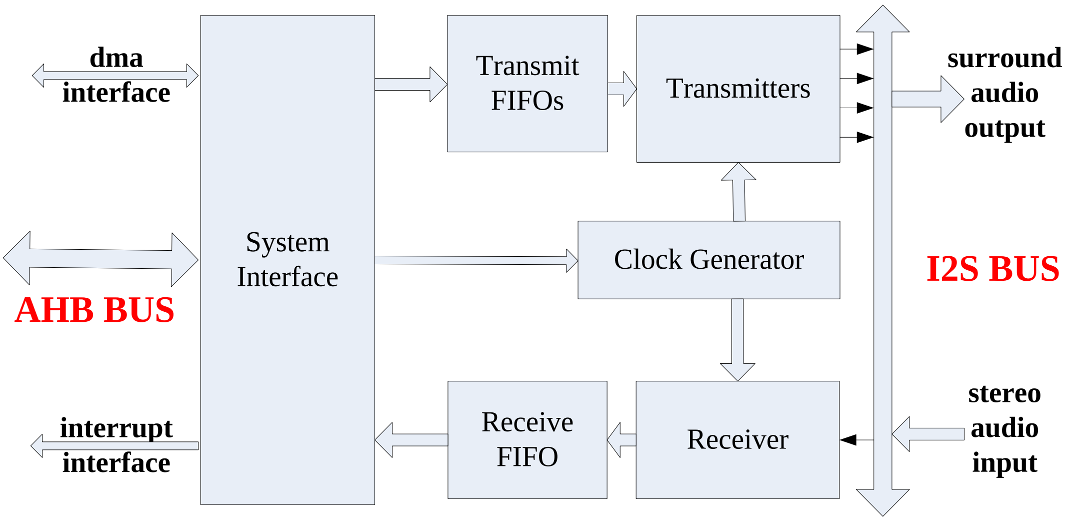


Fig. 19-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters 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.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either master or slave, with I2S or PCM mode surround (up to 7.1 channel) 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 FIFOs

The Transmit FIFOs are the buffer to store transmitted audio data. Each of the size of the four FIFOs 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.

19.3 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.

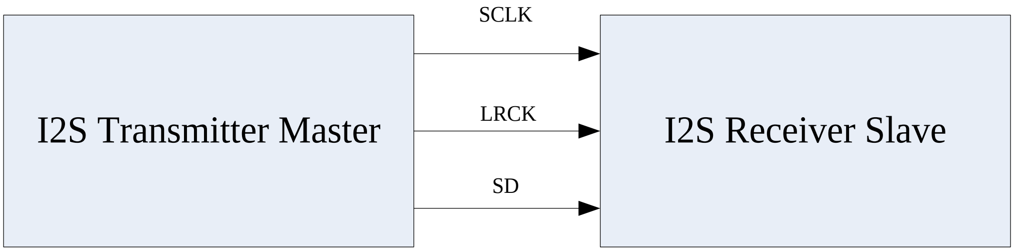


Fig. 19-2 I2S transmitter-master & receiver-slave condition

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.

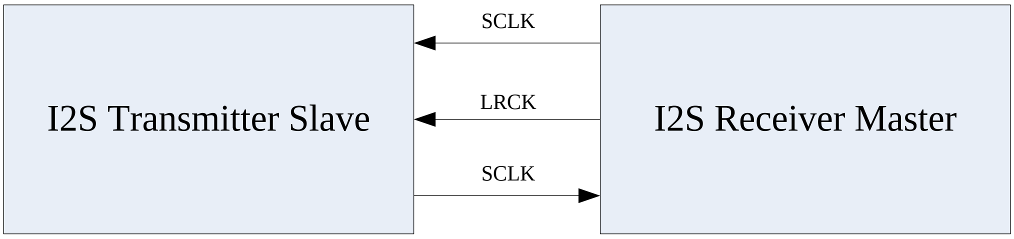


Fig. 19-3 I2S transmitter-slave& receiver-master condition

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 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.

19.3.1 i2s normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx0) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, 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.

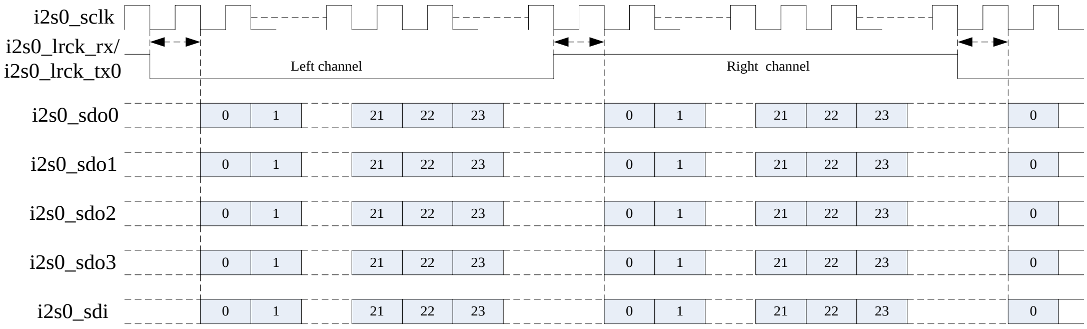


Fig. 19-4 I2S normal mode timing format

19.3.2 i2s left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, 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.

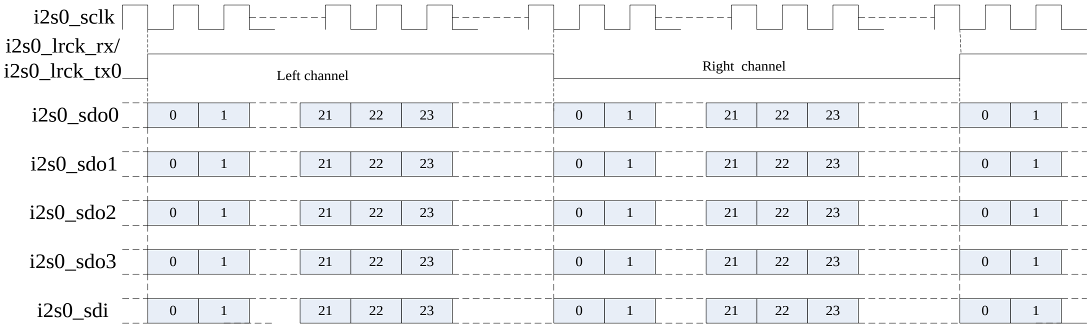


Fig. 19-5 I2S left justified mode timing format

19.3.3 i2s right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo0, i2s_sdo1, i2s_sdo2, i2s_sdo3, 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.

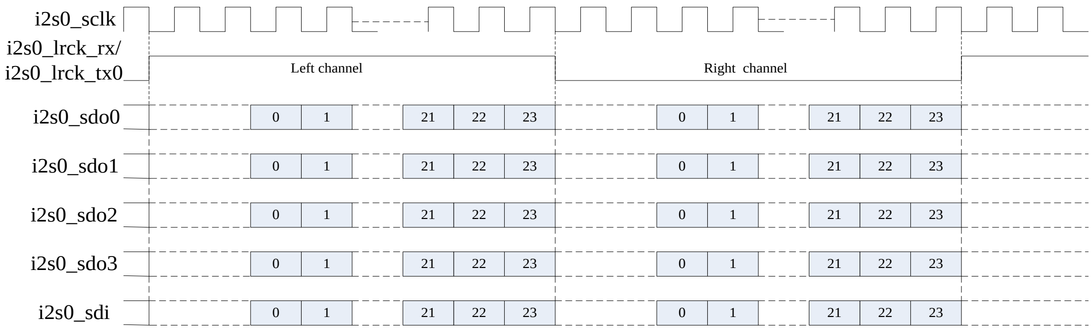


Fig. 19-6 I2S right justified modetiming format

19.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, 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.

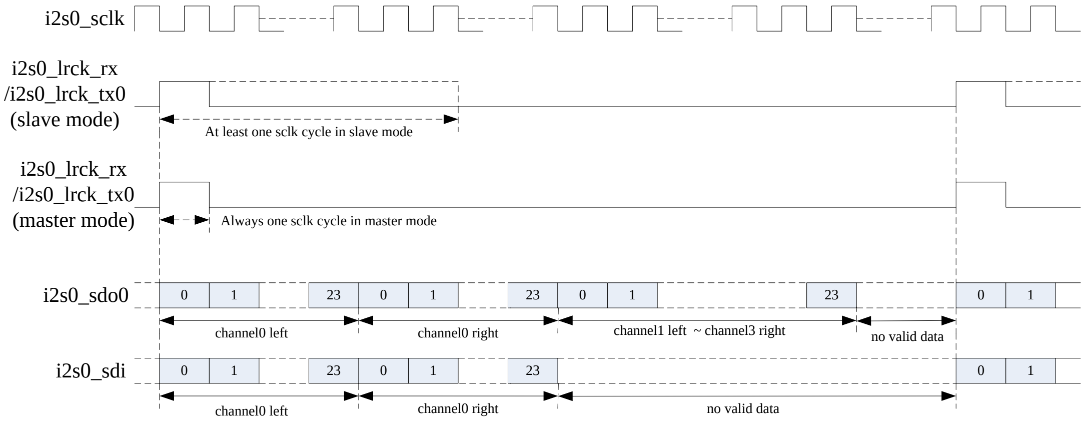


Fig. 19-7 PCM early modetiming format

19.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, 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.

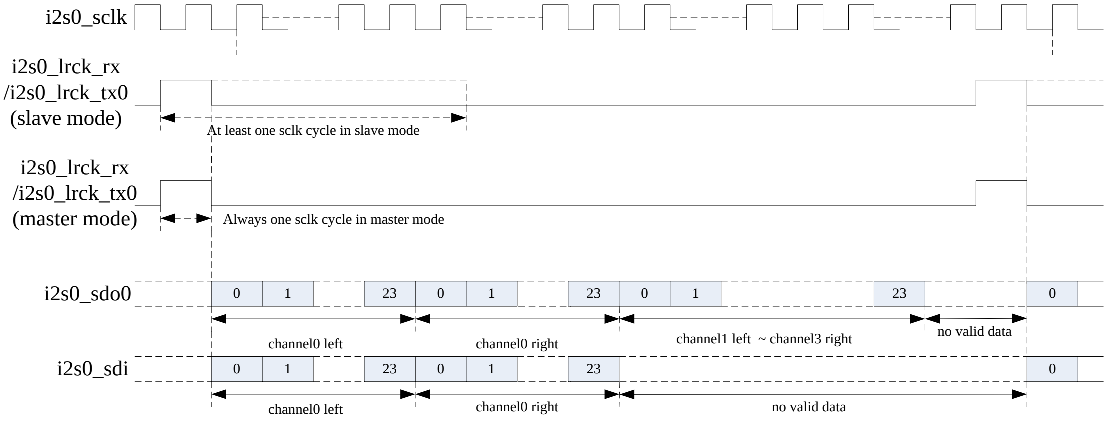


Fig. 19-8 PCM late1 modetiming format

19.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, 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.

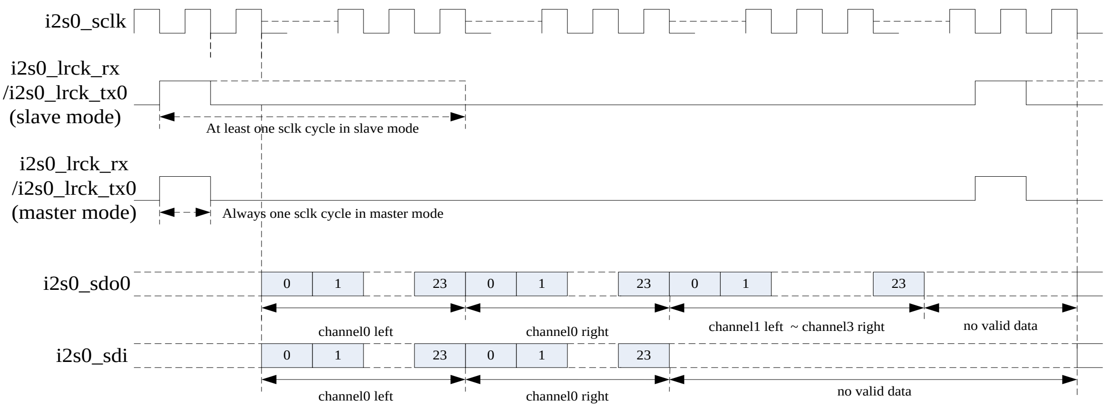


Fig. 19-9 PCM late2 modetiming format

19.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo0, 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.

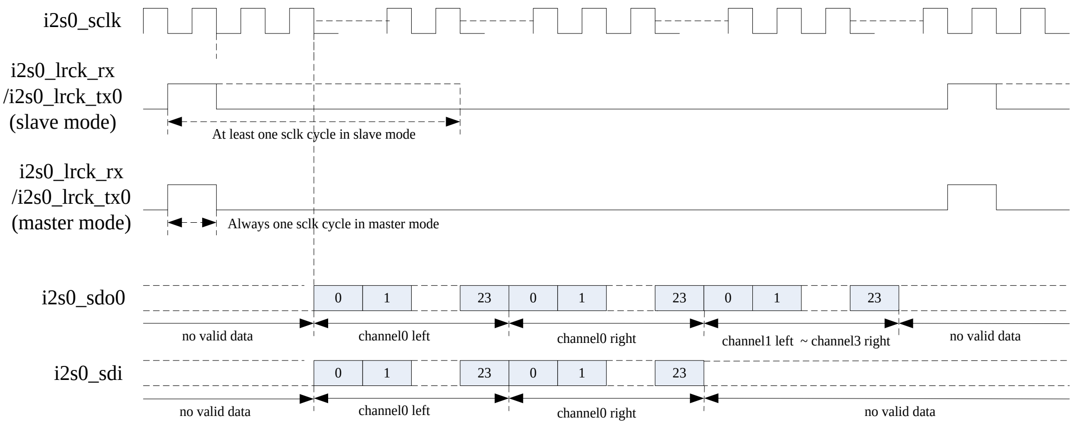


Fig. 19-10 PCM late3 modetiming format

19.4 Register Description

This section describes the control/status registers of the design.

19.4.1 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_FIFOLR	0x000c	W	0x00000000	FIFO level register
I2S_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_INTCR	0x0014	W	0x01f00000	interrupt control register
I2S_INTSR	0x0018	W	0x00000000	interrupt status register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S_TXDR	0x0024	W	0x00000000	Transimt FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register

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

19.4.2 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 jusitified counter (Can be written only when XFER[0] bit is 0.) Only vailid in I2S Right justified format and slave tx mode is selected. Start to tramsmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel select register (Can be written only when XFER[0] bit is 0.) 0:channel 0 enable 1:channel 0 & channel 1 enable 2:channel 0 & channel 1 & channel 2 enable 3:channel 0 & channel 1 & channel 2 & channel 3 enable
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
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB

Bit	Attr	Reset Value	Description
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
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 …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_RXCR
Address: Operational Base + offset (0x0004)
receive operation control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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
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 …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S_CKR
Address: Operational Base + offset (0x0008)
clock generation register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
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 polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)

Bit	Attr	Reset Value	Description

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;
			……
			60,61:Fmclk=62*Ftxsclk;
			62,63:Fmclk=64*Ftxsclk;
			……
			252,253:Fmclk=254*Ftxsclk;
			254,255:Fmclk=256*Ftxsclk;

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 …… 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 …… 253: 254fs 254: 255fs 255: 256fs

I2S_FIFOLR

Address: Operational Base + offset (0x000c)

FIFO level register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
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 is equal to or above this field value + 1.
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2S_INTCR
Address: Operational Base + offset (0x0014)
interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 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 CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.

Bit	Attr	Reset Value	Description
3	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
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 cleard bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleard bit. Write 1 to clear all transmit logic.

I2S_TXDR
Address: Operational Base + offset (0x0400~0x7FC)
Transimt FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transimt FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR
Address: Operational Base + offset (0x0800~0xBFC)
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.

19.5 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s_sdi	I	IO_I2Ssdi_AUDIOgpio6a3	GRF_GPIO6A_IOMUX[6]=1
i2s_clk	O	IO_I2Sclk_AUDIOgpio6b0	GRF_GPIO6B_IOMUX[0]=1
i2s_sclk	I/O	IO_I2Ssclk_AUDIOgpio6a0	GRF_GPIO6A_IOMUX[0]=1

i2s_lrck_rx	I/O	IO_I2Slrckrx_AUDIOgpio6a1	GRF_GPIO6A_IOMUX[2]=1
i2s_lrck_tx	I/O	IO_I2Slrcktx_AUDIOgpio6a2	GRF_GPIO6A_IOMUX[4]=1
i2s_sdo0	O	IO_I2Ssdo0_AUDIOgpio6a4	GRF_GPIO6A_IOMUX[8]=1
i2s_sdo1	O	IO_I2Ssdo1_AUDIOgpio6a5	GRF_GPIO6A_IOMUX[10]=1
i2s_sdo2	O	IO_I2Ssdo2_AUDIOgpio6a6	GRF_GPIO6A_IOMUX[12]=1
i2s_sdo3	O	IO_I2Ssdo3_AUDIOgpio6a7	GRF_GPIO6A_IOMUX[14]=1

19.6 Application Notes

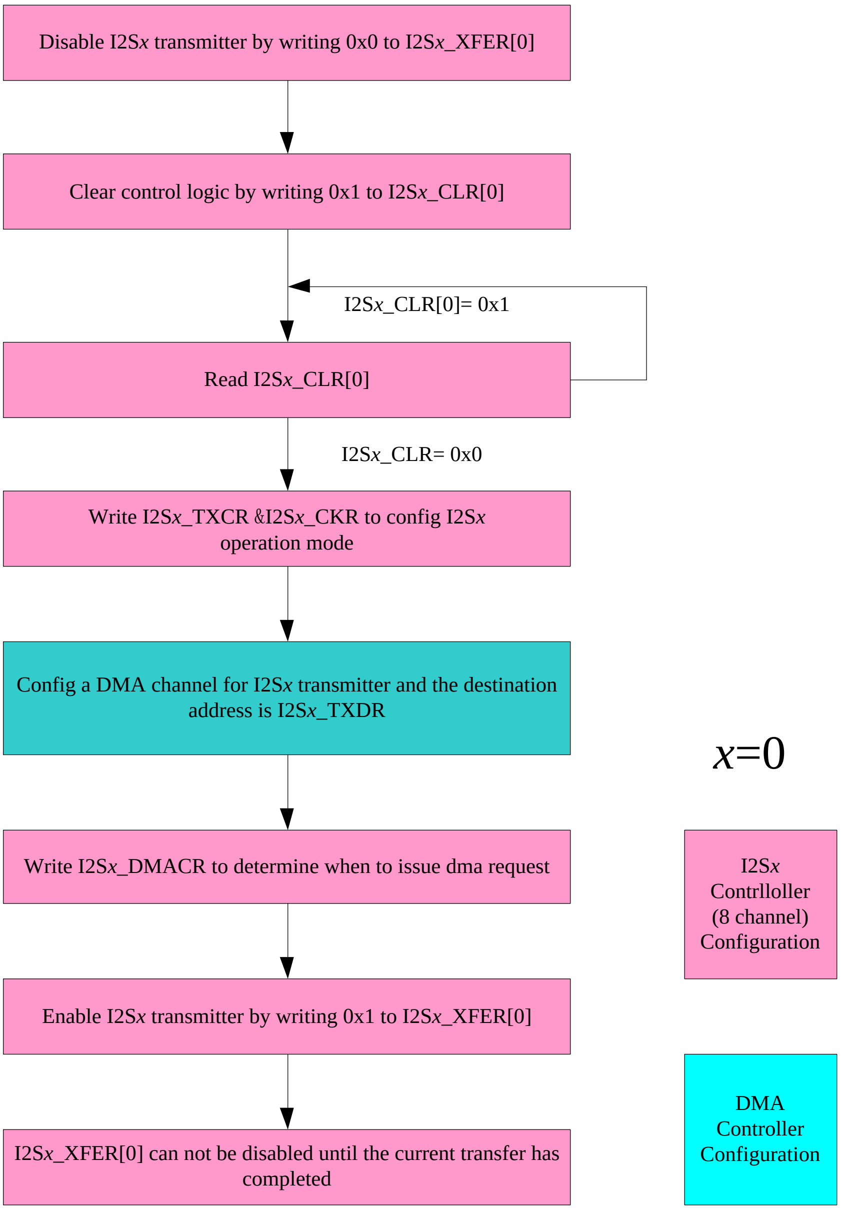

Disable I2Sx transmitter by writing 0x0 to I2Sx_XFER[0]
Clear control logic by writing 0x1 to I2Sx_CLR[0]
I2Sx_CLR[0]= 0x1
Read I2Sx_CLR[0]
I2Sx_CLR= 0x0
Write I2Sx_TXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx transmitter and the destination address is I2Sx_TXDR
Write I2Sx_DMACR to determine when to issue dma request
Enable I2Sx transmitter by writing 0x1 to I2Sx_XFER[0]
I2Sx_XFER[0] can not be disabled until the current transfer has completed
x=0
I2Sx Contrlloller (8 channel) Configuration
DMA Controller Configuration

Fig. 19-11 I2S/PCM controller (8 channel) transmit operation flow chart

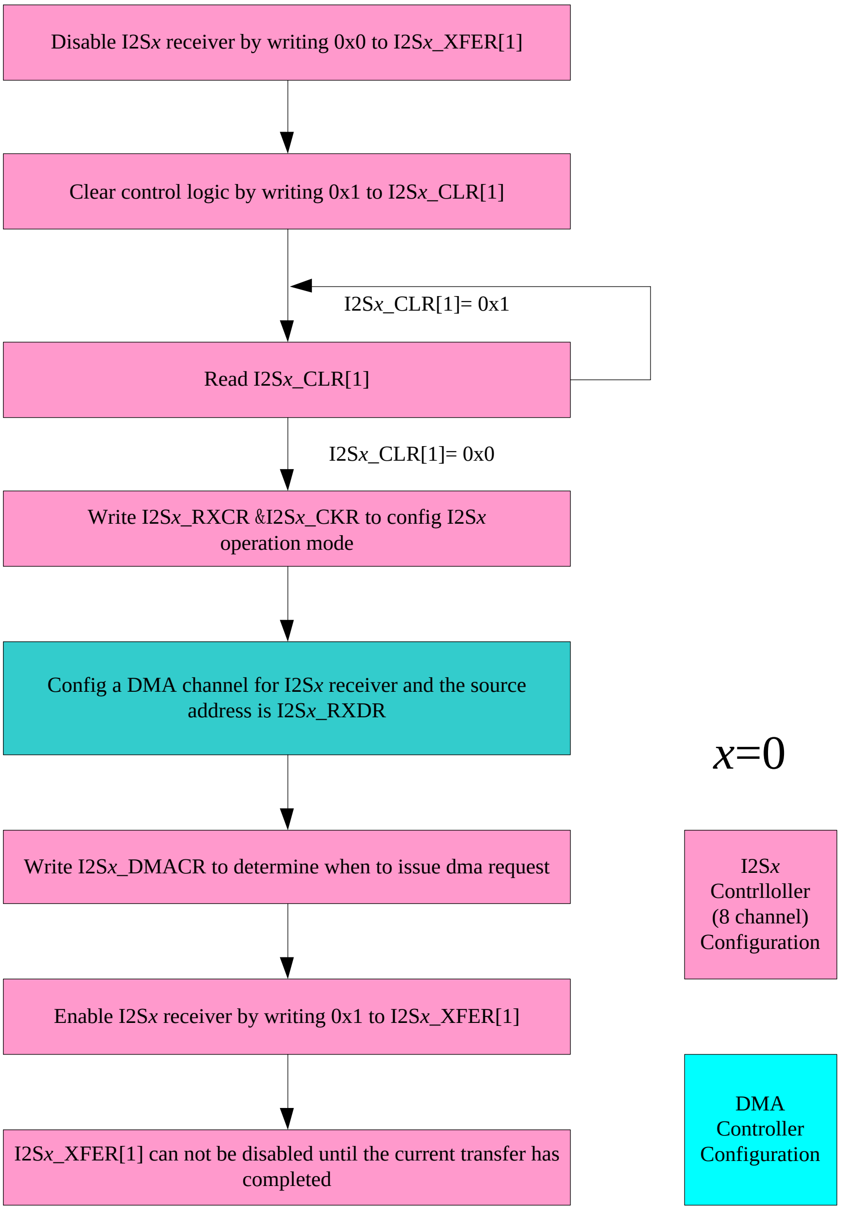

Disable I2Sx receiver by writing 0x0 to I2Sx_XFER[1]
Clear control logic by writing 0x1 to I2Sx_CLR[1]
I2Sx_CLR[1]= 0x1
Read I2Sx_CLR[1]
I2Sx_CLR[1]= 0x0
Write I2Sx_RXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx receiver and the source address is I2Sx_RXDR
x=0
Write I2Sx_DMACR to determine when to issue dma request
I2Sx Contrlloller (8 channel) Configuration
Enable I2Sx receiver by writing 0x1 to I2Sx_XFER[1]
DMA Controller Configuration
I2Sx_XFER[1] can not be disabled until the current transfer has completed

Fig. 19-12 I2S/PCM controller (8 channel) receive operation flow chart

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 Microwire 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

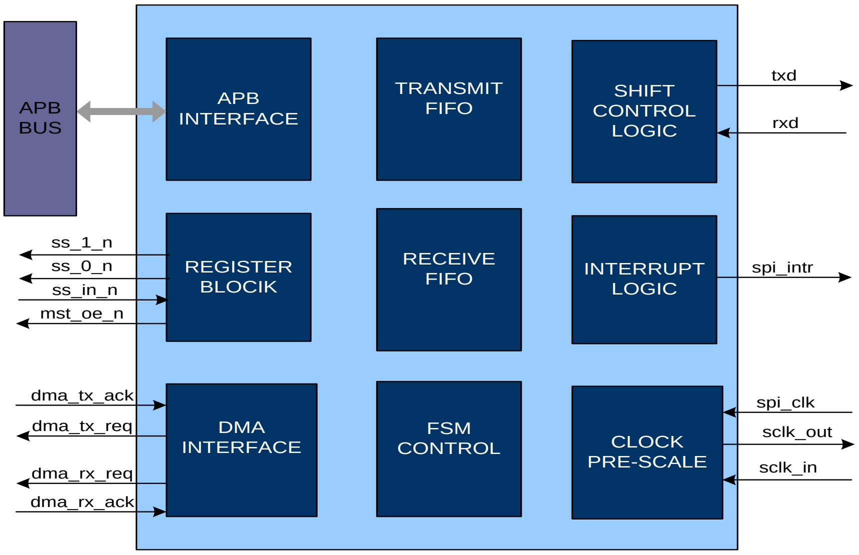


Fig. 20-1 SPI Controller Block diagram

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 8, 16, and 32 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 serialdevice 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 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.

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

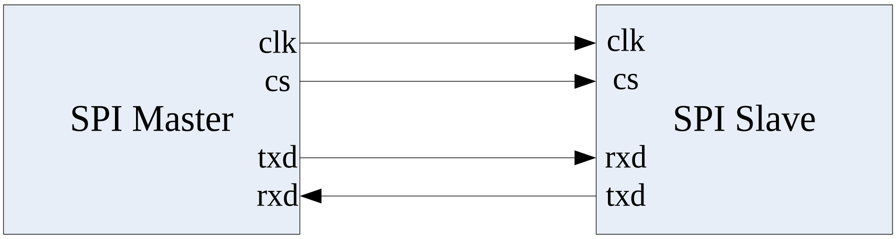


Fig. 20-2 SPI Master and Slave Interconnection

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.

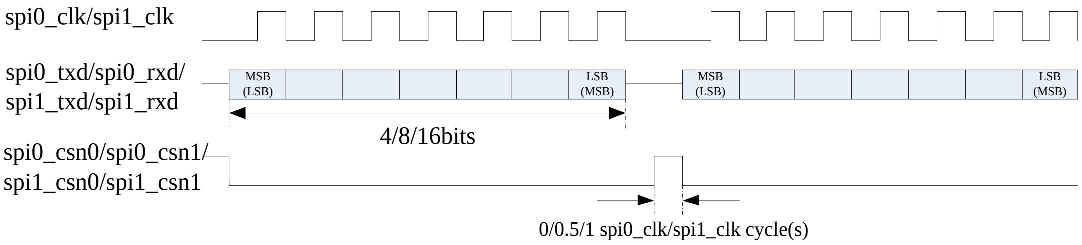


Fig. 20-3 SPI Format (SCPH=0 SCPOL=0)

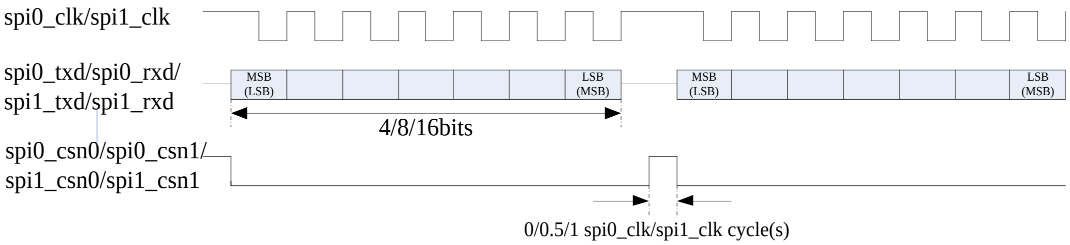


Fig. 20-4 SPI Format (SCPH=0 SCPOL=1)

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.

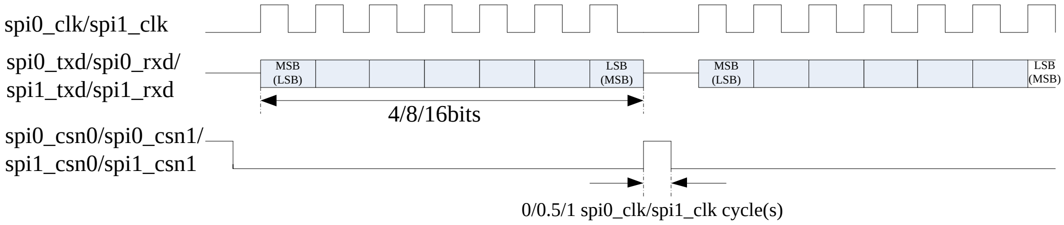


Fig. 20-5 SPI Format (SCPH=1 SCPOL=0)

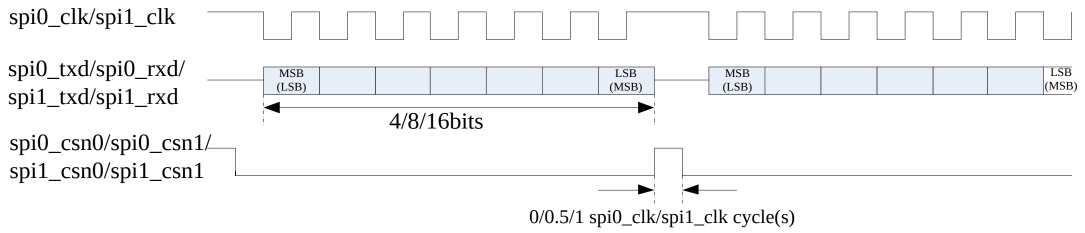


Fig. 20-6 SPI Format (SCPH=1 SCPOL=1)

20.4 Register Description

This section describes the control/status registers of the design. Pay attention that there are two SPI controllers in the chip: spi0 & spi1, so the base address in the following register descriptions can be either spi0 or spi1 base address.

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable
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	0x0400~ 0x07fc	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800~ 0x0bfc	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
17:16	RW	0x0	FRF Frame Format 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11 : Reserved
15:14	RW	0x0	RSD Rxd Sample Delay When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00:do not delay 2'b01:1 cycle delay 2'b10:2 cycles delay 2'b11:3 cycles delay
13	RW	0x0	BHT Byte and Halfword Transform Valid when data frame size is 8bit. 1'b0: apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM First Bit Mode 1'b0: first bit is MSB 1'b1: first bit is LSB

Bit	Attr	Reset Value	Description
11	RW	0x0	EM Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1’b0: little endian 1’b1: big endian
10	RW	0x0	SSD ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 1’b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1’b1: the period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0x0	CSM Chip Select Mode Valid when the frame format is set to Motorola SPI and SPI used as a master. 2’b00: ss_n keep low after every frame data is transferred. 2’b01: ss_n be high for half sclk_out cycles after every frame data is transferred. 2’b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2’b11: reserved
7	RW	0x0	SCPOL Serial Clock Polarity Valid when the frame format is set to Motorola SPI. 1’b0: Inactive state of serial clock is low 1’b1: Inactive state of serial clock is high
6	RW	0x0	SCPH Serial Clock Phase Valid when the frame format is set to Motorola SPI. 1’b0: Serial clock toggles in middle of first data bit 1’b1: Serial clock toggles at start of first data bit

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CFS Control Frame Size Selects the length of the control word for the Microwire frame format. 4’b0000~4’b0010: 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 Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER
Address: Operational Base + offset (0x000c)
Slave Enable Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	SER Slave Select Enable This register is valid only when SPI is configured as a master device.

SPI_BAUDR
Address: Operational Base + offset (0x0010)
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

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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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 (0x0400~0x07fc)
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 (0x0800~0x0bfc)
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 SPI interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_clk	I/O	IO_SPI0clk_TS0data4_UART4EXPctsn_BBgpio5b4	GRF_GPIO5B_IOMUX[9:8]=01
spi0_csn0	I/O	IO_SPI0csn0_TS0data5_UART4EXPrtsn_BBgpio5b5	GRF_GPIO5B_IOMUX[11:10]=01
spi0_txd	O	IO_SPI0txd_TS0data6_UART4EXPsout_BBgpio5b6	GRF_GPIO5B_IOMUX[13:12]=01
spi0_rxd	I	IO_SPI0rxd_TS0data7_UART4EXPsin_BBgpio5b7	GRF_GPIO5B_IOMUX[15:14]=01
spi0_csn1	O	IO_SPI0csn1_TS0sync_BBgpio5c0	GRF_GPIO5C_IOMUX[1:0]=01
spi1_clk	I/O	IO_ISPshutteren_SPI1clk_GPIO30gpio7b4	GRF_GPIO7B_IOMUX[9:8]=10
spi1_csn0	I/O	IO_ISPflashtrigout_SPI1csn0_GPIO30gpio7b5	GRF_GPIO7B_IOMUX[11:10]=10
spi1_rxd	O	IO_ISPprelighttrig_SPI1rxd_GPIO30gpio7b6	GRF_GPIO7B_IOMUX[13:12]=10
spi1_txd	I	IO_ISPshuttertrig_SPI1txd_GPIO30gpio7b7	GRF_GPIO7B_IOMUX[15:14]=10
spi2_clk	I/O	IO_SPI2clk_SCio_GPIO1830gpio8a6	GRF_GPIO8A_IOMUX[13:12]=01
spi2_csn0	I/O	IO_SPI2csn0_SCdetect_GPIO1830gpio8a7	GRF_GPIO8A_IOMUX[15:14]=01
spi2_rxd	I	IO_SPI2rxd_SCrst_GPIO1830gpio8b0	GRF_GPIO8B_IOMUX[1:0]=01
spi2_txd	O	IO_SPI2txd_SCclk_GPIO1830gpio8b1	GRF_GPIO8B_IOMUX[3:2]=01
spi2_csn1	O	IO_SPI2csn1_SCiot1_GPIO1830gpio8a3	GRF_GPIO8A_IOMUX[7:6]=01

Note: spi0_csn1, spi1_csn1,spi2_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 F_{spi_clk}>= 2 × (maximum F_{sclk_out})
When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_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.

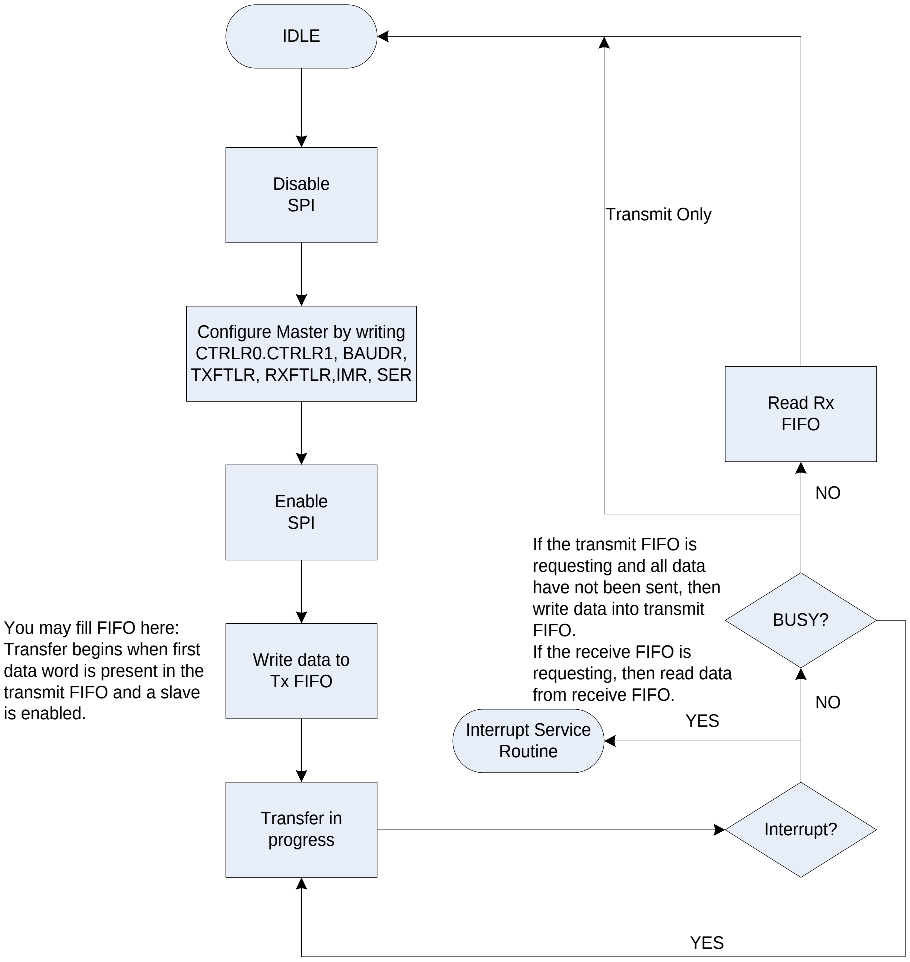

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, BAUDR, TXFTLR, RXFTLR,IMR, SER
Enable SPI
Write data to Tx FIFO
Transfer in progress
You may fill FIFO here: Transfer begins when first data word is present in the transmit FIFO and a slave is enabled.
Transmit Only
Read Rx FIFO
NO
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.
BUSY?
NO
YES
Interrupt Service Routine
Interrupt?
YES

Fig. 20-7 SPI Master transfer flow diagram

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

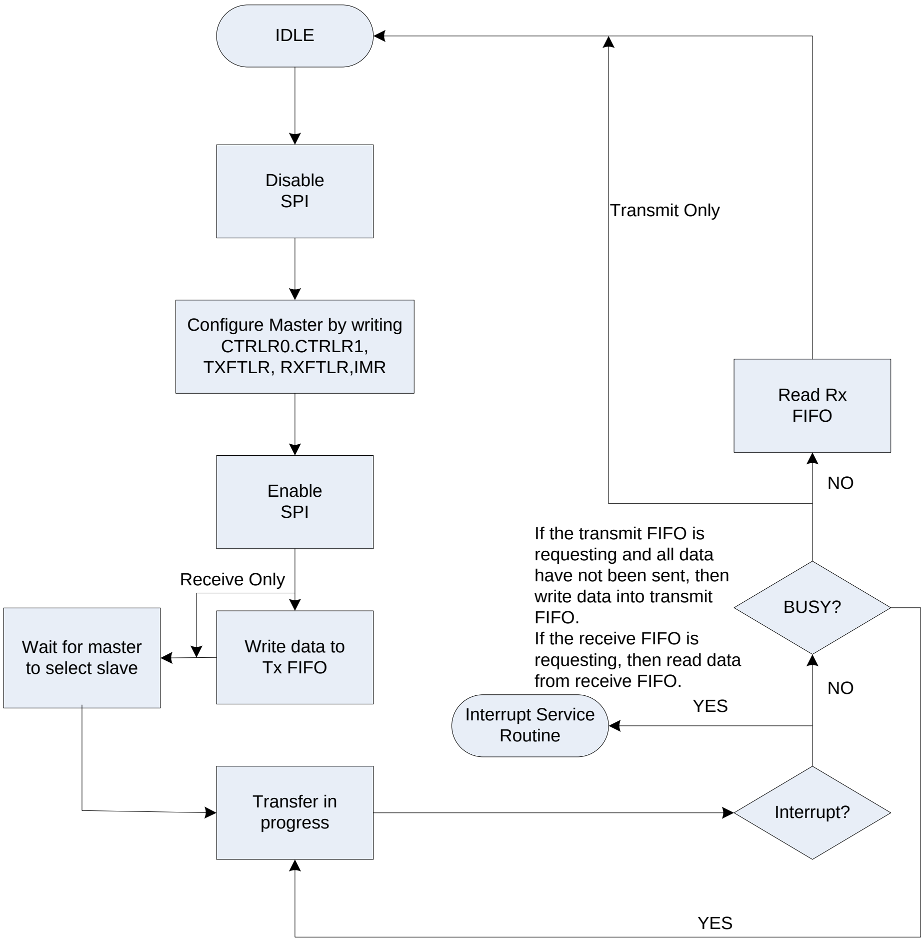

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, TXFTLR, RXFTLR,IMR
Enable SPI
Receive Only
Wait for master to select slave
Write data to Tx FIFO
Transfer in progress
Interrupt?
YES
Interrupt Service Routine
NO
BUSY?
NO
Read Rx FIFO
Transmit Only
YES
If the transmit FIFO is requesting and all data have not been sent, then write data into transmit FIFO.
If the receive FIFO is requesting, then read data from receive FIFO.

Fig. 20-8 SPI Slave transfer flow diagram

Chapter 21 SPDIF Transmitter

21.1 Overview

The SPDIF transmitter is a self-clocking, serial, unidirectional interface for the interconnection of digital audio equipment for consumer and professional applications, using linear PCM coded audio samples.

It provides the basic structure of the interface. Separate documents define items specific to particular applications.

When used in a professional application, the interface is primarily intended to carry monophonic or stereophonic programmes, at a 48 kHz sampling frequency and 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 a 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 as 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 format specification for these applications is not part of this standard.

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
- Support to carry signals sample at max 192khz

21.2 Block Diagram

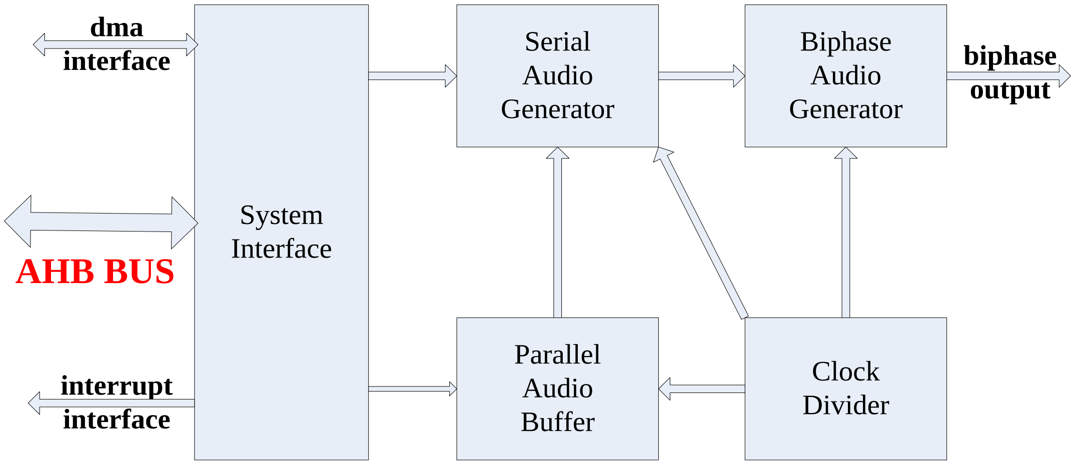


Fig. 21-1 SPDIF transmitter Block Diagram

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, and 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.

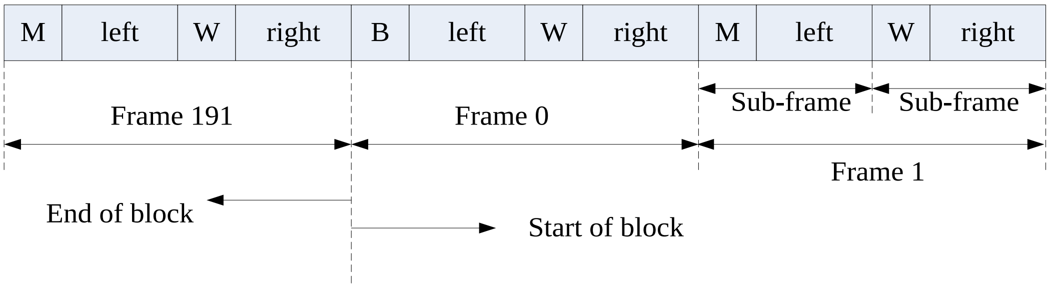


Fig. 21-2 SPDIF Frame Format

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

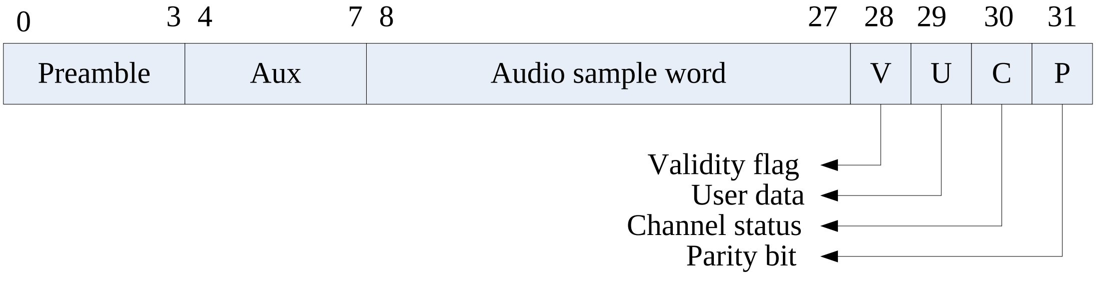


Fig. 21-3 SPDIF Sub-frame Format

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

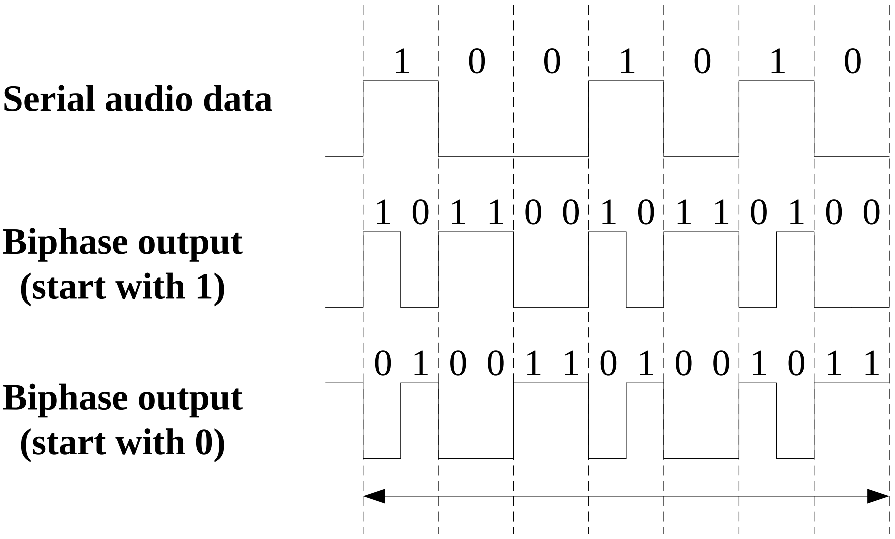


Fig. 21-4 SPDIF Channel Coding

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.

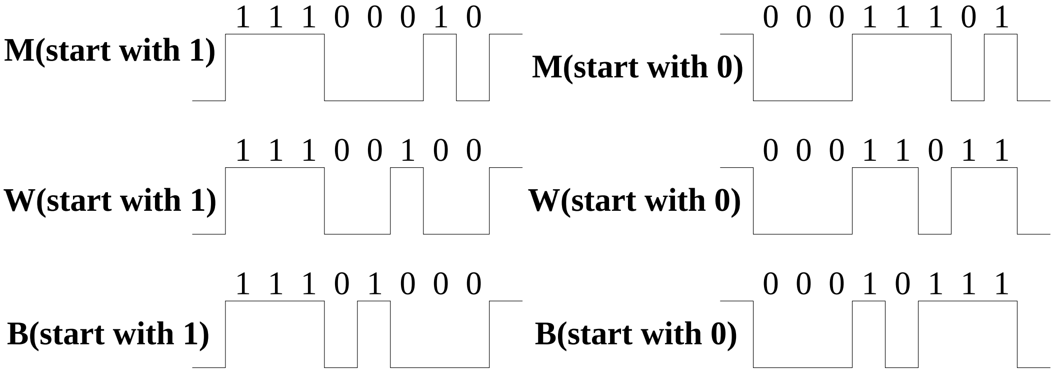


Fig. 21-5 SPDIF Preamble

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 burstpayload 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 burstpayload, 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 time slot 27 and the LSB is placed into time slot 12.

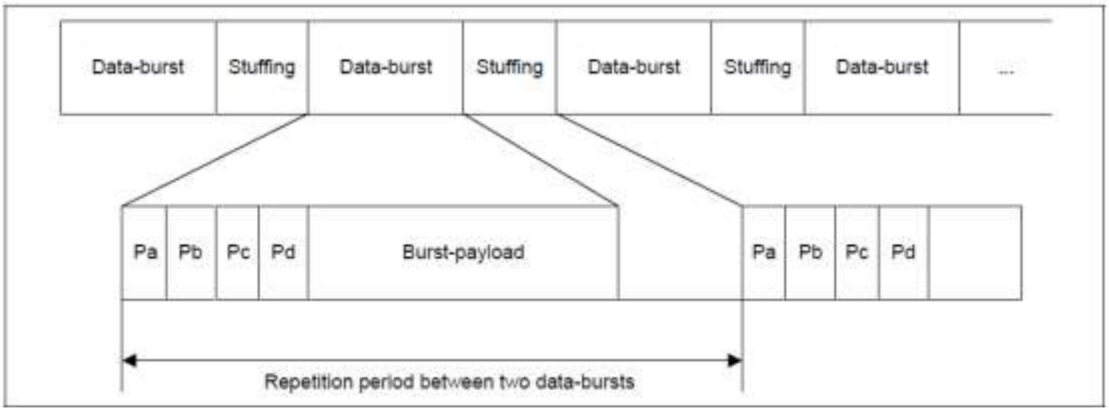


Fig. 21-6 Format of Data-burst

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

Name	Offset	Size	Reset Value	Description
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 caculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:10	RO	0x0	reserved
9	RW	0x0	PRE_CHANGE Preamble Change The bit only is valid when set to non-linear PCM mode; 0: the Preamble will change when block finish; 1: the Preamble will change every 192 frames just like the linear PCM mode.
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

Bit	Attr	Reset Value	Description
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
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

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 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 Ueser 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
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

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 (0x0100)
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 birstream number is 0.

Bit	Attr	Reset Value	Description
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
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 IOMUX Setting

Module Pin	IO	Pad Name	IOMUX Setting
spdif_tx	O	SPDIFtx_AUDIOgpio6b3	GPIO6B_IOMUX[7:6]= 2'b01

Notes: 1. O=output

21.6 Application Notes

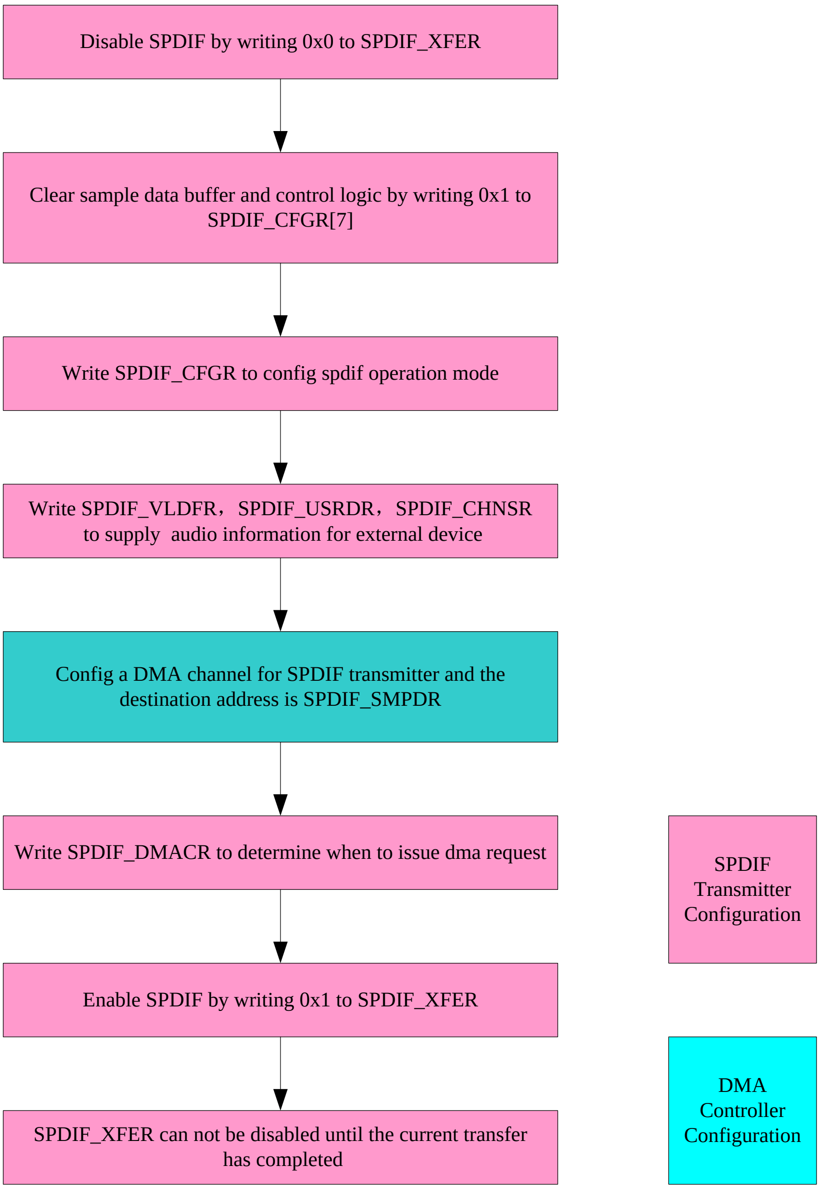

Disable SPDIF by writing 0x0 to SPDIF_XFER
Clear sample data buffer and control logic by writing 0x1 to SPDIF_CFGR[7]
Write SPDIF_CFGR to config spdif operation mode
Write SPDIF_VLDFR，SPDIF_USRDR，SPDIF_CHNSR to supply audio information for external device
Config a DMA channel for SPDIF transmitter and the destination address is SPDIF_SMPDR
Write SPDIF_DMACR to determine when to issue dma request
Enable SPDIF by writing 0x1 to SPDIF_XFER
SPDIF_XFER can not be disabled until the current transfer has completed
SPDIF Transmitter Configuration
DMA Controller Configuration

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 terminate 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 terminate 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 calculation the new repetition value if the definition of the repetition is different for some audio formats such as AC-3.

Chapter 22 Transport Stream Processing(TSP)

22.1 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.

TPS supports the following features:

- Supports two TS input channels and one TS output channel
- 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 serial and parallel output mode with PCR adjustment, and lsb-msb or msb-lsb bit ordering can be chosen in the serial output mode
- Supports 2 TS sources: demodulators and local memory
- Supports 2 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
- DMAC supports:
- Word alignment transfer
- Fixed and incrementing addressing
- Word size transfer
- burst modes: Incr4, Incr8, Inc16; burst transfer will be done with INCR mode if the remaining data or address space is not capable to perform a complete burst transfer
- Hardware/software trigger mode
- LLP(List Link Programming) Mode
- DMA done and error interrupt for each PTI channel
- 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

22.2 Block Diagram

The TSP comprises of following components:

- AMBA AHB slave interface
- Register block
- PTI
- DMAC
- TS Out Interface

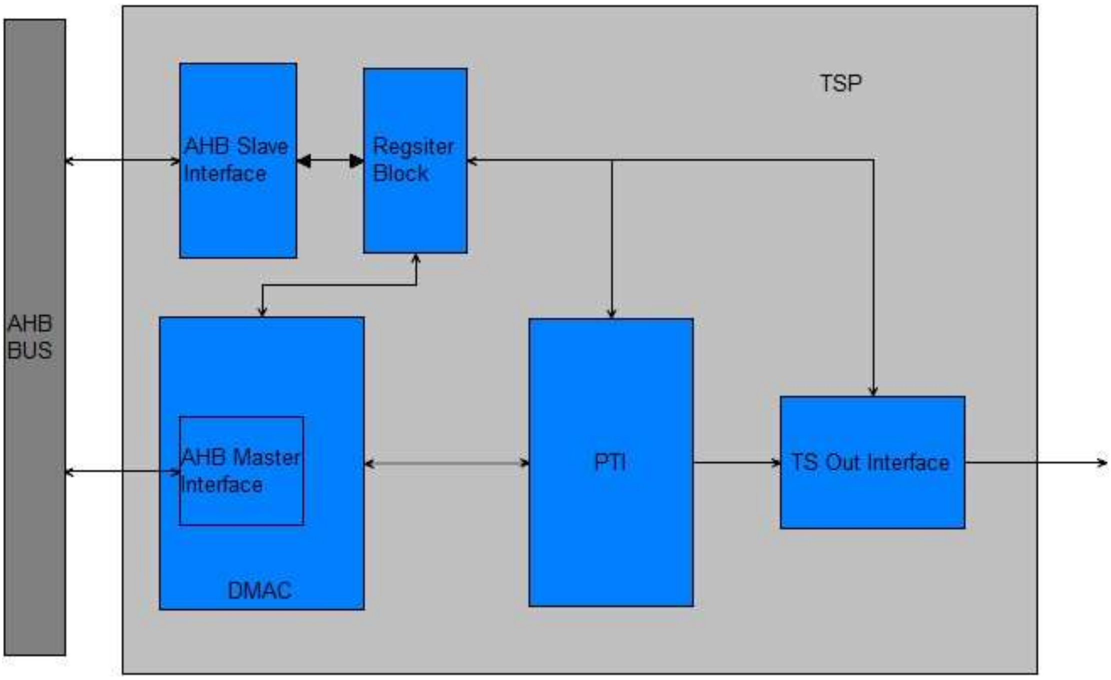


Fig. 22-1 TSP architecture

AHB Slave INTERFACE

The host processor can get access to the 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.

TS Out Interface

TS out interface can output either PID-filtered or non-PID-filtered TS packets from one PTI channel in a certain stream mode as configured. The TS receiver conforms to the stream mode to receive the TS packets.

DMAC

The DMAC performs all DMA transfers which get access to memory.

22.3 Function Description

22.3.1 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.

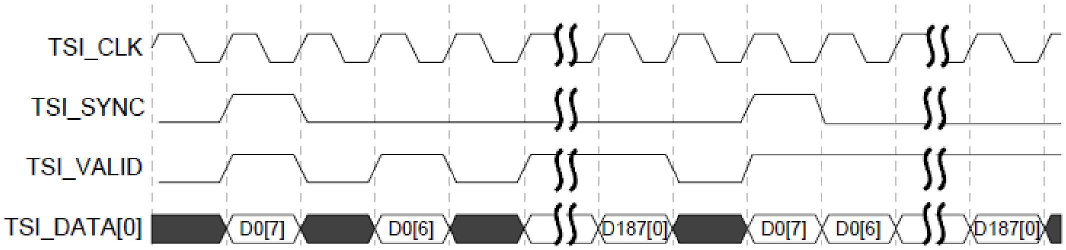


Fig. 22-2 Sync/Valid Serial Mode with Msb-Lsb Bit Ordering

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.

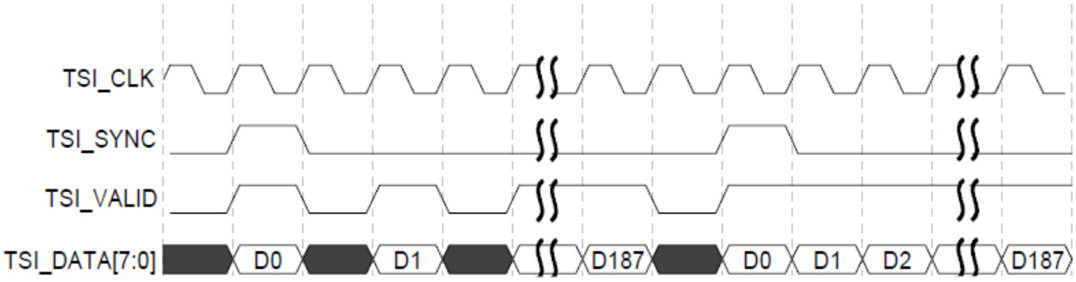


Fig. 22-3 Sync/valid Parallel Mode

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.

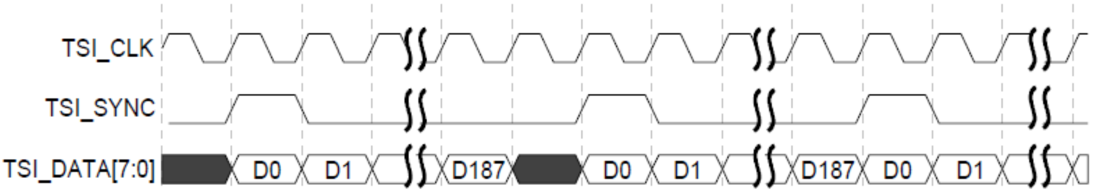


Fig. 22-4 Sync/Burst Parallel Mode

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.

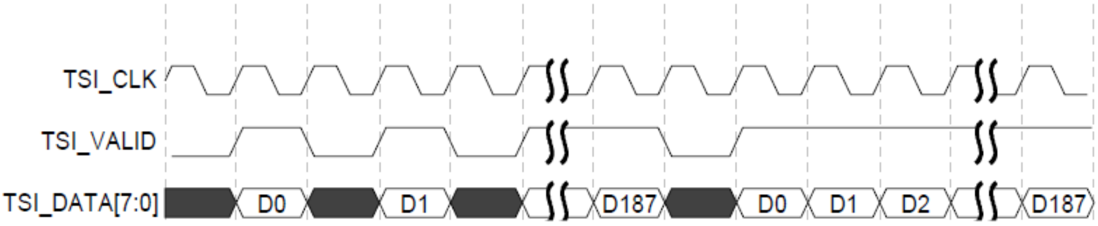


Fig. 22-5 Nosync/Valid Parallel Mode

When active high, TSI_VALID implies a valid byte of a TS packet.

22.3.2 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 field applicable to the program being decoded.

System clock is 27Mhz.

22.3.3 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.

22.4 Register Description

22.4.1 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_ADDR	0x000c	W	0x00000000	PVR DMA transaction starting address
TSP_PVR_INT_STS	0x0010	W	0x00000000	PVR DMA Interrupt Status Register
TSP_PVR_INT_ENA	0x0014	W	0x00000000	DMA Interrupt Enable Register
TSP_TSOUT_CTRL	0x0018	W	0x00000000	TS Out Control Register
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

Name	Offset	Size	Reset Value	Description
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
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

Name	Offset	Size	Reset Value	Description
TSP_PTIx_PIDn_FILT_2	0x090c	W	0x00000000	Fliter Word 2
TSP_PTIx_PIDn_FILT_3	0x0910	W	0x00000000	Fliter Word 3

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

22.4.2 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 ;
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

Bit	Attr	Reset Value	Description

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/WSC	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/WSC	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_ADDR
Address: Operational Base + offset (0x000c)
PVR DMA transaction starting address

Bit	Attr	Reset Value	Description

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
31:2	RO	0x0	reserved
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
31:2	RO	0x0	reserved
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 ;

Bit	Attr	Reset Value	Description
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;

0	RW	0x0	start TS out start 1: to start TS out function ; 0: to stop TS out function;

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 ;

Bit	Attr	Reset Value	Description
8	RW	0x0	out_byteswap Output byteswap function When enabled, the word to be transferred to memory buffer "B4B3B2B1" is performed byteswapping to "B1B2B3B4".
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/WSC	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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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
12	W1C	0x0	pid44_done PID44 Channel Status 1 means done
11	W1C	0x0	pid43_done PID43 Channel Status 1 means done
10	W1C	0x0	pid42_done PID42 Channel Status 1 means done
9	W1C	0x0	pid41_done PID41 Channel Status 1 means done
8	W1C	0x0	pid40_done PID40 Channel Status 1 means done
7	W1C	0x0	pid39_done PID39 Channel Status 1 means done
6	W1C	0x0	pid38_done PID38 Channel Status 1 means done
5	W1C	0x0	pid37_done PID37 Channel Status 1 means done
4	W1C	0x0	pid36_done PID36 Channel Status 1 means done
3	RW	0x0	pid35_done PID35 Channel Status 1 means done
2	W1C	0x0	pid34_done PID34 Channel Status 1 means done
1	W1C	0x0	pid33_done PID33 Channel Status 1 means done

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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
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

Bit	Attr	Reset Value	Description
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
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
30	W1C	0x0	pid62_error PID62 Error Interrupt Status
29	W1C	0x0	pid61_error PID61 Error Interrupt Status
28	W1C	0x0	pid60_error PID60 Error Interrupt Status
27	W1C	0x0	pid59_error PID59 Error Interrupt Status
26	W1C	0x0	pid58_error PID58 Error Interrupt Status
25	W1C	0x0	pid57_error PID57 Error Interrupt Status
24	W1C	0x0	pid56_error PID56 Error Interrupt Status
23	W1C	0x0	pid55_error PID55 Error Interrupt Status
22	W1C	0x0	pid54_error PID54 Error Interrupt Status
21	W1C	0x0	pid53_error PID53 Error Interrupt Status
20	W1C	0x0	pid52_error PID52 Error Interrupt Status

Bit	Attr	Reset Value	Description
19	W1C	0x0	pid51_error PID51 Error Interrupt Status
18	W1C	0x0	pid50_error PID50 Error Interrupt Status
17	W1C	0x0	pid49_error PID49 Error Interrupt Status
16	W1C	0x0	pid48_error PID48 Error Interrupt Status
15	W1C	0x0	pid47_error PID47 Error Interrupt Status
14	W1C	0x0	pid46_error PID46 Error Interrupt Status
13	W1C	0x0	pid45_error PID45 Error Interrupt Status
12	W1C	0x0	pid44_error PID44 Error Interrupt Status
11	W1C	0x0	pid43_error PID43 Error Interrupt Status
10	W1C	0x0	pid42_error PID42 Error Interrupt Status
9	W1C	0x0	pid41_error PID41 Error Interrupt Status
8	W1C	0x0	pid40_error PID40 Error Interrupt Status
7	W1C	0x0	pid39_error PID39 Error Interrupt Status
6	W1C	0x0	pid38_error PID38 Error Interrupt Status
5	W1C	0x0	pid37_error PID37 Error Interrupt Status
4	W1C	0x0	pid36_error PID36 Error Interrupt Status
3	W1C	0x0	pid35_error PID35 Error Interrupt Status
2	W1C	0x0	pid34_error PID34 Error Interrupt Status
1	W1C	0x0	pid33_error PID33 Error Interrupt Status
0	W1C	0x0	pid32_error PID32 Error Interrupt Status

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
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

Bit	Attr	Reset Value	Description
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
17	RW	0x0	pid17_done_ena PID17 Done Enable
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

Bit	Attr	Reset Value	Description
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
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

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
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

Bit	Attr	Reset Value	Description
20	RW	0x0	pid52_done PID52 Done Enable 1:enabled 0:disabled
19	RW	0x0	pid51_done PID51 Done Enable 1:enabled 0:disabled
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

Bit	Attr	Reset Value	Description
9	RW	0x0	pid41_done PID41 Done Enable 1:enabled 0:disabled
8	RW	0x0	pid40_done PID40 Done Enable 1:enabled 0:disabled
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

Bit	Attr	Reset Value	Description
31	RW	0x0	pid31_error PID31 Error Interrupt Enable 1:enabled 0:disabled
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

Bit	Attr	Reset Value	Description
20	RW	0x0	pid20_error PID20 Error Interrupt Enable 1:enabled 0:disabled
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

Bit	Attr	Reset Value	Description
9	RW	0x0	pid9_error PID9 Error Interrupt Enable 1:enabled 0:disabled
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

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

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

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

Bit	Attr	Reset Value	Description
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/WSC	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/WSC	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

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

22.5 Interface Description

Table 22-1 TSP Interface Description

Module Pin	IO	Pad Name	IOMUX Setting
ts_data0	I/O	IO_UART1BBsin_TS0data0_BBgpio5b0	GPIO5B_IOMUX[1:0]= 2'b10
ts_data1	I/O	IO_UART1BBsout_TS0data1_BBgpio5b1	GPIO5B_IOMUX[3:2]= 2'b10
ts_data2	I/O	IO_UART1BBctsn_TS0data2_BBgpio5b2	GPIO5B_IOMUX[5:4]= 2'b10
ts_data3	I/O	IO_UART1BBrtsn_TS0data3_BBgpio5b3	GPIO5B_IOMUX[7:6]= 2'b10
ts_data4	I/O	IO_SPI0clk_TS0data4_UART4EXPctsn_BBgpio5b4	GPIO5B_IOMUX[9:8]= 2'b10

Module Pin	IO	Pad Name	IOMUX Setting
ts_data5	I/O	IO_SPI0csn0_TS0data5_UART4EXPrtsn_BBgpio5b5	GPIO5C_IOMUX[11:10]= 2'b10
ts_data6	I/O	IO_SPI0txd_TS0data6_UART4EXPsout_BBgpio5b6	GPIO5B_IOMUX[13:12]= 2'b10
ts_data7	I/O	IO_SPI0rxd_TS0data7_UART4EXPsin_BBgpio5b7	GPIO5B_IOMUX[15:14]= 2'b10
ts_valid	I/O	IO_TS0valid_BBgpio5c1	GPIO5C_IOMUX[2]= 1'b1
ts_sync	I/O	IO_SPI0csn1_TS0sync_BBgpio5c0	GPIO5C_IOMUX[1:0]= 2'b10
ts_err	I/O	IO_TS0err_BBgpio5c3	GPIO5C_IOMUX[6]= 1'b1
ts_clk	I/O	IO_TS0clk_BBgpio5c2	GPIO5C_IOMUX[4]= 1'b1
hsadc_data0	I	IO_CIFdata2_HOSTdin0_HSADCdata0_DVPgpio2a0	GPIO2A_IOMUX[1:0]= 2'b11
hsadc_data1	I	IO_CIFdata3_HOSTdin1_HSADCdata1_DVPgpio2a1	GPIO2A_IOMUX[3:2]= 2'b11
hsadc_data2	I	IO_CIFdata4_HOSTdin2_HSADCdata2_DVPgpio2a2	GPIO2A_IOMUX[5:4]= 2'b11
hsadc_data3	I	IO_CIFdata5_HOSTdin3_HSADCdata3_DVPgpio2a3	GPIO2A_IOMUX[7:6]= 2'b11
hsadc_data4	I	IO_CIFdata6_HOSTckinp_HSADCdata4_DVPgpio2a4	GPIO2A_IOMUX[9:8]= 2'b11
hsadc_data5	I	IO_CIFdata7_HOSTckinn_HSADCdata5_DVPgpio2a5	GPIO2A_IOMUX[11:10]= 2'b11
hsadc_data6	I	IO_CIFdata8_HOSTdin4_HSADCdata6_DVPgpio2a6	GPIO2A_IOMUX[13:12]= 2'b11
hsadc_data7	I	IO_CIFdata9_HOSTdin5_HSADCdata7_DVPgpio2a7	GPIO2A_IOMUX[15:14]= 2'b11
hsadc_valid	I	IO_CIFhref_HOSTdin7_HSADCTSvalid_DVPgpio2b1	GPIO2B_IOMUX[3:2]= 2'b11
hsadc_sync	I	IO_CIFvsync_HOSTdin6_HSADCTSsync_DVPgpio2b0	GPIO2B_IOMUX[1:0]= 2'b11
hsadc_err	I	IO_CIFclkout_HOSTwkreq_HSADCTSfail_DVPgpio2b3	GPIO2B_IOMUX[7:6]= 2'b01
gps_clk	I	IO_CIFclkin_HOSTwkack_GPSclk_HSADCclkout_DVPgpio2b2	GPIO2B_IOMUX[5:4]= 2'b11
gpst1_clk	I	IO_UART3GPSctsn_GPSrfclk_GPST1clk_GPIO30gpio7b1	GPIO7B_IOMUX[3:2]= 2'b11

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

22.6 Application Notes

22.6.1 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.

Note: PTI1 addr = PTI0 addr + 0x1000;

22.6.2 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.

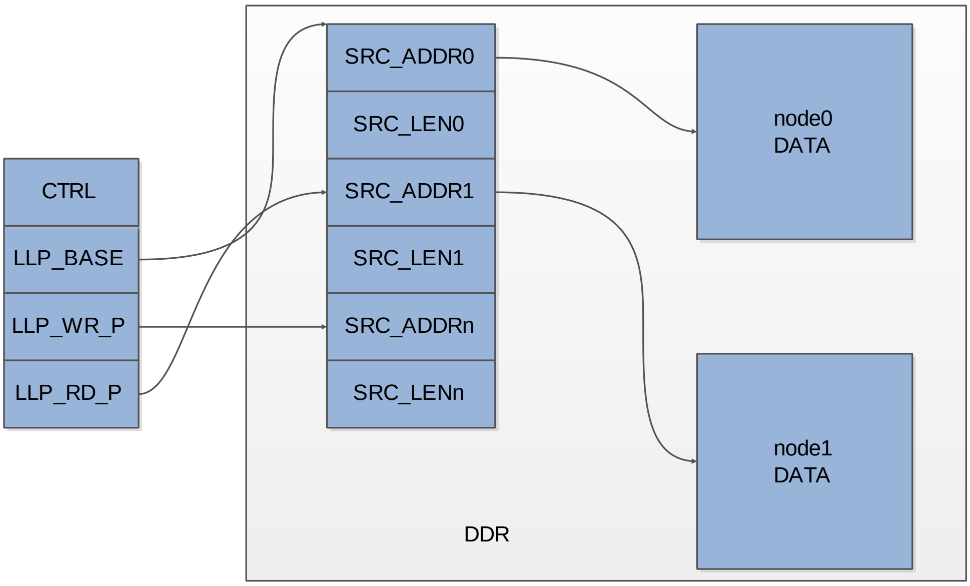
1.TS_IN Interface

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.



(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.*

22.6.3 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.

22.6.4 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.

22.6.5 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:

1. *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.*
2. *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.*

1.Demux data obtain

- **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.

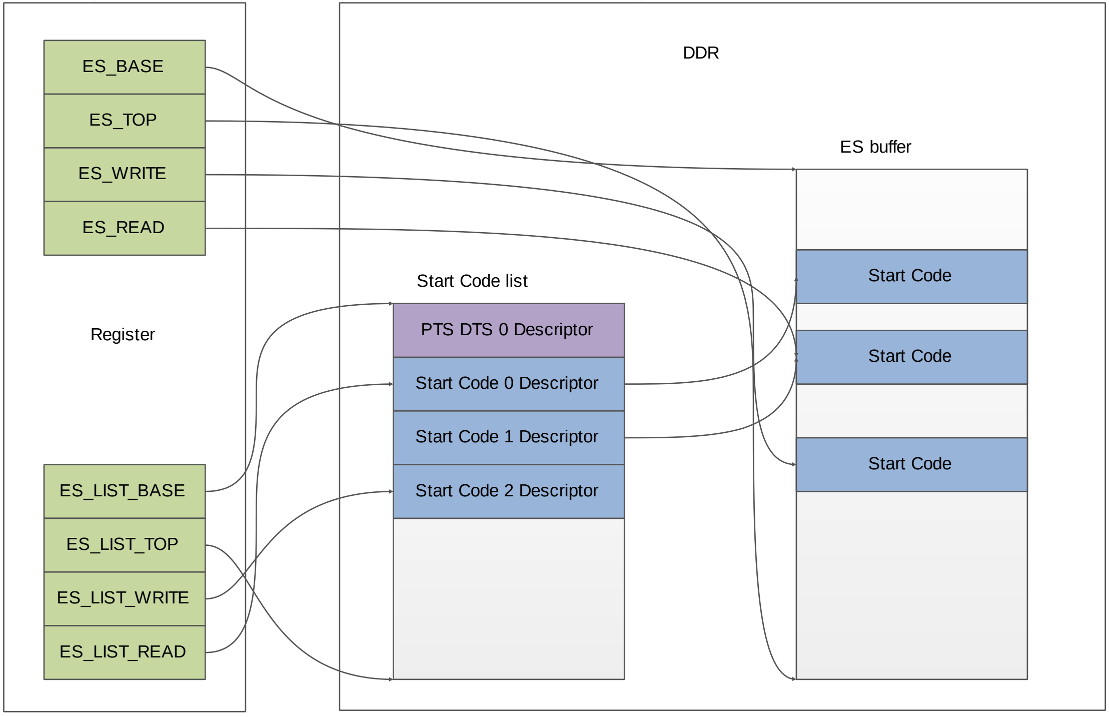
Section Length = {First Word[11:8], First Word[23:16]};
Total Length = Section Length;

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.



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:

(1) 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

(2) 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.

22.6.6 TS Out Interface

All the configuration is done by writing TSP_TSOUT_CTRL. Before programming this register, make sure that you have enabled the TS OUT interface. If you want to disable TS out interface, write '0' to the START bit(bit 0) of TSP_TSOUT_CTRL, and then disable it in the TSP_GFCG.
Each PTI channel can provide TS out interface with PID-filtering TS Packets or non-PID-filtering TS packets, and therefore there are totally 4 sources can be chosen for TS out interface.

22.6.7 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.

22.6.8 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 23 High-Speed ADC Interface (HSADC)

23.1 Overview

HS-ADC Interface Unit is an interface unit for connecting the TS interface and GPS ADC interface to AMBA AHB bus. It fetches the bus data received by the TS interface and GPS ADC interface and stores them to internal asynchronous FIFO after the ADC clock is active. The HS-ADC Interface Unit generates the DMA request signal when data length of the asynchronous FIFO over the almost full level or almost empty level.
HS-ADC supports the following features:

- Support Transport-Stream(TS) Interface with 8bits data bus
- Support GPS interface with 2bits or 4bits data bus
- Support combined interrupt output, source including: full interrupt, empty interrupt
- Support DMA transfer mode through generating DMA request from the event of almost full or almost empty, etc.
- Support two channel mode: single channel and dual channel
- Support the most significant bit negation or not
- Support sign bit extension
- Support two storage mode: input data are stored to high 8bit or 10bit and stored to low 8bit or 10bit of 16bit when pushed into FIFO.
- Support an asynchronous build-in FIFO with 128x64 size

23.2 Block Diagram

The HS-ADC diagram is as follows.

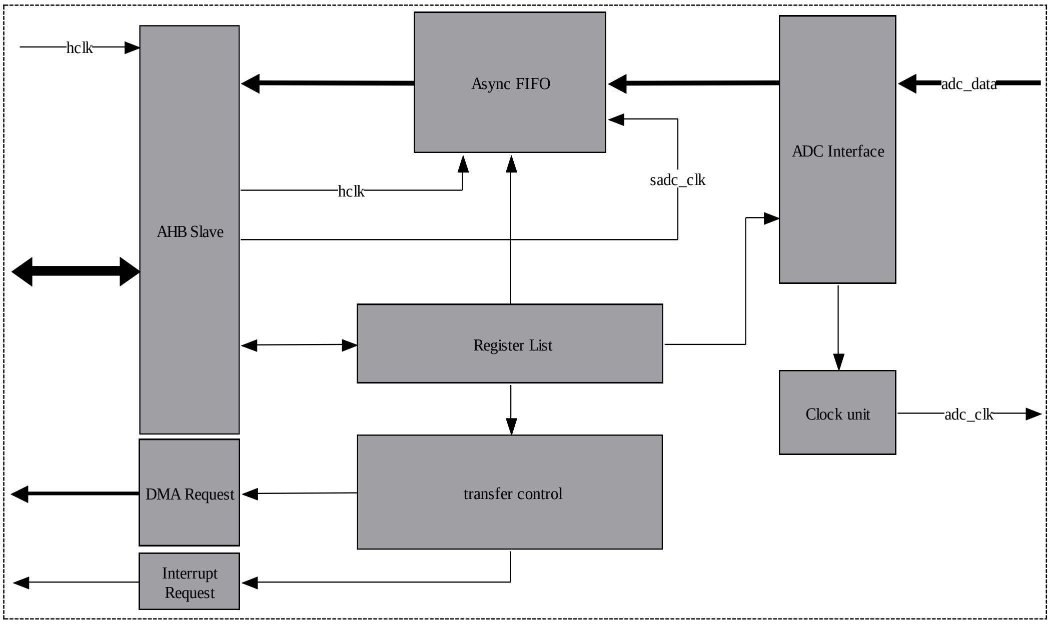


Fig. 23-1 HS-ADC Architecture

23.3 Function Description

This module can be configured for two interfaces: GPS interface, TS interface.

23.3.1 GPS interface

When this module is used as GPS interface, user should configure GRF register to select 2bits or 4bitsGPS data input and gps_clk as GPS clock input from pad.
Also, user should configure CRU_CLKSEL22_CON[5:4] to select gps_clk as HS-ADC controller working clock source.

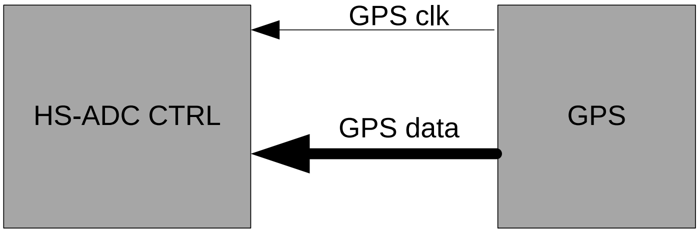


Fig. 23-2 GPS Application Diagram

23.3.2 TS interface

When this module is used as TS interface, user should configure GRF register to select 8bit TS data, ts_sync, ts_valid and ts_fail input and gps_clk input as TS clock input from pad.
Also, user should configure CRU_CLKSEL22_CON[5:4] to select gps_clk from pad as TS clock input.

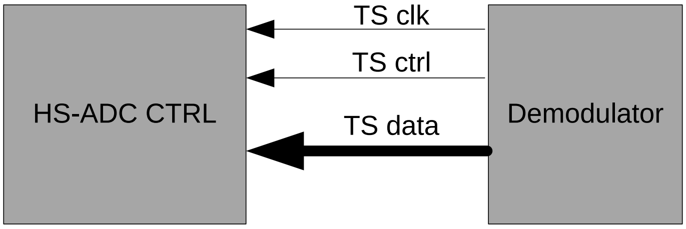


Fig. 23-3 TS Application Diagram

23.4 Register Description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
HSADC_CTRL	0x0000	W	0x00000000	Control register
HSADC_IER	0x0004	W	0x00000000	Interrupt control register
HSADC_ISR	0x0008	W	0x00000000	Interrupt status register
HSADC_TS_FAIL	0x000c	W	0x00000000	ts fail register
HSADC_CGCTL	0x0010	W	0x00000000	HSADC clock gating control
HSADC_DATA	0x0020	W	0x00000000	The data register of hsadc controller

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

23.4.2 Detail Register Description

HSADC_CTRL
Address: Operational Base + offset (0x0000)
Control register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:24	RW	0x0	almost_full_level Define almost full trigger level 0x0~"0xf" - configure valid range (Notes: 1 level indicate 4 entries data in the async FIFO. and this configure range mapping to 64 - 124 entries data in the async FIFO.)
23:20	RO	0x0	reserved

Bit	Attr	Reset Value	Description
19:16	RW	0x0	almost_empty_level Define almost empty trigger level 0x0~"0xf" - configure valid range (Notes: 1 level indicate 4 entries data in the async FIFO. and this configure range mapping to 0 - 60 entries data in the async FIFO.)
15	RW	0x0	gps_auto_gate_en GPS interface auto clock gating enable 1'b1: auto clock gating 1'b0: not auto clock gating
14	RW	0x0	ts_auto_gate_en TS interface auto clock gating enable 1'b1: auto clock gating 1'b0: not auto clock gating
13:12	RO	0x0	reserved
11	RW	0x0	gpsw GPS interface data width select 1'b0: 2bit data mode 1'b1: 4bit data mode
10	RW	0x0	ts_sync_en TS sync interface enable Field0000 Description
9	RW	0x0	ts_valid_en TS valid interface enable Enable ts interface "ts_valid" signal as data valid indicator 1'b0: disable 1'b1: enable
8	RW	0x0	ts_gps_sel MPEG-TS and GPS input select 1'b0: GPS is selected 1'b1: MPEG-TS is selected
7:6	RO	0x0	reserved
5	RW	0x0	dma_req_mode DMA request mode select 1'b1: almost full generate DMA request signal (Notes: this mode generate DMA request signal from almost full condition and cancel DMA request signal from almost empty condition. so you need configure two level by almost full level and almost empty level) 1'b0: almost empty generate DMA request signal (Notes: this mode generate DMA request signal from almost empty condition and that only once DMA request.)

Bit	Attr	Reset Value	Description
4:1	RO	0x0	reserved
0	RW	0x0	adc_en HS-ADC Interface Unit Enable Bit 1'b1: enable (Notes: will return 1 when the hardware started transfer) 1'b0: disable (Notes: other bit can be modify only the hardware return 0)

HSADC_IER
Address: Operational Base + offset (0x0004)
Interrupt control register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	int_empty_en Interrupt en/disable bit for the empty interrupt flag of async FIFO 1'b1: enable 1'b0: disable
0	RW	0x0	int_full_en Interrupt en/disable bit for the full interrupt flag of async FIFO 1'b1: enable 1'b0: disable

HSADC_ISR
Address: Operational Base + offset (0x0008)
Interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	int_empty_stat_ind Async FIFO empty interrupt flag 1'b1(R): This bit will be set to "1" when Async FIFO empty status and that only to read operation. 1'b0(W): Write "0" to bit for clear the interrupt flag and that only to wrtie operation.
0	RW	0x0	int_full_stat_ind Async FIFO full interrupt flag 1'b1(R): This bit will be set to "1" when Async FIFO full status and that only to read operation. 1'b0(W): Write "0" to bit for clear the interrupt flag and that only to wrtie operation.

HSADC_TS_FAIL
Address: Operational Base + offset (0x000c)
ts fail register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ts_fail_ahb TS stream fail indicator this signal only valid when select TS stream input(mpts=1) 1'b0: TS stream decode successfully 1'b1: TS stream decode fail

HSADC_CGCTL
Address: Operational Base + offset (0x0010)
HSADC Clock Gating control

Bit	Attr	Reset Value	Description
31:1	RW	0x00000000	cycle_cfg clock gated cycles configuration when configure cg_enable to 1 and cycle_cfg to non-zero value,HSADC clock will be gated for cycle_cfg cycles ,then clock recover.
0	RW	0x0	cg_enable clock gating enable control 1'b0: clock gating disable 1'b1: clock gating enable

HSADC_DATA
Address: Operational Base + offset (0x0020)
Data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DATA

23.5 Interface Description

23.5.1 TS mode

Table 23-1 IOMUX configuration in TS mode

Module Pin	IO	Pad Name	IOMUX Setting
hsadc_data0	I	IO_CIFdata2_HOSTdin0_HSADCdata0_DVPgpio2a0	GPIO2A_IOMUX[1:0]= 2'b11
hsadc_data1	I	IO_CIFdata3_HOSTdin1_HSADCdata1_DVPgpio2a1	GPIO2A_IOMUX[3:2]= 2'b11
hsadc_data2	I	IO_CIFdata4_HOSTdin2_HSADCdata2_DVPgpio2a2	GPIO2A_IOMUX[5:4]= 2'b11

Module Pin	IO	Pad Name	IOMUX Setting
hsadc_data3	I	IO_CIFdata5_HOSTdin3_HSADCdata3_DVPgpio2a3	GPIO2A_IOMUX[7:6]= 2'b11
hsadc_data4	I	IO_CIFdata6_HOSTckinp_HSADCdata4_DVPgpio2a4	GPIO2A_IOMUX[9:8]= 2'b11
hsadc_data5	I	IO_CIFdata7_HOSTckinn_HSADCdata5_DVPgpio2a5	GPIO2A_IOMUX[11:10]= 2'b11
hsadc_data6	I	IO_CIFdata8_HOSTdin4_HSADCdata6_DVPgpio2a6	GPIO2A_IOMUX[13:12]= 2'b11
hsadc_data7	I	IO_CIFdata9_HOSTdin5_HSADCdata7_DVPgpio2a7	GPIO2A_IOMUX[15:14]= 2'b11
hsadc_valid	I	IO_CIFhref_HOSTdin7_HSADCTSvalid_DVPgpio2b1	GPIO2B_IOMUX[3:2]= 2'b11
hsadc_sync	I	IO_CIFvsync_HOSTdin6_HSADCTSsync_DVPgpio2b0	GPIO2B_IOMUX[1:0]= 2'b11
hsadc_err	I	IO_CIFclkout_HOSTwkreq_HSADCTSfail_DVPgpio2b3	GPIO2B_IOMUX[7:6]= 2'b01
gps_clk	I	IO_CIFclkin_HOSTwkack_GPSclk_HSADCclkout_DVPgpio2b2	GPIO2B_IOMUX[5:4]= 2'b11
gpst1_clk	I	IO_UART3GPSctsn_GPSrfclk_GPST1clk_GPIO30gpio7b1	GPIO7B_IOMUX[3:2]= 2'b11

23.5.2 GPS mode

Table 23-2 IOMUX configuration in GPS mode

Module Pin	IO	Pad Name	IOMUX Setting
gps_clk	I	IO_CIFclkin_HOSTwkack_GPSclk_HSADCclkout_DVPgpio2b2	GPIO2B_IOMUX[5:4]= 2'b11
gpst1_clk	I	IO_UART3GPSctsn_GPSrfclk_GPST1clk_GPIO30gpio7b1	GPIO7B_IOMUX[3:2]= 2'b11
hsadc_data0	I	IO_CIFdata2_HOSTdin0_HSADCdata0_DVPgpio2a0	GPIO2A_IOMUX[1:0]= 2'b11
hsadc_data1	I	IO_CIFdata3_HOSTdin1_HSADCdata1_DVPgpio2a1	GPIO2A_IOMUX[3:2]= 2'b11
hsadc_data2	I	IO_CIFdata4_HOSTdin2_HSADCdata2_DVPgpio2a2	GPIO2A_IOMUX[5:4]= 2'b11
hsadc_data3	I	IO_CIFdata5_HOSTdin3_HSADCdata3_DVPgpio2a3	GPIO2A_IOMUX[7:6]= 2'b11

23.6 Application Notes

The following sections will describe the operation of DMA requests and DMA transfers.

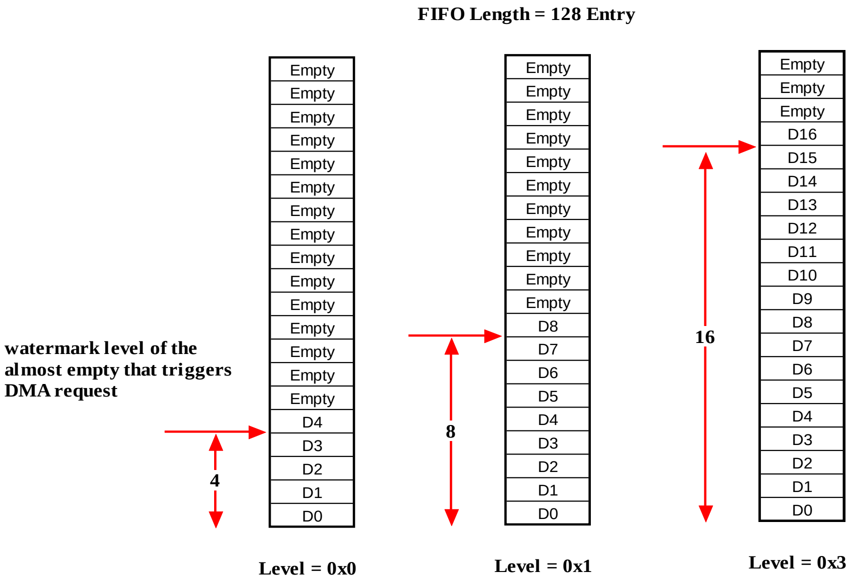


Fig. 23-4 Almost empty triggers a DMA request by DMA request mode

The DMA request signal will be generated from a watermark level trigger when data stored to FIFO over the watermark level of almost empty, where the watermark level can be configured through HSADC_CTRL[19:16] by software. This DMA request mode doesn't care the watermark level of almost full. The sample for watermark level configuration is shown in figure above.

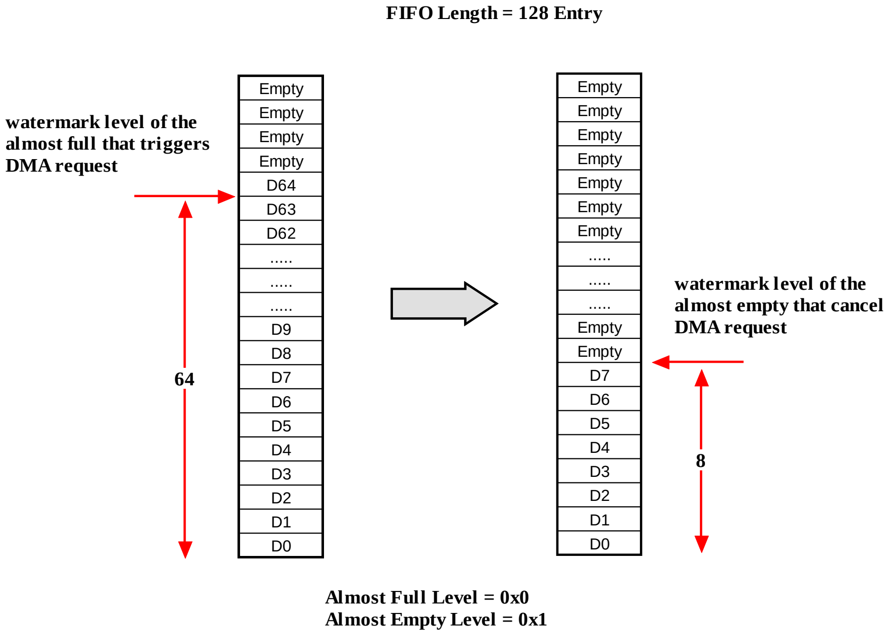

FIFO Length = 128 Entry
watermark level of the almost full that triggers DMA request
Empty
D64
D63
D62
.....
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
64
watermark level of the almost empty that cancel DMA request
8
Almost Full Level = 0x0
Almost Empty Level = 0x1

Fig. 23-5 Almost full triggers a DMA request by DMA request mode

The DMA request signal will be generated from a watermark level trigger when data stored to FIFO over the watermark level of almost full. It continues to generate request signal when the number of data in FIFO greater than watermark level of almost empty. This DMA request mode needs configure two watermark levels: watermark level of almost empty at the HSADC_CTRL[19:16] and watermark level of almost full at the HSADC_CTRL[27:24]. The sample for watermark level configuration is shown in figure above. When controller works in TS mode, the interface signal ts_sync should always be used.

Chapter 24 GMAC Ethernet Interface

24.1 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.

24.1.1 Features

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 deassertion 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.

24.2 Block Diagram

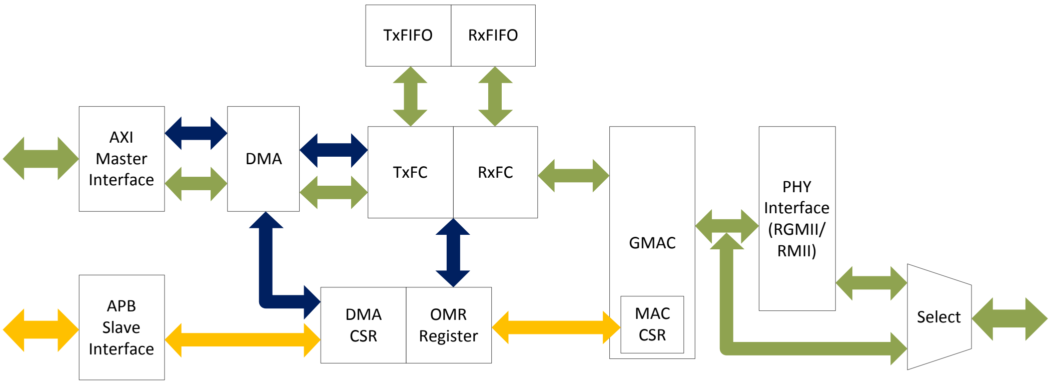


Fig. 24-1 GMAC architecture

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.

24.3 Function Description

24.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig 32-2.

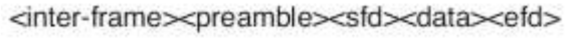

<inter-frame><preamble><sfd><data><efd>

Fig. 24-2 MAC Frame structure

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble. The bit value is 10101011.

The data in a well formed frame shall consist of N octets data.

24.3.2 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).

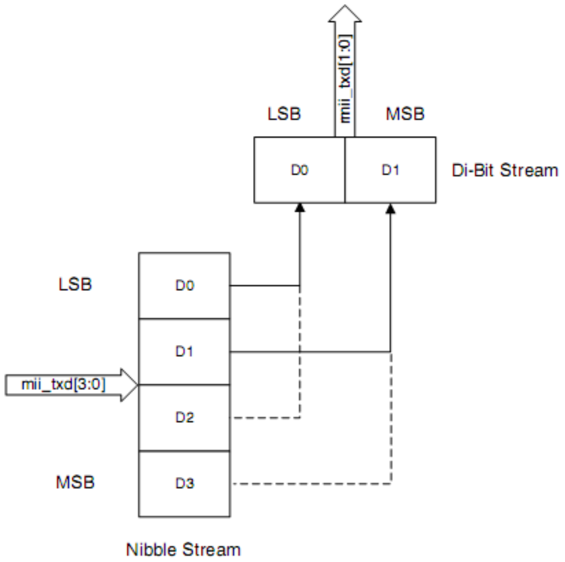


Fig. 24-3 RMII transmission bit ordering

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 gRup and yield the correct frame data.

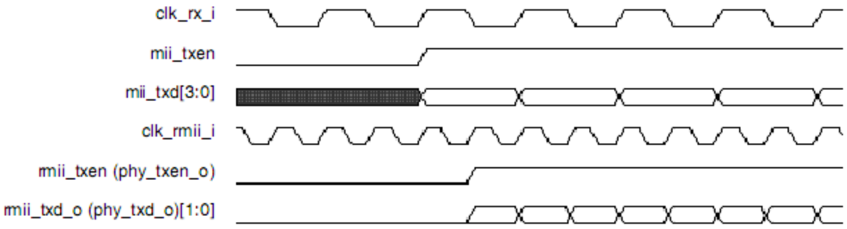


Fig. 24-4 Start of MII and RMII transmission in 100-Mbps mode

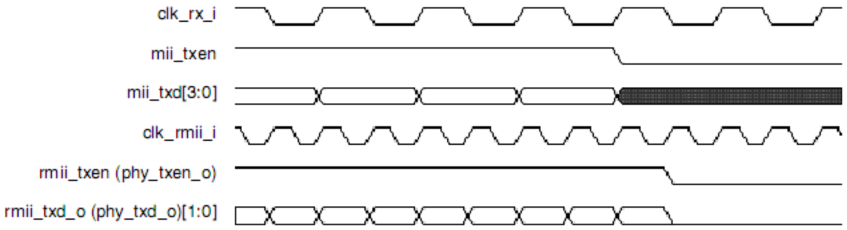


Fig. 24-5 End of MII and RMII Transmission in 100-Mbps Mode

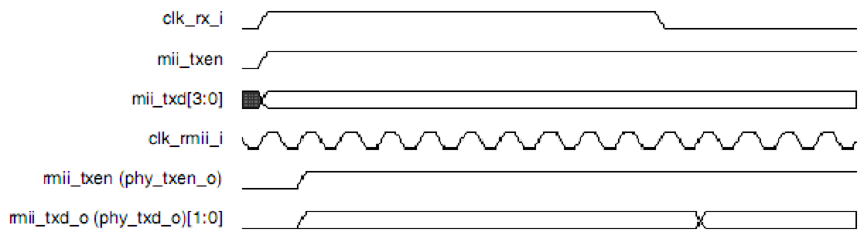


Fig. 24-6 Start of MII and RMII Transmission in 10-Mbps Mode

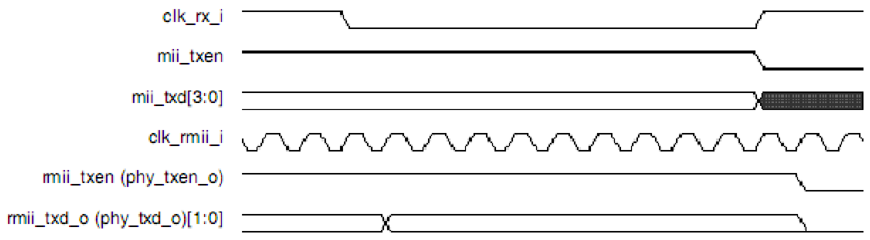


Fig. 24-7 End of MII and RMII Transmission in 10-Mbps Mode

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).

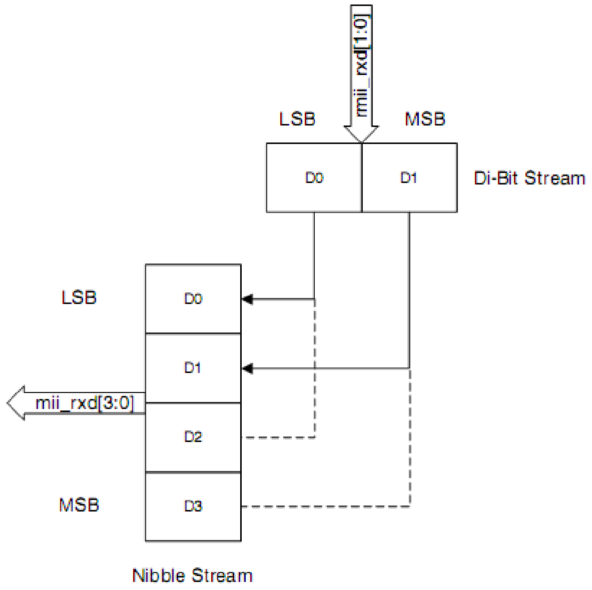


Fig. 24-8 RMII receive bit ordering

24.3.3 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 transmit 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.

24.3.4 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 fromthe 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.

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE

Fig. 24-9 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 2í01
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.

24.3.5 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 FF FF FF FF FF
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.

24.3.6 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.

24.4 Register description

24.4.1 Register 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
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

Name	Offset	Size	Reset Value	Description
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 Mulitcast 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
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

Name	Offset	Size	Reset Value	Description
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 Adderss Register
GMAC_HW_FEA_REG	0x1058	W	0x000d0f17	The presence of the optional features/functions of the core

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

24.4.2 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/SGMII/SMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY.

Bit	Attr	Reset Value	Description
23	RW	0x0	WD Watchdog Disable When this bit is set, the 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
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.

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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.
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

Bit	Attr	Reset Value	Description
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 transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive.
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmit 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.

Bit	Attr	Reset Value	Description
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set.

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_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 When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE 802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_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

Bit	Attr	Reset Value	Description
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.
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

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.
15:8	RO	0x0	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the deassertion 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.

Bit	Attr	Reset Value	Description
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.
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.

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register 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

Bit	Attr	Reset Value	Description
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state.
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved
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

Bit	Attr	Reset Value	Description
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.

Bit	Attr	Reset Value	Description
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception.
0	R/WSC	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
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.

Bit	Attr	Reset Value	Description
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 RGMIIinterface. 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
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

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
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/WSC	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

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/WSC	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.
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.

Bit	Attr	Reset Value	Description
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
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.

Bit	Attr	Reset Value	Description
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 - 2KBytes) 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 - 2KBytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/WSC	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.

Bit	Attr	Reset Value	Description
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read.
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/WSC	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.

Bit	Attr	Reset Value	Description
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.

Bit	Attr	Reset Value	Description
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
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.

Bit	Attr	Reset Value	Description
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.
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

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

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
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

Bit	Attr	Reset Value	Description
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.
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.
8:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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

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
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.

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
0	R/WSC	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.

Bit	Attr	Reset Value	Description
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.
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

Bit	Attr	Reset Value	Description
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor.
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory.

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.

Bit	Attr	Reset Value	Description
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register 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

Bit	Attr	Reset Value	Description
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO.
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received.
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state.
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].

Bit	Attr	Reset Value	Description
3	W1C	0x0	TJT Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, meaning that the transmitter had been excessively active. The transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.
2	W1C	0x0	TU Transmit Buffer Unavailable This bit indicates that the Next Descriptor in the Transmit List is owned by the host and cannot be acquired by the DMA. Transmission is suspended. Bits[22:20] explain the Transmit Process state transitions. To resume processing transmit descriptors, the host should change the ownership of the bit of the descriptor and then issue a Transmit Poll Demand command.
1	W1C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped.
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.

Bit	Attr	Reset Value	Description
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits.
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset.
23:22	RO	0x0	reserved
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register 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

Bit	Attr	Reset Value	Description
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 deasserted 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 deassertion is effective only after flow control is asserted.

Bit	Attr	Reset Value	Description
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.
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 wriiten, 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

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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
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.

Bit	Attr	Reset Value	Description
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.
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.

Bit	Attr	Reset Value	Description
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

Bit	Attr	Reset Value	Description
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

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 Register0 (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

Bit	Attr	Reset Value	Description
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16.
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8.
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4.
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, the 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

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.

GMAC_HW_FEA_REG
Address: Operational Base + offset (0x1058)
The presence of the optional features/functions of the core Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	Reserved
24	RO	0x0	Alternate (Enhanced Descriptor)
23:20	RO	0x0	Reserved
19	RO	0x1	RxFIFO > 2048 Bytes
18	RO	0x1	IP Checksum Offload (Type 2) in Rx
17	RO	0x0	IP Checksum Offload (Type 1) in Rx
16	RO	0x1	Checksum Offload in Tx
15:14	RO	0x0	Reserved
13	RO	0x0	IEEE 1588-2008 Advanced Time-Stamp

Bit	Attr	Reset Value	Description
12	RO	0x0	IEEE 1588-2002 Time-Stamp
11	RO	0x1	RMON module
10	RO	0x1	PMT Magic Packet
9	RO	0x1	PMT Remote Wakeup
8	RO	0x1	SMA (MDIO) Interface
7	RO	0x0	Reserved
6	RO	0x0	PCS registers (TBI/SGMII/RTBI PHY interface)
5	RO	0x0	Multiple MAC Address Registers
4	RO	0x1	HASH Filter
3	RO	0x0	Reserved
2	RO	0x1	Half-Duplex support
1	RO	0x1	1000 Mbps support
0	RO	0x1	10/100 Mbps support

24.5 Interface Description

Table 24-1 RMII Interface Description

Module pin name	Direction	Pad name	IOMUX
RMII interface			
mac_clk	I/O	GPIO4_A[3]	GRF_GPIO4AL_IOMUX[14:12]=3'b011
mac_txen	O	GPIO4_A[4]	GRF_GPIO4AH_IOMUX[2:0]=3'b011
mac_txd1	O	GPIO3_D[5]	GRF_GPIO3DH_IOMUX[6:4]=3'b011
mac_txd0	O	GPIO3_D[4]	GRF_GPIO3DH_IOMUX[2:0]=3'b011
mac_rxdv	I	GPIO4_A[1]	GRF_GPIO4AL_IOMUX[6:4]=3'b011
mac_rxer	I	GPIO4_A[2]	GRF_GPIO4AL_IOMUX[10:8]=3'b011
mac_rxd1	I	GPIO3_D[7]	GRF_GPIO3DH_IOMUX[14:12]=3'b011
mac_rxd0	I	GPIO3_D[6]	GRF_GPIO3DH_IOMUX[10:8]=3'b011
Management interface			
mac_mdio	I/O	GPIO4_A[5]	GRF_GPIO4AH_IOMUX[5:4]=2'b11
mac_mdc	O	GPIO4_A[0]	GRF_GPIO4AL_IOMUX[1:0]=2'b11

Table 24-2 RGMII Interface Description

Module pin name	Direction	Pad name	IOMUX
RGMII/RMII interface			
mac_clk	I/O	GPIO4_A[3]	GRF_GPIO4AL_IOMUX[14:12]=3'b011
mac_txclk	O	GPIO4_B[1]	GRF_GPIO4BL_IOMUX[6:4]=3'b011
mac_txen	O	GPIO4_A[4]	GRF_GPIO4AH_IOMUX[2:0]=3'b011
mac_txd3	O	GPIO3_D[1]	GRF_GPIO3DL_IOMUX[6:4]=3'b011
mac_txd2	O	GPIO3_D[0]	GRF_GPIO3DL_IOMUX[2:0]=3'b011
mac_txd1	O	GPIO3_D[5]	GRF_GPIO3DH_IOMUX[6:4]=3'b011
mac_txd0	O	GPIO3_D[4]	GRF_GPIO3DH_IOMUX[2:0]=3'b011
mac_rxclk	I	GPIO4_A[6]	GRF_GPIO4AH_IOMUX[10:8]=3'b011
mac_rxdv	I	GPIO4_A[1]	GRF_GPIO4AL_IOMUX[6:4]=3'b011
mac_rxd3	I	GPIO3_D[3]	GRF_GPIO3DL_IOMUX[14:12]=3'b011
mac_rxd2	I	GPIO3_D[2]	GRF_GPIO3DL_IOMUX[10:8]=3'b011
mac_rxd1	I	GPIO3_D[7]	GRF_GPIO3DH_IOMUX[14:12]=3'b011
mac_rxd0	I	GPIO3_D[6]	GRF_GPIO3DH_IOMUX[10:8]=3'b011

mac_crs	I	GPIO4_A[7]	GRF_GPIO4AH_IOMUX[14:12]=3'b011
mac_col	I	GPIO4_B[0]	GRF_GPIO4BL_IOMUX[2:0]=3'b011
Management interface			
mac_mdio	I/O	GPIO4_A[5]	GRF_GPIO4AH_IOMUX[5:4]=2'b11
mac_mdc	O	GPIO4_A[0]	GRF_GPIO4AL_IOMUX[1:0]=2'b11

24.6 Application Notes

24.6.1 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.

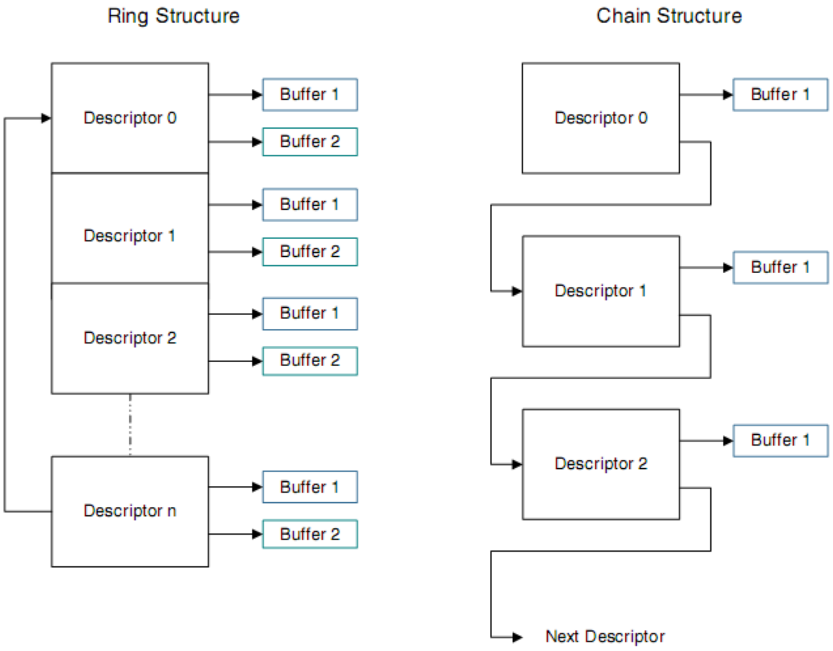


Fig. 24-10 Descriptor Ring and Chain Structure

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).

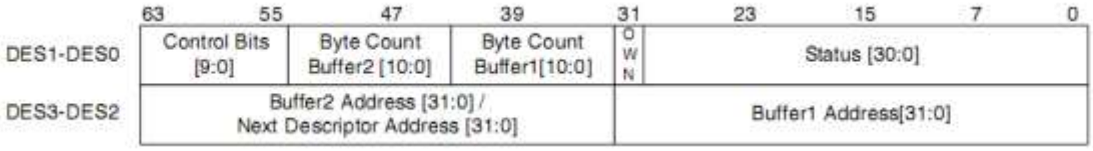


Fig. 24-11 Rx/Tx Descriptors definition

24.6.2 Receive Descriptor

The GMAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMA always 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 24-3 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 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 GMACCore.
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 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 24-4 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 24-5 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 24-6 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.

24.6.3 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 24-7 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 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 24-8 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram conforms to IPv4. The Checksum engine detects whether the TCP, UDP, or ICMP segment is encapsulated in IPv4 or IPv6 and processes its data accordingly.
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 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 24-9 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 24-10 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.)

24.6.4 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. Steps (1) and (2) are to be followed if the TBI/SGMII/RTBI PHY interface is enabled, otherwise follow steps (3) and (4).
1. Program the AN Control register GMAC_AN_CTRL to enable Auto-negotiation ANE (bit-12). Setting ELE (bit-14) of this register will enable the PHY to loop back the transmit data.
2. Check the AN Status Register GMAC_AN_STATUS for completion of the Auto-negotiation process. ANC (bit-5) should be set, and link status (bit-2), when set, indicates that the link is up.
3. 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.
4. 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 register GMAC_GMII_ADDR (bits 15-11).
5. Provide the MAC address registers (GMAC_MAC_ADDR0_HI and GMAC_MAC_ADDR0_LO). If more than one MAC address is enabled in your configuration (during configuration in coreConsultant), program them appropriately).
6. If Hash filtering is enabled in your configuration, program the Hash filter register (GMAC_HASH_TAB_HI and GMAC_HASH_TAB_LO).
7. 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.
8. 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
9. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
10. 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.
11. 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 egister 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.

24.6.5 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].

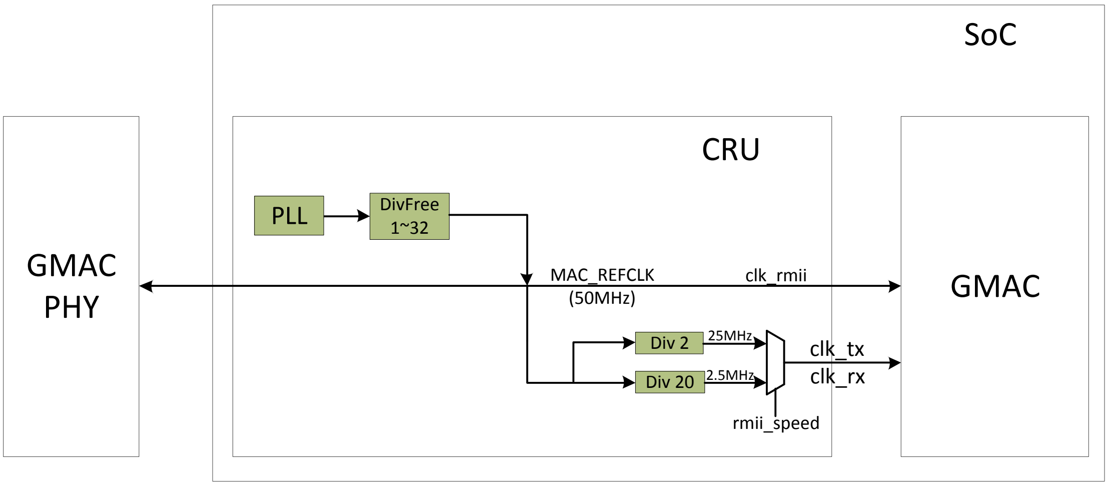


Fig. 24-12 RMII clock architecture when clock source from CRU

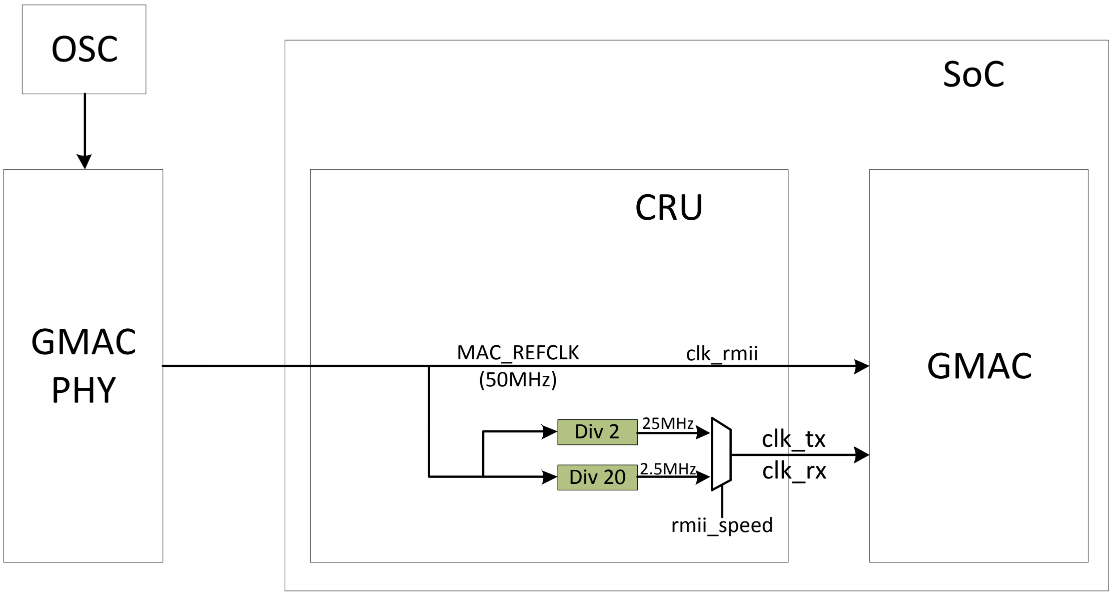


Fig. 24-13 RMII clock architecture when clock source from external OSC

In RGMII mode, clock architecture only supports that TX clock source is from CRU as following figure.

In order to dynamicly adjust the timing between TX/RX clock 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.

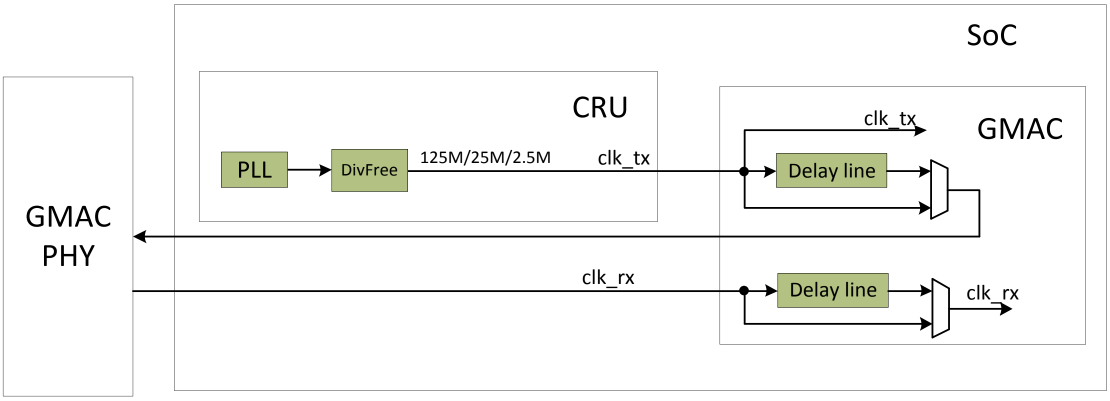
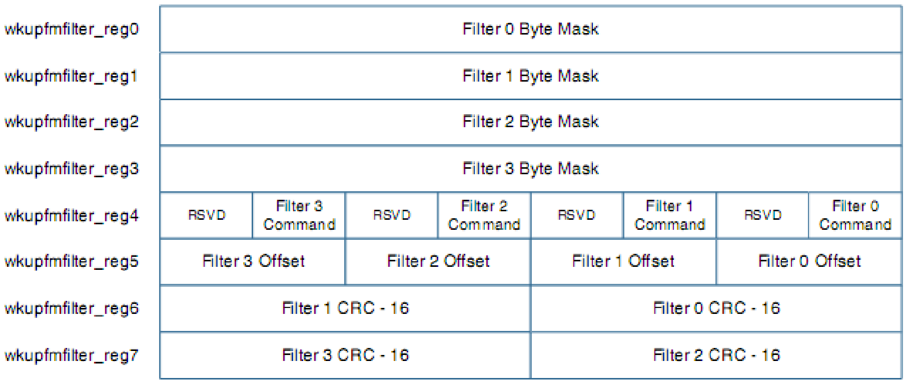


Fig. 24-14 RGMII clock architecture when clock source from CRU

24.6.6 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. 24-15 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.

24.6.7 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.

24.6.8 GRF Register Summary

GRF Register	Register Description
GRF_SOC_CON1[8:6]	PHY interface select 3'b001: RGMII 3'b100: RMII All others: Reserved
GRF_SOC_CON1[9]	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_SOC_CON1[10]	gmac_speed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_SOC_CON1[11]	RMII clock selection 1'b1: 25MHz 1'b0: 2.5MHz
GRF_SOC_CON1[13:12]	RGMII clock selection 2'b00: 125MHz 2'b11: 25MHz 2'b10: 2.5MHz
GRF_SOC_CON1[14]	RMII mode selection 1'b1: RMII mode 1’b0: Reserved
GRF_SOC_CON3[6:0]	RGMII TX clock delayline value
GRF_SOC_CON3[13:7]	RGMII RX clock delayline value
GRF_SOC_CON3[14]	RGMII TX clock delayline enable 1'b1: enable 1'b0: disable
GRF_SOC_CON3[15]	RGMII RX clock delayline enable 1'b1: enable 1'b0: disable

Chapter 25 PS/2 Controller

25.1 Overview

The PS/2 is a bi-directional serial bus protocol that provides an efficient and simple method of information exchange between host and devices. This PS/2 controller supports master mode acting as a bridge between AMBA APB protocol and generic PS/2 bus system.
PS/2 controller supports the following features:

- Support 32bits AMBA2.0 APB bus interface protocol
- Support PS/2 data communication protocol
- Support PS/2 master mode
- Software programmable timing requirement to support max PS/2 clock frequency to 33KHZ
- Support status to be queried for data communication error
- Support interrupt mode for data communication finish
- Support timeout mechnism for data communication
- Support interrupt mode for data communication timeout

25.2 Block Diagram

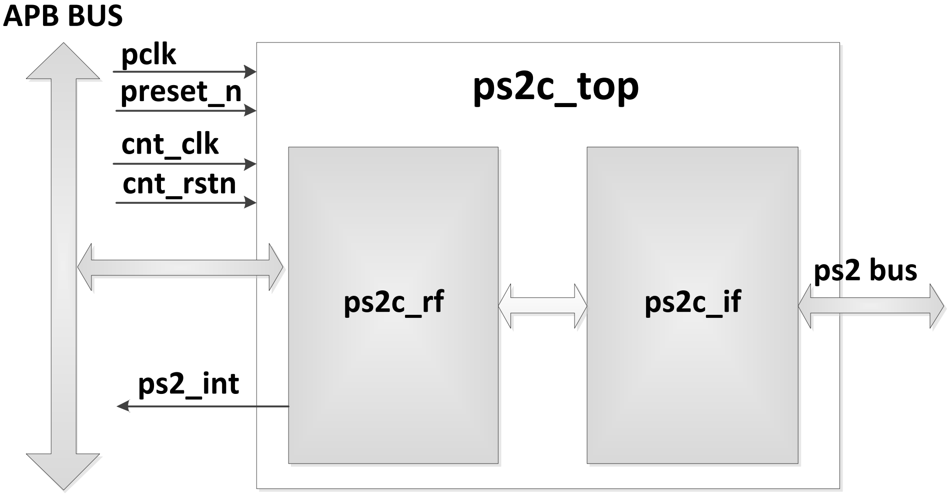


Fig. 25-1 PS/2 controller architecture

PS2C_RF
PS2C_RF module is used to control the PS/2 controller operation by the host with APB interface. It implements the register configuration and the interrupt functionality. It also collects the status signals to be queried by the host.

PS2C_IF
PS2C_IF module implements the PS/2 master operation for transmitting data to and receiving data from other PS/2 devices. The PS/2 master operation works at the pclk domain.
PS2C_IF module also implements timing requirement counters and all the counters work at the cnt_clk domain.

PS2C_TOP
PS2C_TOP module is the top module of the PS/2 controller.

25.3 Function description

This chapter provides a description about the functions and behavior of PS/2 controller.
The PS/2 controller supports only Master function. The operations of PS/2 controller is divided to 3 parts and described separately: receiving mode, sending mode and inhibition mode.

25.3.1 PS/2 receiving mode

Configure the PS2C_CTRL[2:1] as 2’b00 and enable controller, the PS/2 controller will work at receiving mode. In receiving mode, the PS/2 controller can receive the data transmitted from the PS/2 device. The receiving timing is showed as fig,

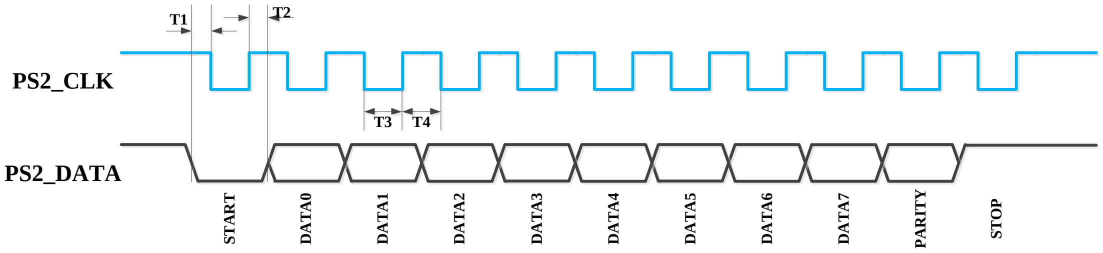


Fig. 25-2 PS/2 host receiving timing

In receiving mode, the PS/2 controller waits to receive data transmitted from PS/2 device. Every transmission includes 11 bits data (one "start" bit, eight "information" bits, one "parity" bit and one "stop" bit).

When negedge of ps2_clk is captured and ps2_data is low, the "start" bit is received. Then controller captures the following 10 bits data at the negedge of ps2_clk.

After finish receiving a data package, the controller will generate the receiving finish status and receiving finish interrupt (if receiving finish interrupt is enabled). The host can get the receiving data by reading the PS2C_RBR register. And if there is error in the data package, the controller will generate the error status.

In the controller, there is a receiving timeout dection circuit. This circuit is enabled by configuring the PS2C_CTRL[3] as 1'b1. The receiving timeout period is set by configuring the PS2C_RTR register. The unit of PS2C_RTR is cnt_clk cycle. If a data package receiving time exceeds the receiving timeout period after controller receives "start" bit, the controller will generate the receiving timeout status and receiving timeout interrupt (if receiving timeout interrupt is enabled).

25.3.2 PS/2 sending mode

Configure the PS2C_CTRL[2:1] as 2'b01 and enable controller, the PS/2 controller will work at sending mode. In sending mode, the PS/2 controller can send the data to the PS/2 device. The transmission timing is showed as fig,

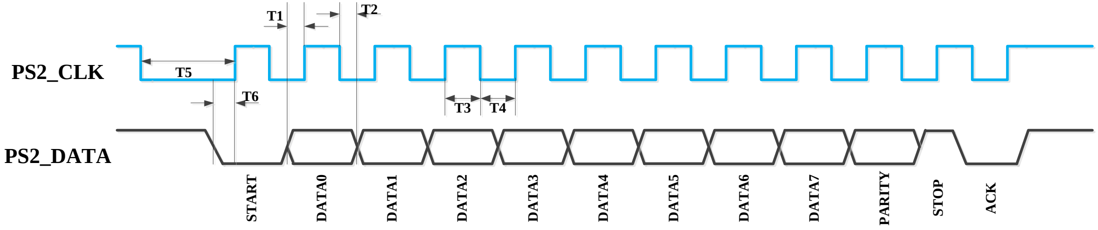


Fig. 25-3 PS/2 host sending timing

In sending mode, the host sends data by writing PS2C_TBR register which includes 8 information bits. Every transmission includes 12 bits data (one "start" bit, eight "information" bits, one "parity" bit, one "stop" bit and one "ack" bit from PS/2 device).

When there is data to be sent to PS/2 device, controller pulls low the ps2_clk for T5 time duration (at least 100 microseconds) to request the PS/2 device receiving data. During T5 time duration, controller needs to pull low the ps2_data to low for T6 time duration (at least 5 microseconds). After T5 time duration, the "start" bit will be sent.

After the "start" bit is sent, controller sends the following 10 bits data captured by PS/2 device at the posedge of ps2_clk. The controller changes the sending bit data at the low duration of ps2_clk, and T1 (at least 5 microseconds) and T2 (at least 5 microseconds) time duration must be met.

When finish sending the 11 bits data, controller will capture the "ack" bit at the next negedge of ps2_clk. If the "ack" bit is 1'b0, it represents data package sending success, otherwise represents sending error.

Host can configure the PS2C_TRR1, PS2C_TRR2 and PS2C_TRR3 registers respectively to meet the time of T5, T6 and T2. The unit of PS2C_TRR1, PS2C_TRR2 and PS2C_TRR3 is cnt_clk cycle.

After finish sending a data package, the controller will generate the sending finish status and sending finish interrupt (if sending finish interrupt is enabled). And if the "ack" bit is 1'b1, the controller will generate the error status.

In the controller, there is a sending timeout dection circuit. This circuit is enabled by configuring the PS2C_CTRL[4] as 1'b1. The sending timeout period is set by configuring the PS2C_WTR register. The unit of PS2C_WTR is cnt_clk cycle. If a data package sending time exceeds the sending timeout period after controller sends request to device, the controller will generate the sending timeout status and sending timeout interrupt (if sending timeout interrupt is enabled).

25.3.3 PS/2 inhibition mode

Configure the PS2C_CTRL[2:1] as 2'b10 and enable controller, the PS/2 controller will work at inhibition mode. In inhibition mode, the PS/2 controller will pull ps2_clk low to inhibit the PS/2 device sending data. The controller will not release the inhibition until disable controller.

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
PS2C_CTRL	0x0000	W	0x00000000	PS2C control register
PS2C_RBR	0x0004	W	0x00000000	PS2C receiving buffer register
PS2C_TBR	0x0008	W	0x00000000	PS2C sending buffer register
PS2C_STAT	0x000c	W	0x00000010	PS2C status register
PS2C_IER	0x0010	W	0x00000000	PS2C interrupt enable register
PS2C_ICR	0x0014	W	0x00000000	PS2C interrupt clear register
PS2C_ISR	0x0018	W	0x00000000	PS2C interrupt status register
PS2C_TRR1	0x001c	W	0x00000b39	PS2C time require register1
PS2C_TRR2	0x0020	W	0x000000ef	PS2C time require register2
PS2C_TRR3	0x0024	W	0x000000ef	PS2C time require register3
PS2C_RTR	0x0028	W	0x00000000	PS2C receiving timeout require register
PS2C_WTR	0x002c	W	0x00000000	PS2C sending timeout require register
PS2C_FLT	0x0030	W	0x00000000	PS2C filter width selection register

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

25.4.2 Detail Register Description

PS2C_CTRL

Address: Operational Base + offset (0x0000)

PS2C control register

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved

Bit	Attr	Reset Value	Description
20:16	WO	0x00	write_enable bit0~4 write enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 cannot be written by software. ...... When bit 20=1, bit 4 can be written by software. When bit 20=0, bit 4 cannot be written by software.
15:5	RO	0x0	reserved
4	RW	0x0	ps2c_tm_timeout_en ps2c sending timeout enable This bit is used to enable the sending timeout circuit. 1'b0: ps2c sending timeout disable. 1'b1: ps2c sending timeout enable. Only when this bit is set, the PS2C_STAT[6](ps2c_tm_timeouot) is access.
3	RW	0x0	ps2c_rv_timeout_en ps2c receiving timeout enable This bit is used to enable the receiving timeout circuit. 1'b0: ps2c receiving timeout disable. 1'b1: ps2c receiving timeout enable. Only when this bit is set, the PS2C_STAT[2](ps2c_rv_timeouot) is access.
2:1	RW	0x0	ps2c_mode ps2c work mode selection 2'b00: ps2c works as receiving mode. 2'b01: ps2c works as sending mode. 2'b10: ps2c works as inhibition mode, inhibits the ps2 device send data. 2'b11: reserved.
0	RW	0x0	ps2c_enable ps2c enable signal 1'b0: ps2c disable. 1'b1: ps2c enable.

PS2C_RBR
Address: Operational Base + offset (0x0004)
PS2C receiving buffer register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	ps2c_rbuf ps2c receiving buffer

PS2C_TBR
Address: Operational Base + offset (0x0008)
PS2C sending buffer register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	WO	0x00	ps2c_tbuf ps2c sending buffer

PS2C_STAT
Address: Operational Base + offset (0x000c)
PS2C status register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:21	WO	0x0	write_enable2 bit5~6 write enable When bit 21=1, bit 5 can be written by software. When bit 21=0, bit 5 cannot be written by software. When bit 22=1, bit 6 can be written by software. When bit 22=0, bit 6 cannot be written by software.
20:19	RO	0x0	reserved
18:17	WO	0x0	write_enable1 bit1~2 write enable When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 cannot be written by software. When bit 18=1, bit 2 can be written by software. When bit 18=0, bit 2 cannot be written by software.
16:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	ps2c_tm_busy ps2c sending busy This bit is set when write PS2C_TBR to start data sending to ps2 device, and is cleared when ps2c receives ack signal or happen sending timeout. 1'b0: ps2c is in sending idle status. 1'b1: ps2c is in sending busy status.
6	RW	0x0	ps2c_tm_timeout ps2c sending timeout When ps2c sending busy status exceeds the ps2c_tm_timeout_req cnt_clk cycles time, this bit is set. 1'b0: ps2c is not sending timeout. 1'b1: ps2c is sending timeout. This bit is cleared by writing 1'b0.
5	RW	0x0	ps2c_tm_err ps2c sending error When ps2c_tbe is 1'b1, this bit indicates whether the data sent to ps2 device is successfully received. 1'b0: ps2 device ack 1'b0 to ps2c. 1'b1: ps2 device ack 1'b1 to ps2c. This bit is cleared by writing 1'b0.
4	RO	0x1	ps2c_tbe ps2c sending buffer empty 1'b0: ps2c sending buffer not empty. 1'b1: ps2c sending buffer empty. This bit is cleared by writing the PS2C_TBR.
3	RO	0x0	ps2c_rv_busy ps2c receiving busy This bit is set when ps2c receives start signal, and is cleared when ps2c receives stop signal or happen receiving timeout. 1'b0: ps2c is in receiving idle status. 1'b1: ps2c is in receiving busy status.
2	RW	0x0	ps2c_rv_timeout ps2c receiving timeout When ps2c receiving busy status exceeds the ps2c_rv_timeout_req cnt_clk cycles time, this bit is set. 1'b0: ps2c is not receiving timeout. 1'b1: ps2c is receiving timeout. This bit is cleared by writing 1'b0.

Bit	Attr	Reset Value	Description
1	RW	0x0	ps2c_rv_err ps2c receiving error When ps2c_ibf is 1'b1, this bit indicates whether the receiving data is odd parity error. 1'b0: odd parity sucess. 1'b1: odd parity error. This bit is cleared by writing 1'b0.
0	RO	0x0	ps2c_rbf ps2c receiving buffer full 1'b0: ps2c receiving buffer not full. 1'b1: ps2c receiving buffer full. This bit is cleared by reading the PS2C_RBR.

PS2C_IER
Address: Operational Base + offset (0x0010)
PS2C interrupt enable register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	WO	0x0	write_enable bit0~3 write enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 cannot be written by software. ...... When bit 19=1, bit 3 can be written by software. When bit 19=0, bit 3 cannot be written by software.
15:4	RO	0x0	reserved
3	RW	0x0	ps2c_tm_timeout_inten enable ps2c sending timeout interrupt 1'b0: disable interrupt. 1'b1: enable interrupt.
2	RW	0x0	ps2c_fsh_tm_inten enable ps2c finish sending one byte interrupt 1'b0: disable interrupt. 1'b1: enable interrupt.
1	RW	0x0	ps2c_rv_timeout_inten enable ps2c receiving timeout interrupt 1'b0: disable interrupt. 1'b1: enable interrupt.

Bit	Attr	Reset Value	Description
0	RW	0x0	ps2c_fsh_rv_inten enable ps2c finish receiving one bype interrupt 1'b0: disable interrupt. 1'b1: enable interrupt.

PS2C_ICR
Address: Operational Base + offset (0x0014)
PS2C interrupt clear register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	WO	0x0	write_enable bit0~3 write enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 cannot be written by software. ...... When bit 19=1, bit 3 can be written by software. When bit 19=0, bit 3 cannot be written by software.
15:4	RO	0x0	reserved
3	W1C	0x0	ps2c_tm_timeout_intclr clear ps2c sending timeout interrupt 1'b0: not clear interrupt. 1'b1: clear interrupt.
2	W1C	0x0	ps2c_fsh_tm_intclr clear ps2c finish sending one byte interrupt 1'b0: not clear interrupt. 1'b1: clear interrupt.
1	W1C	0x0	ps2c_rv_timeout_intclr clear ps2c receiving timeout interrupt 1'b0: not clear interrupt. 1'b1: clear interrupt.
0	W1C	0x0	ps2c_fsh_rv_intclr clear ps2c finish receiving one bype interrupt 1'b0: not clear interrupt. 1'b1: clear interrupt.

PS2C_ISR
Address: Operational Base + offset (0x0018)
PS2C interrupt status register

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

Bit	Attr	Reset Value	Description
3	RO	0x0	ps2c_tm_timeout_intst ps2c sending timeout interrupt status 1'b0: interrupt is not valid. 1'b1: interrupt is valid.
2	RO	0x0	ps2c_fsh_tm_intst ps2c finish sending one byte interrupt status 1'b0: interrupt is not valid. 1'b1: interrupt is valid.
1	RO	0x0	ps2c_rv_timeout_intst ps2c receiving timeout interrupt status 1'b0: interrupt is not valid. 1'b1: interrupt is valid.
0	RO	0x0	ps2c_fsh_rv_intst ps2c finish receiving one bype interrupt status 1'b0: interrupt is not valid. 1'b1: interrupt is valid.

PS2C_TRR1
Address: Operational Base + offset (0x001c)
PS2C time require register1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000b39	ps2c_time_req1 ps2c time require 1 When ps2c sends data to ps2 device, this value defines the width of the low ps2_clk_o to send transmission request to the ps2 device. The unit is one cnt_clk cycle.

PS2C_TRR2
Address: Operational Base + offset (0x0020)
PS2C time require register2

Bit	Attr	Reset Value	Description
31:0	RW	0x000000ef	ps2c_time_req2 ps2c time require 2 When ps2c sends data to ps2 device, this value defines the width between first ps2_data_o falling edge and first ps2_clk_o rising edge. The unit is one cnt_clk cycle.

PS2C_TRR3
Address: Operational Base + offset (0x0024)
PS2C time require register3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x000000ef	ps2c_time_req3 ps2c time require 3 When ps2c sending data to ps2 device, this value defines data change time after ps2_clk_i falling edge. The unit is one cnt_clk cycle.

PS2C_RTR
Address: Operational Base + offset (0x0028)
PS2C receiving timeout require register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ps2c_rv_timeout_req ps2c receiving timeout require time The unit is one cnt_clk cycle.

PS2C_WTR
Address: Operational Base + offset (0x002c)
PS2C sending timeout require register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ps2c_tm_timeout_req ps2c sending timeout require time The unit is one cnt_clk cycle.

PS2C_FLT
Address: Operational Base + offset (0x0030)
PS2C filter width selection register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	ps2c_clk_flt_sel ps2c filter width selection for ps2_clk The unit is one cnt_clk cycle.

25.5 Interface description

Table 25-1 PS/2 controller Interface Description

Module pin	Direction	Pad name	IOMUX
PS/2 controller 0 Interface			
ps2_clk	I/O	GPIO8_A[0]	GPIO8A_IOMUX[1:0] = 01
ps2_data	I/O	GPIO8_A[1]	GPIO8A_IOMUX[3:2] = 01

25.6 Application Notes

The PS/2 controller operation flow charts below are to describe how the software configures and performs a PS/2 transmission through this PS/2 controller. Descriptions are divided into 3 sections: receiving data mode, sending data mode and inhibition mode.
Note: Before change the controller’s working mode, software needs to disable the controller firstly.

- Receiving data mode

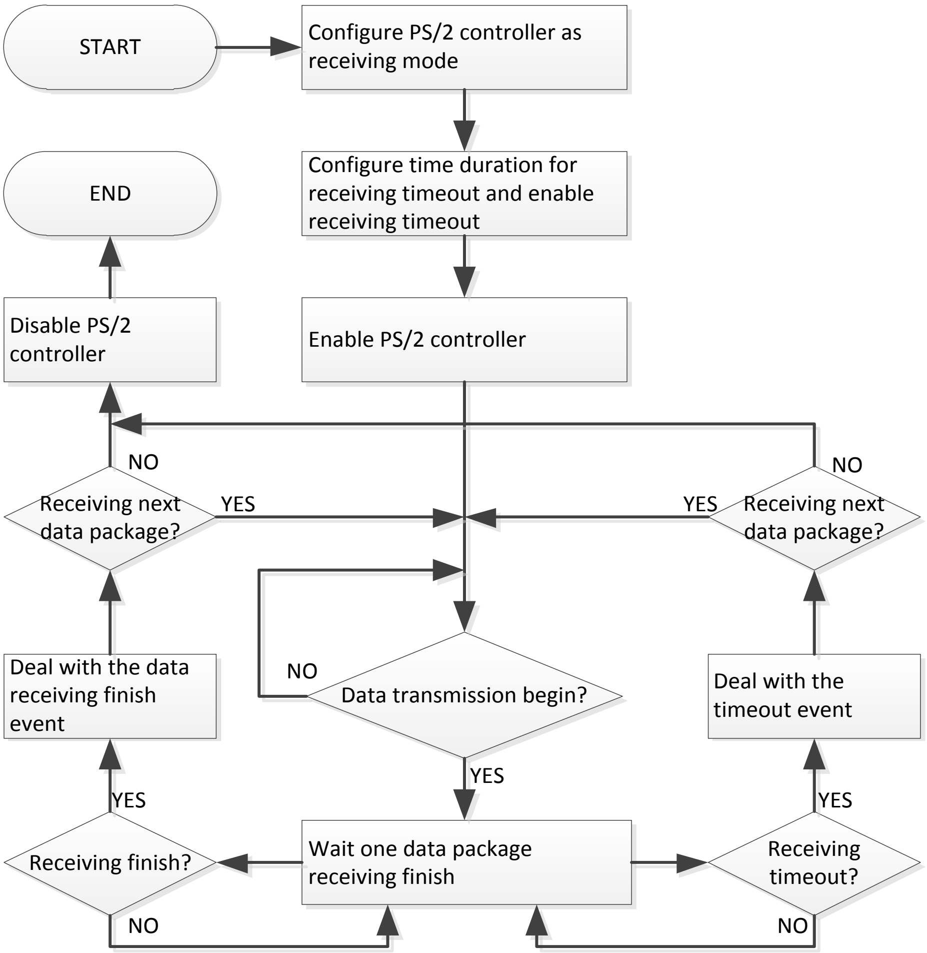

START
Configure PS/2 controller as receiving mode
END
Configure time duration for receiving timeout and enable receiving timeout
Disable PS/2 controller
Enable PS/2 controller
NO
NO
Receiving next data package?
YES
YES
Receiving next data package?
NO
Data transmission begin?
Deal with the data receiving finish event
Deal with the timeout event
YES
YES
YES
Receiving finish?
Wait one data package receiving finish
Receiving timeout?
NO
NO

Fig. 25-4 Flow chat for PS/2 controller receiving data mode

- Sending data mode

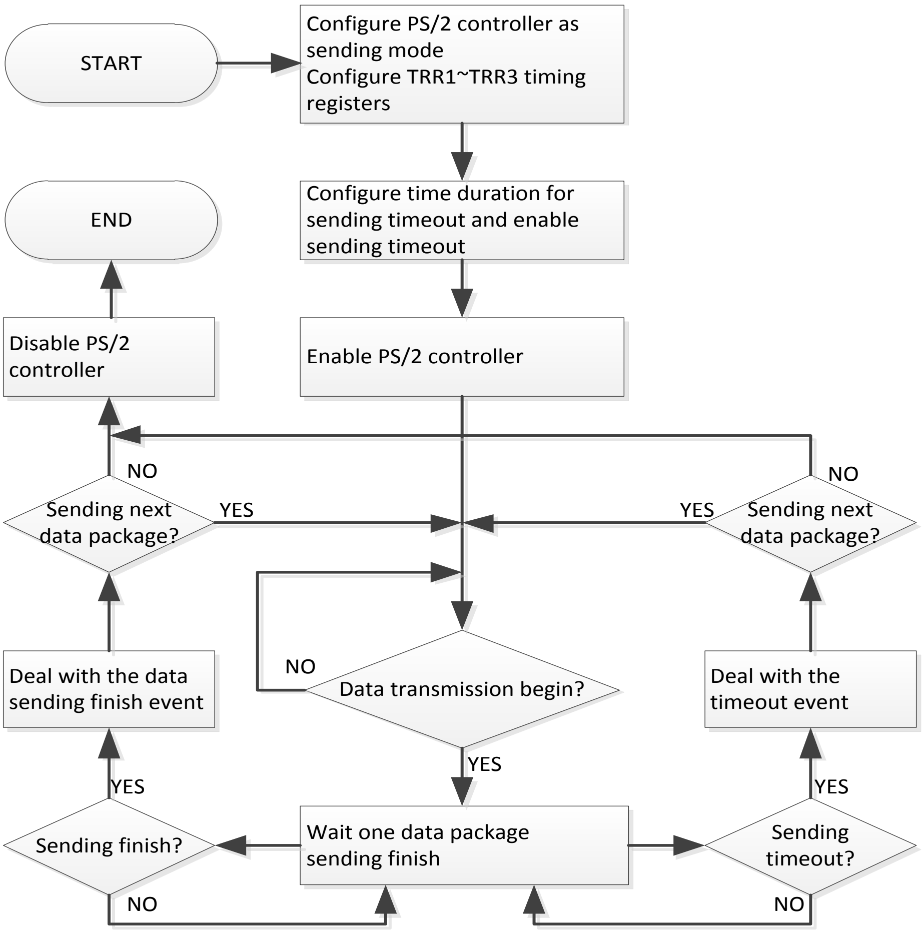

START
Configure PS/2 controller as sending mode
Configure TRR1~TRR3 timing registers
Configure time duration for sending timeout and enable sending timeout
END
Disable PS/2 controller
Enable PS/2 controller
NO
NO
Sending next data package?
YES
YES
Sending next data package?
NO
Data transmission begin?
Deal with the data sending finish event
Deal with the timeout event
YES
YES
YES
Sending finish?
Wait one data package sending finish
Sending timeout?
NO
NO

Fig. 25-5 Flow chat for PS/2 controller sending data mode

- Inhibition PS/2 device mode

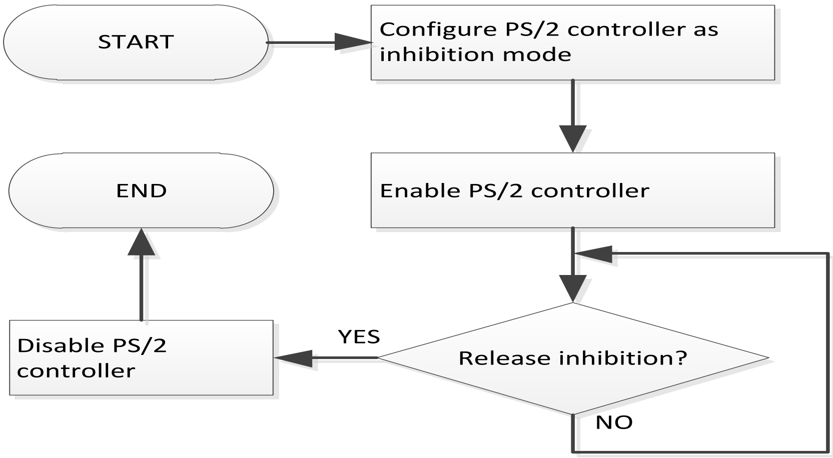


Fig. 25-6 Flow chat for PS/2 controller inhibition mode

Chapter 26 Smart Card Controller

26.1 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

26.2 Block Diagram

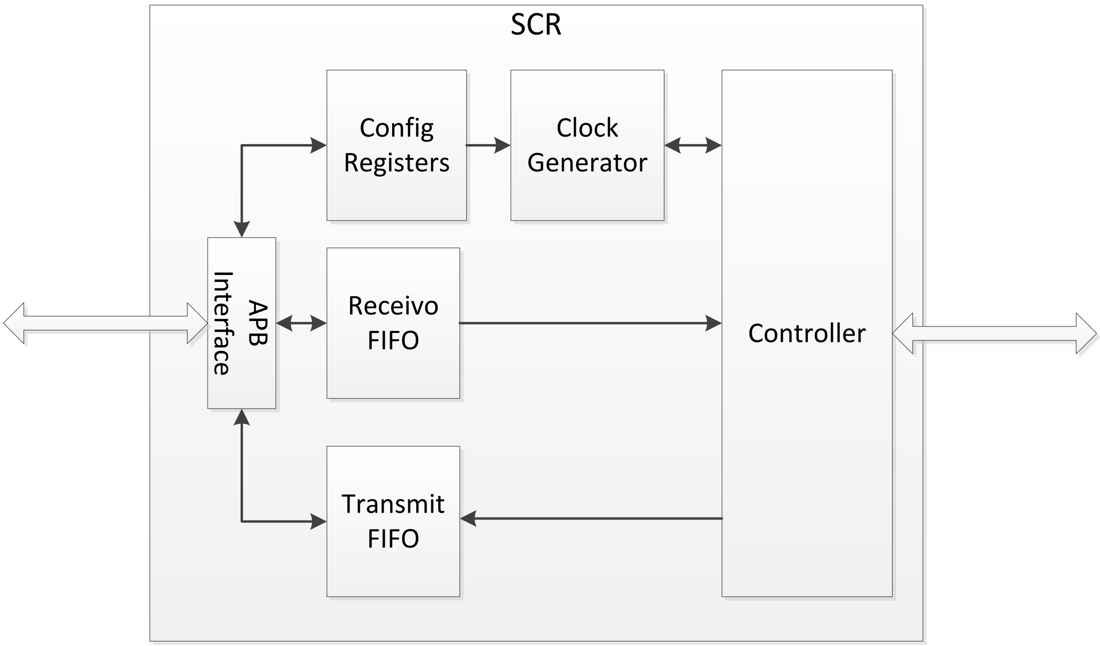


Fig. 26-1 SCR Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface.

26.2.1 APB Interface

The host processor gets access to PWM Register Block through the APB slave interface.

26.2.2 Configuration Registers

The Configuration Registers block provides control over all functions of the Smart Card Reader

26.2.3 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.

26.2.4 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.

26.2.5 Transmit FIFO

The Transmit FIFO is used to store the data to be transmitted to the Smart Card.

26.2.6 Clock Generator

The Clock Generator generates the Smart Card Clock signal and the Baud Clock Impulse signal, used in timing the Smart Card Reader.)

26.3 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

26.3.1 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.

26.3.2 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.

26.3.3 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 in .

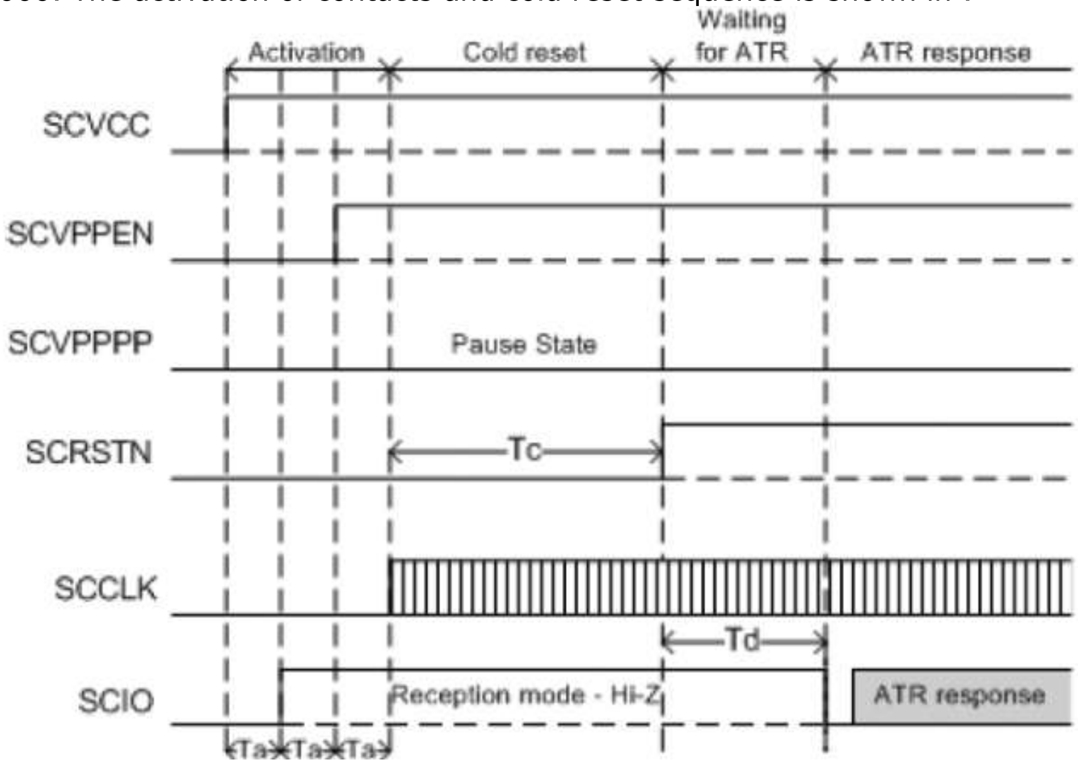


Fig. 26-2 Activation, Cold Reset and ATR

26.3.4 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.

26.3.5 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 . After the SCRSTN assertion, the Warm Reset sequence then continues the same way as the Cold Reset sequence.

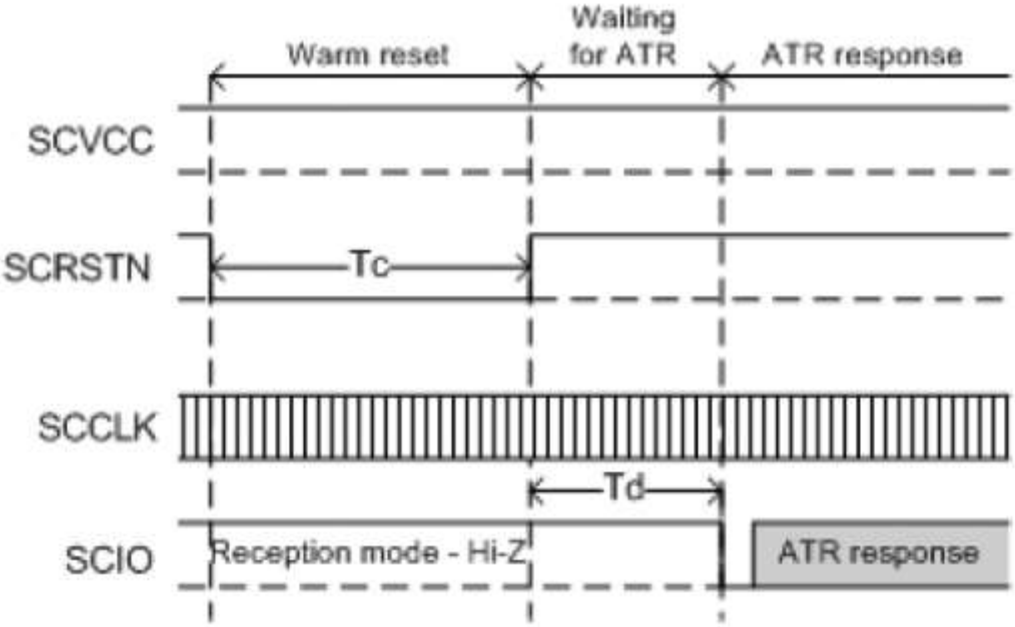


Fig. 26-3 Warm Reset and ATR

26.3.6 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.

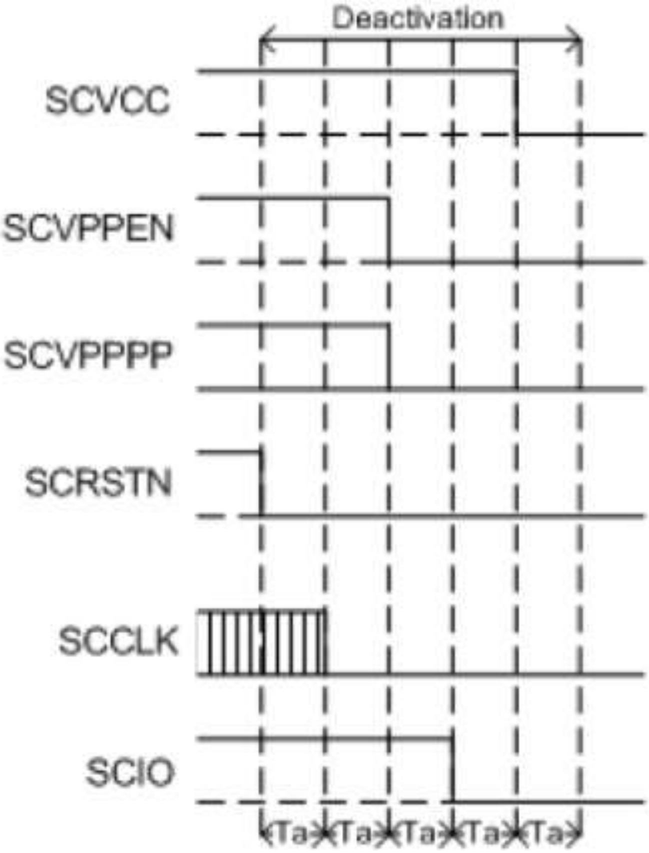


Fig. 26-4 Deactivation Sequence

26.4 Register description

26.4.1 Register Summary

Name	Offset	Size	Reset Value	Description

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 Guardtime
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
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*

26.4.2 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.

Bit	Attr	Reset Value	Description
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 Guardtime (minimum delay between the leading edges of two consecutive characters) is reduced to 11 ETU when T=1 protocol is used and Guardtime 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
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.

Bit	Attr	Reset Value	Description
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 deasserted. 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, iverse 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

Bit	Attr	Reset Value	Description
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
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.
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.

Bit	Attr	Reset Value	Description
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 guardtime 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.
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
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.

Bit	Attr	Reset Value	Description
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 guardtime 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.
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).

Bit	Attr	Reset Value	Description
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 Field0000 Description
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

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 guardtime

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 guardtime.

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 Guardtime

Bit	Attr	Reset Value	Description
7:0	RW	0x00	SCGUTI Smart Card Guardtime 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

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

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 16 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

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

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.

26.5 Interface Description

Table 26-1 IOMUX Setting

Module Pin	IO	Pad Name	IOMUX Setting
scclk_0	O	I2C1SENSORscl_SCclk_GPIO1830gpio8a	GPIO8A_IOMUX[11:10]= 2'b10
scclk_1	O	SPI2txd_SCclk_GPIO1830gpio8b1	GPIO8B_IOMUX[3:2]= 2'b10
scrst_0	O	I2C1SENSORsda_SCrst_GPIO1830gpio8a4	GPIO8A_IOMUX[9:8]= 2'b10
scrst_1	O	SPI2rxd_SCrst_GPIO1830gpio8b0	GPIO8B_IOMUX[1:0]= 2'b10
scvcc18v	O	PS2clk_SCvcc18v_GPIO1830gpio8a0	GPIO8A_IOMUX[1:0]= 2'b10
scvcc33v	O	PS2data_SCvcc33v_GPIO1830gpio8a1	GPIO8A_IOMUX[3:2]= 2'b10
scdetect_0	I	SPI2csn0_SCdetect_GPIO1830gpio8a7	GPIO8A_IOMUX[15:14]= 2'b10
scdetect_1	I	SCdetectt1_GPIO1830gpio8a2	GPIO8A_IOMUX[5:4]= 2'b01
sci0_0	I/O	SPI2clk_SCio_GPIO1830gpio8a6	GPIO8A_IOMUX[13:12]= 2'b10
sci0_1	I/O	SPI2csn1_SCiot1_GPIO1830gpio8a3	GPIO8A_IOMUX[7:6]= 2'b10

Notes: 1. I=input, O=output, I/O=input/output, bidirectional

2. Pull up scio pin when data transaction

26.6 Application Notes

26.6.1 SCR Clock

The Smart Card Clock signal is used as the main clock for the smart card. Its frequency can be adjusted 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:

$$\mathrm{SCCLK}_{\mathrm{freq}} = \frac{CLK\_freq}{2*(SCCDIV+1)}, \qquad SCCDIV \cong \frac{CLK_{freq}}{2*SCCLK_{freq}} - 1$$

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 Card Reader 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:

$$\mathrm{BAUD}_{\mathrm{rate}} = \frac{CLK\_freq}{2*(BAUDDIV+1)}$$

The duration of one bit, Elementary Time Unit (ETU) and parameters F and D are defined in the ISO/IEC 7816-3 specification.

$$\frac{1}{BAUD\_rate} \cong ETU = \frac{F}{D} * \frac{1}{SCCLK_{freq}}, \frac{F}{D} \cong \frac{BAUDDIV+1}{SCCDIV+1}$$

BAUDDIV equation based on SCCDIV value and Smart Card parameters F and D is following:

$$\mathrm{BAUDDIV} \cong (\mathrm{SCCDIV}+1) * \frac{F}{D} - 1$$

During the first answer to reset response after the cold reset, the initial ETU must be equal to 372 Smart Card 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 Card parameters F and D.

Baud Tune Register (BAUDTUNE) 3-bit value that can be used to increase the accuracy of the Baud Clock impulses timing by using the BAUDTUNE Increment from Table listed below in combination with BAUDDIV register value.

Table 26-2 BAUDTUNE register

BAUDTUNE	000	001	010	011	100	101	110	111
BAUDTUNE$_{\mathrm{INCR}}$	+0	+0.125	0.25	+0.375	+0.5	+0.625	+0.75	+0.875

$$\mathrm{BAUDDIV} + \mathrm{BAUDTUNE}_{INCR} \cong (SCCDIV+1) * \frac{F}{D} - 1$$

The BAUDDIV register value (nearest integer) can be computed using following equation:

$$\mathrm{BAUDDIV} \cong (\mathrm{SCCDIV}+1) * \frac{F}{D} - 1 - BAUDTUNE_{INCR}$$

Chapter 27 SARADC

27.1 Overview

The ADC is a 3-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 100KSPS with 1MHz A/D converter clock.

27.2 Block Diagram

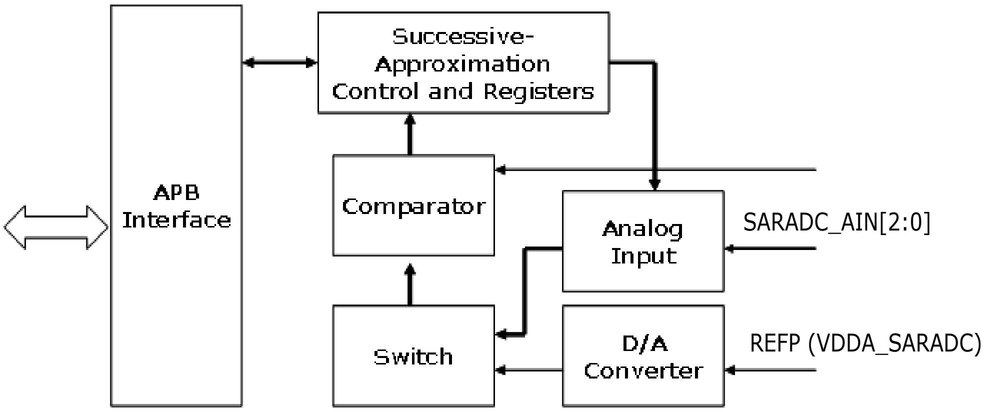


Fig. 27-1 RK3288 SAR-ADC block diagram

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.

27.3 Function description

In RK3288, 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.

27.4 Register Description

This section describes the control/status registers of the design.

27.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

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

27.4.2 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) 1’b0: ADC stop 1’b1: 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. 1’b0: Disable 1’b1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit 1’b0: ADC power down 1’b1: 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]). 3’b000: Input source 0 (SARADC_AIN[0]) 3’b001: Input source 1 (SARADC_AIN[1]) 3’b010: Input source 2 (SARADC_AIN[2]) 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	0x08	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

27.5 Timing Diagram

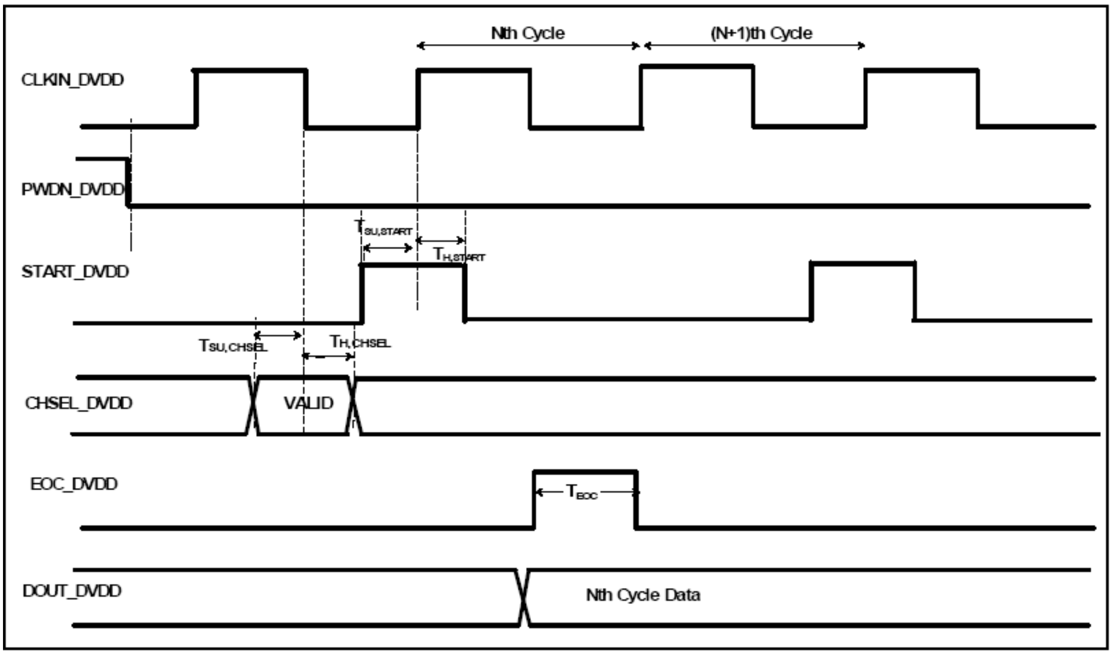
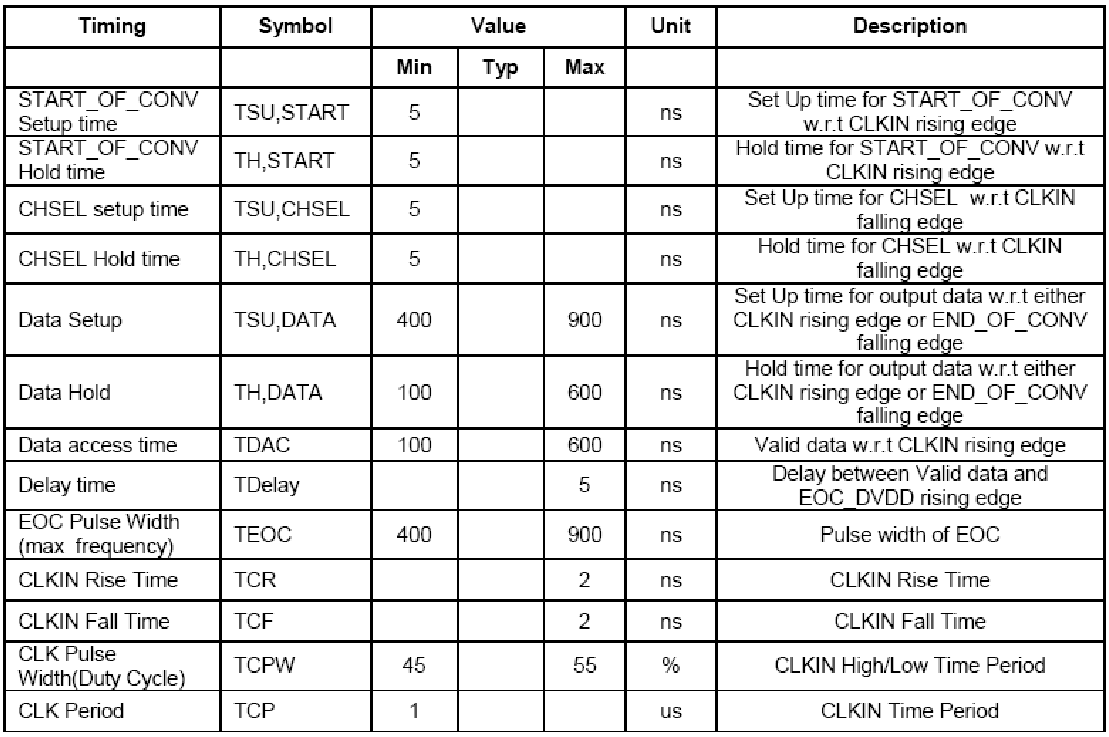


Fig. 27-2 SAR-ADC timing diagram in single-sample conversion mode

The following table has shows the detailed value for timing parameters in the above diagram.

Table 27-1 RK3288 SAR-ADC timing parameters list



Timing	Symbol	Value			Unit	Description
		Min	Typ	Max		
START_OF_CONV Setup time	TSU,START	5			ns	Set Up time for START_OF_CONV w.r.t CLKIN rising edge
START_OF_CONV Hold time	TH,START	5			ns	Hold time for START_OF_CONV w.r.t CLKIN rising edge
CHSEL setup time	TSU,CHSEL	5			ns	Set Up time for CHSEL w.r.t CLKIN falling edge
CHSEL Hold time	TH,CHSEL	5			ns	Hold time for CHSEL w.r.t CLKIN falling edge
Data Setup	TSU,DATA	400		900	ns	Set Up time for output data w.r.t either CLKIN rising edge or END_OF_CONV falling edge
Data Hold	TH,DATA	100		600	ns	Hold time for output data w.r.t either CLKIN rising edge or END_OF_CONV falling edge
Data access time	TDAC	100		600	ns	Valid data w.r.t CLKIN rising edge
Delay time	TDelay			5	ns	Delay between Valid data and EOC_DVDD rising edge
EOC Pulse Width (max frequency)	TEOC	400		900	ns	Pulse width of EOC
CLKIN Rise Time	TCR			2	ns	CLKIN Rise Time
CLKIN Fall Time	TCF			2	ns	CLKIN Fall Time
CLK Pulse Width(Duty Cycle)	TCPW	45		55	%	CLKIN High/Low Time Period
CLK Period	TCP	1			us	CLKIN Time Period

27.6 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 coversion is completed
- Read the conversion result from SARADC_DATA[9:0]

Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 28 GraphicsProcessing Unit (GPU)

28.1 Overview

With support for 2D graphics, 3D graphics, and GPGPU computing, the GPU of ARM Mali-T760MP4 provides a complete graphics acceleration platform based on open standards.

The GPU brings graphics power to a wide range of different products, from mobile user interfaces to tablets, HDTV, and mobile gaming. One single driver stack for all graphics configurations simplifies application porting, system integration, and maintenance.Scheduling and performance scaling are fully handled within the graphics system with no special considerations required from the application developer.

The GPU takes graphics instructions from software applications running on the application processor. It processes the graphics instructions and puts the results back into system memory. These results are sent to the framebuffer and on to the display device.

GPU supports the following features:

- Fully programmable architecture
- A rich API feature set with high-performance support for both shader-based and fixed-function graphics APIs.
- Anti-aliasing capabilities
- An effective core forGeneral Purpose computing on GPU(GPGPU) applications
- High memory bandwidth and low power consumption for 3D graphics content
- Scalability for products from smart phones to high-end mobile computing
- Performance leading 3D graphics
- Image quality using double-precision FP64, and anti-aliasing
- Standard bus interfaces
- An ACE-Lite interface to access external memory
- Easy integration
- Latency tolerance Configurable per-core power management for enabling the optimal power and performance combination for each application
- Coherency aware interconnects for system memory and resource sharing
- Frame buffer compression
- Supported API graphics industry standards
  - OpenGL ES 1.1 Specification at Khronos
  - OpenGL ES 2.0 Specification at Khronos
  - OpenGL ES 3.0 Specification at Khronos
  - OpenCL 1.0 Specification at Khronos,Full Profile specification
  - OpenCL 1.1 Specification at Khronos,Full Profile specification
  - DirectX 9 Specification
  - DirectX 11.1 Specification

28.2 Block Diagram

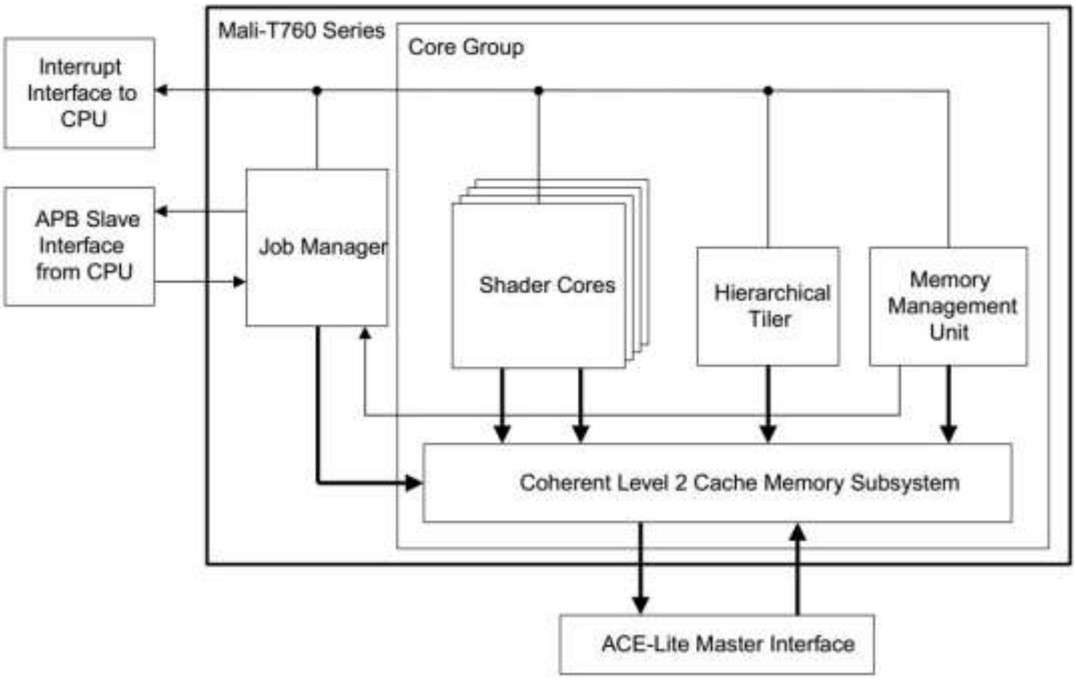

Mali-T760 Series
Core Group
Interrupt Interface to CPU
APB Slave Interface from CPU
Job Manager
Shader Cores
Hierarchical Tiler
Memory Management Unit
Coherent Level 2 Cache Memory Subsystem
ACE-Lite Master Interface

Fig. 28-1 GPU Block Diagram

28.3 Application Notes

28.3.1 Clock and Reset control

We can conifig CRU_CLKSEL34_CON[7:6] to select GPU AXI clock source among CPLL/GPLL/usbphy(480MHz)/NPLL,and then set CRU_CLKSEL34_CON[4:0] to divide clock source from 1:1 to 1:32.
We can config CRU_CLKGATE5_CON[7] to gate the gpu clock and config CRU_CLKGATE11_CON[13] to gate the gpu AXI clock.
We can reset gpu core by setting CRU_SOFTRST7_CON[8].

28.3.2 GPU interrupt

The GPU generates interrupts for job handling, memory management, and events not tied to a specific shader core. For these, the number of logical interrupts is individually controlled, but there is only a single physical interrupt line for each group.
The following top level interrupts are raised by the GPU:
The GPU_IRQGPU (GIC interrupt 40)
Exceptions that are not associated with specific jobs. These cannot be recovered.
GPU_IRQJOB (GIC interrupt 38)
Signals the completion or failure of a job running on the GPU.
GPU_IRQMMU (GIC interrupt 39)
Exceptions caused by memory management. These are potentially recoverable.
The application processor interprets the interrupts generated by the GPU . The software queries the states of the registers in the job manager to determine what must be done to handle the interrupt. The job manager is not required to wait for the result of the interrupt, it can be starting on the next set of jobs to be executed.