Rockchip
RK3308
Technical Reference Manual
Part1

Revision 1.1
Aug. 2018

Revision History

Date	Revision	Description
2018-8-10	1.1	Update some feature
2018-5-20	1.0	Initial Release

Table of Content

Figure Index

Table Index

Warranty Disclaimer

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

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

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

Copyright and Patent Right

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

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

Trademarks

Rockchip and Rockchip^{TM} 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.

Chapter 1 System Overview

1.1 Address Mapping

RK3308 boot from internal BootROM, which supports remap function by software programming. Remap is controlled by SGRF_SOC_CON0[8]. When remap is set to 1, the BootROM is un-accessible and System SRAM is mapped to address 0xffff0000.

Table 1-1 RK3308 Address Mapping

Addr	IP	Addr	IP	Addr	IP
ff210000	No Secure OTPC 64K	ff3d0000	Reserved 192K	ffff0000	BootROM(32K)/System SRAM(64K) 64K
ff200000	Reserved 64K	ff3c0000	VAD 64K	fffc0000	Reserved 192K
ff1f0000	TSADC CTL 64K	ff3c0000	VAD 64K	fff80000	System SRAM(256K) 256K
ff1e0000	SARADC CTL 64K	ff3b0000	SPDIF_8CH_RX 64K	ff840000	Reserved 7.25M
ff1c0000	Reserved 128K	ff3a0000	SPDIF_8CH_TX 64K	ff800000	A35 DEBUG 256K
ff1b0000	TIMER_6CH_1 64K	ff390000	Reserved 64K	ff600000	Reserved 2M
ff1a0000	TIMER_6CH_0 64K	ff380000	PDM_8CH 64K	ff5f0000	DDR FireWall 64K
ff190000	Reserved 64K	ff370000	Reserved 64K	ff5c0000	Interconnect Service 192K
ff180000	PWM_4CH 64K	ff360000	I2S_2CH_1 64K	ff590000	Reserved 192K
ff150000	Reserved 192K	ff350000	I2S_2CH_0 64K	ff580000	GIC 64K
ff140000	SPI2 64K	ff340000	Reserved 64K	ff570000	Reserved 64K
ff130000	SPI1 64K	ff330000	I2S_8CH_3 64K	ff560000	Audio Codec PHY 64K
ff120000	SPI0 64K	ff320000	I2S_8CH_2 64K	ff550000	CPU BOOST 64K
ff0f0000	Reserved 192K	ff310000	I2S_8CH_1 64K	ff540000	OTP Masker 64K
ff0e0000	UART4 64K	ff300000	I2S_8CH_0 64K	ff530000	DDR PHY 64K
ff0d0000	UART3 64K	ff2f0000	CRYPTO 64K	ff520000	PMU 64K
ff0c0000	UART2 64K	ff2e0000	VOP 64K	ff510000	Reserved 64K
ff0b0000	UART1 64K	ff2d0000	DMAC1 64K	ff500000	CRU 64K

Addr	IP	Addr	IP	Addr	IP
ff0a0000	UART0 64K	ff2c0000	DMAC0 64K	ff4f0000	Reserved 64K
ff090000	Reserved 64K	ff2b0000	SGRF 64K	ff4e0000	MAC 64K
ff080000	WDT 64K	ff2a8000	Secure OTPC 32K	ff4d0000	Reserved 64K
ff070000	I2C3 64K	ff2a0000	KEY Reader 32K	ff4c0000	SFC 64K
ff060000	I2C2 64K	ff290000	Secure DMAC1 64K	ff4b0000	NANDC 64K
ff050000	I2C1 64K	ff280000	Secure DMAC0 64K	ff4a0000	SDIO 64K
ff040000	I2C0 64K	ff270000	Reserved 64K	ff490000	EMMC 64K
ff030000	DDR STANDBY 64K	ff260000	GPIO4 64K	ff480000	SD/MMC 64K
ff020000	DFI MONITOR 64K	ff250000	GPIO3 64K	ff460000	Reserved 128K
ff010000	DDR UPCTL 64K	ff240000	GPIO2 64K	ff450000	USB2.0 HOST OHCI 64K
ff000000	GRF 64K	ff230000	GPIO1 64K	ff440000	USB2.0 HOST EHCI 64K
00000000	DDR 4080M	ff220000	GPIO0 64K	ff400000	USB2.0 OTG 256K

The following table shows the BootROM /System SRAM address before and after remapping. Before remapping, 0xfff80000 and 0xffff0000 is the base address for System SRAM and BootROM separately. After remapping, BootROM could not be accessed, both 0xfff80000 and 0xffff0000 are the base address for System SRAM and only 64KB System SRAM space could be accessed from 0xffff0000 base address.

Table 1-2 RK3308 remap function

Before remap		After remap	
fff80000	System SRAM(256K) 256K	fff80000/ffff0000	System SRAM(256K) 256K@fff80000 64K@ffff0000
ffff0000	BootROM(32K) 64K	N/A	BootROM(32K) N/A

1.2 System Boot

RK3308 provides system boot from off-chip devices such as NAND Flash, eMMC memory, Serial NAND or NOR Flash, SDMMC card. When boot code is not ready in these devices, the boot code can be programmed through USB OTG interface by running ROM code in BootROM. All of the ROM code is hard code in internal BootROM. Below figure shows the procedure of ROM code execution.

The following features are supported by ROM code.

- Support system boot from the following device:

  - Asynchronism NAND Flash, 8bit data width
  - eMMC Interface, 8bits data width
  - Serial NOR Flash, 1bit data width
  - SDMMC Card, 4bits data width
- Support system code download by USB OTG

Following figure shows RK3308 boot procedure flow.

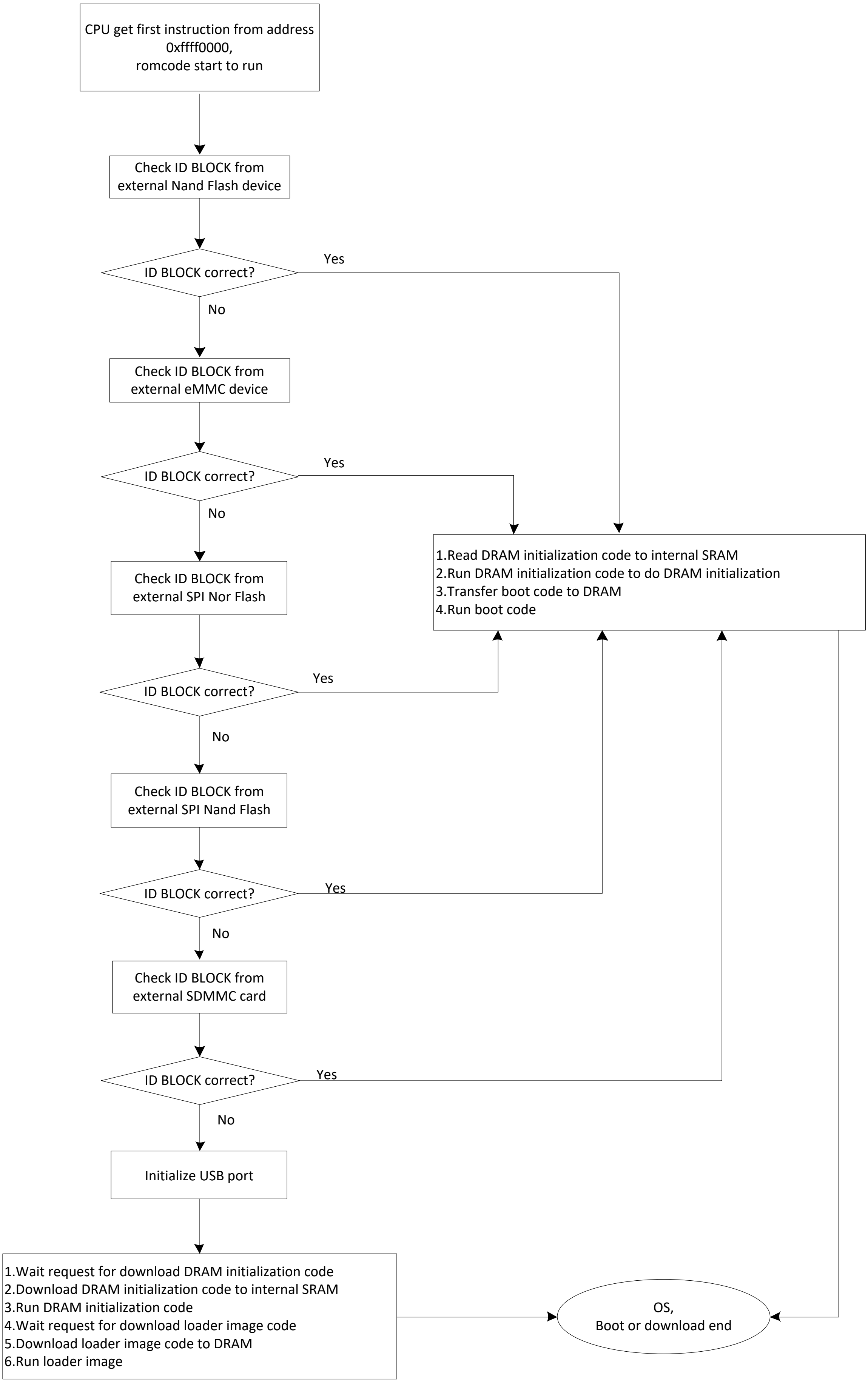

CPU get first instruction from address 0xffff0000, romcode start to run
Check ID BLOCK from external Nand Flash device
ID BLOCK correct?
Yes
No
Check ID BLOCK from external eMMC device
ID BLOCK correct?
Yes
No
Check ID BLOCK from external SPI Nor Flash
ID BLOCK correct?
Yes
No
Check ID BLOCK from external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

Fig. 1-1 RK3308 boot procedure flow

1.3 System Interrupt Connection

RK3308 provides an General Interrupt Controller(GIC) for CPU, which has 89 SPI (shared peripheral interrupts) interrupt sources and 4 PPI (Private peripheral interrupt) interrupt sources and separately generates one nIRQ and one nFIQ to each CPU Core. The triggered level type for each SPI interrupt is high and for each PPI interrupt is low, both of them is Non-programmable. The detailed interrupt sources connection is shown below. For detailed GIC setting, please refer to GIC Chapter.

Table 1-3 RK3308 Interrupt connection list

Interrupt Type	Interrupt ID	Source	Polarity
PPI	16	Reserved	Low level
	17	Reserved	Low level
	18	Reserved	Low level
	19	Reserved	Low level
	20	Reserved	Low level
	21	Reserved	Low level
	22	Reserved	Low level
	23	Reserved	Low level
	24	Reserved	Low level
	25	Reserved	Low level
	26	nCNTHPIRQ	Low level
	27	nCNTVIRQ	Low level
	28	Reserved	Low level
	29	nCNTPSIRQ	Low level
	30	nCNTPNSIRQ	Low level
	31	Reserved	Low level
SPI	32	DMAC0	High level
	33	DMAC0_IRQ_ABORT	High level
	34	DMAC1	High level
	35	DMAC1_IRQ_ABORT	High level
	36	Reserved	High level
	37	Reserved	High level
	38	Reserved	High level
	39	Reserved	High level
	40	DDR UPCTL	High level
	41	DFI MONITOR	High level
	42	WDT	High level
	43	I2C0	High level
	44	I2C1	High level
	45	I2C2	High level
	46	I2C3	High level
	47	SPI0	High level
	48	SPI1	High level
	49	SPI2	High level
	50	UART0	High level
	51	UART1	High level
	52	UART2	High level

Interrupt Type	Interrupt ID	Source	Polarity
	53	UART3	High level
	54	UART4	High level
	55	PWM_4CH	High level
	56	PWM_4CH_PWR	High level
	57	TIMER_6CH_0(0)	High level
	58	TIMER_6CH_0(1)	High level
	59	TIMER_6CH_0(2)	High level
	60	TIMER_6CH_0(3)	High level
	61	TIMER_6CH_0(4)	High level
	62	TIMER_6CH_0(5)	High level
	63	TIMER_6CH_1(0)	High level
	64	TIMER_6CH_1(1)	High level
	65	TIMER_6CH_1(2)	High level
	66	TIMER_6CH_1(3)	High level
	67	TIMER_6CH_1(4)	High level
	68	TIMER_6CH_1(5)	High level
	69	SARADC CTL	High level
	70	TSADC CTL	High level
	71	Reserved	High level
	72	GPIO0	High level
	73	GPIO1	High level
	74	GPIO2	High level
	75	GPIO3	High level
	76	GPIO4	High level
	77	CRYPTO	High level
	78	VOP_INTR	High level
	79	Reserved	High level
	80	I2S_8CH_0	High level
	81	I2S_8CH_1	High level
	82	I2S_8CH_2	High level
	83	I2S_8CH_3	High level
	84	I2S_2CH_0	High level
	85	I2S_2CH_1	High level
	86	PDM_8CH	High level
	87	SPDIF_8CH_TX	High level
	88	SPDIF_8CH_RX	High level
	89	VAD	High level
	90	Reserved	High level
	91	Reserved	High level
	92	SECURE_OTPC	High level
	93	NO_SECURE_OTPC	High level
	94	KEY_READER	High level
	95	OTP_MASKER	High level
	96	MAC_INTR	High level

Interrupt Type	Interrupt ID	Source	Polarity
	97	MAC_INTR_PMT	High level
	98	USB2.0 OTG	High level
	99	USB2.0 OTG BVALID	High level
	100	USB2.0 OTG ID	High level
	101	USB2.0 OTG LINESTATE	High level
	102	USB2.0 OTG DISCONNECT	High level
	103	USB2.0 HOST EHCI	High level
	104	USB2.0 HOST OHCI	High level
	105	USB2.0 HOST ARB	High level
	106	USB2.0 HOST LINESTATE	High level
	107	USB2.0 HOST DISCONNECT	High level
	108	SD/MMC	High level
	109	EMMC	High level
	110	SDIO	High level
	111	~ SD/MMC DETECT PIN	High level
	112	SD/MMC DETECT	High level
	113	NANDC	High level
	114	SFC	High level
	115	~nPMUIRQ[0]	High level
	116	~nPMUIRQ[1]	High level
	117	~nPMUIRQ[2]	High level
	118	~nPMUIRQ[3]	High level
	119	Reserved	High level
	120	Reserved	High level
	121	Reserved	High level
	122	Reserved	High level
	123	~nCOMMIRQ[0]	High level
	124	~nCOMMIRQ[1]	High level
	125	~nCOMMIRQ[2]	High level
	126	~nCOMMIRQ[3]	High level
	127	~nAXIERRIRQ	High level
	128	Reserved	High level
	129	Reserved	High level
	130	Reserved	High level
	131	Reserved	High level
	132	Reserved	High level
	133	Reserved	High level
	134	Reserved	High level
	135	Reserved	High level
	136	Reserved	High level
	137	Reserved	High level
	138	Reserved	High level
	139	Reserved	High level
	140	Reserved	High level

Interrupt Type	Interrupt ID	Source	Polarity
	141	Reserved	High level
	142	CPU_SLV_ERROR	High level
	143	PERF_INT_CA35	High level
	144	PMU	High level
	145	HWFFC_INTR	High level
	146	HPDET	High level
	147	Reserved	High level
	148	Reserved	High level

1.4 System DMA Hardware Request Connection

RK3308 provides two DMAC in the system. The handshake channel number between peripheral and DMAC is described in this section. For detailed descriptions of DMAC, please refer to DMAC Chapter

Table 1-4 RK3308 DMAC0 hardware request connection list

Req Number	Source	Polarity
0	SPI0 tx	High level
1	SPI0 rx	High level
2	SPI1 tx	High level
3	SPI1 rx	High level
4	UART0 tx	High level
5	UART0 rx	High level
6	UART1 tx	High level
7	UART1 rx	High level
8	UART2 tx	High level
9	UART2 rx	High level
10	UART3 tx	High level
11	UART3 rx	High level

Table 1-5 RK3308 DMAC1 hardware request connection list

Req Number	Source	Polarity
0	I2S_8CH_0 tx	High level
1	I2S_8CH_0 rx	High level
2	I2S_8CH_1 tx	High level
3	I2S_8CH_1 rx	High level
4	I2S_8CH_2 tx	High level
5	I2S_8CH_2 rx	High level
6	Reserved	High level
7	I2S_8CH_3 rx	High level
8	I2S_2CH_0 tx	High level
9	I2S_2CH_0 rx	High level
10	Reserved	High level
11	I2S_2CH_1 rx	High level
12	PDM_8CH rx	High level
13	SPDIF_8CH_TX tx	High level
14	SPDIF_8CH_RX rx	High level

Req Number	Source	Polarity
15	PWM_4CH rx	High level
16	SPI2 tx	High level
17	SPI2 rx	High level
18	UART4 tx	High level
19	UART4 rx	High level

Chapter 2 Clock & Reset Unit (CRU)

2.1 Overview

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

- Compliance to the AMBA APB interface
- Embedded 4 PLLs
- Flexible selection of clock source
- Supports the respective divided clocks
- Supports the respective gating of all clocks
- Supports the respective software reset of all modules

2.2 Block Diagram

CRU comprises with:

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

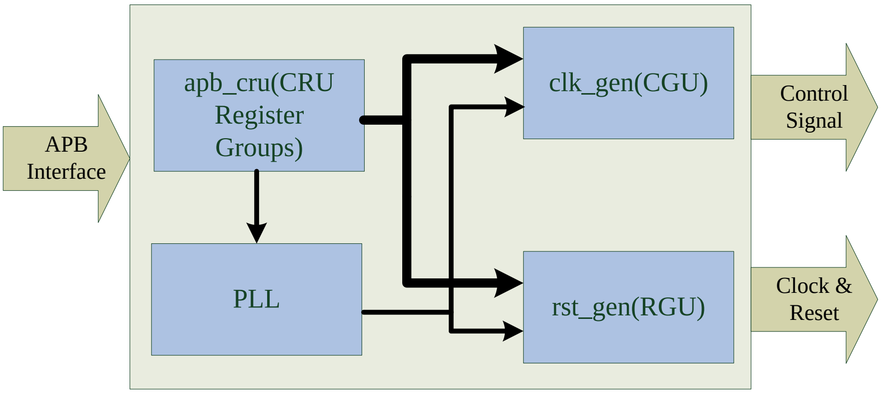


Fig. 2-1 CRU Block Diagram

2.3 Function Description

2.3.1 PLL

There are 4 PLLs in the chip: ARM PLL, DDR PLL, VOICE PLL0 and VOICE PLL1.
PLL is a general purpose, high-performance PLL-based clock generator. The PLL is a multi-function, general purpose frequency synthesizer. Ultra-wide input and output ranges along with best-in-class jitter performance allow the PLL to be used for almost any clocking application. With excellent supply noise immunity, the PLL is ideal for use in noisy mixed signal SoC environments.
All PLL supports the following features:

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

PLL block diagram is in following figure.

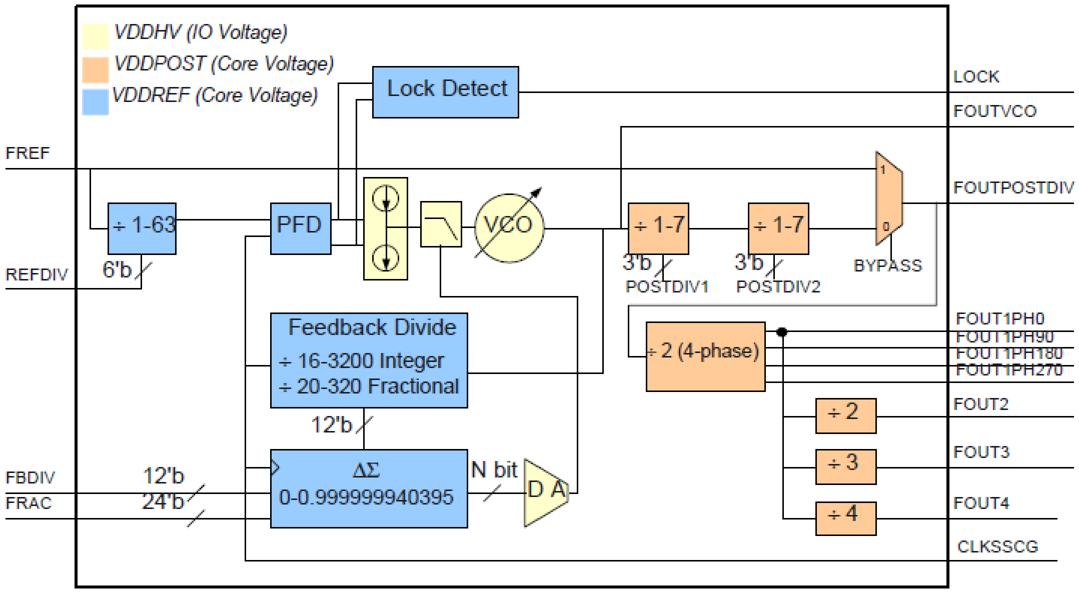


Fig. 2-2 PLL Block Diagram

The REF(reference clock) of all PLL is xin_osc0(24MHz crystal oscillator), and only the output clock FOUTPOSTDIV(PLL post divided output) is used in the SoC.

The setting(REFDIV, FBDIV, FRAC, POSTDIV1 and POSTDIV2) of all PLL are controlled by CRU registers. A submodule HWFFC (hardware fast frequency change) is designed for changing the APLL frequency with few software interactions. So the setting of APLL also can be controlled by hardware, and the actually setting can be read by some shadow registers when HWFFC mode is enabled.

For each PLL, a submodule SSMOD(Spread spectrum modulator) is designed to work with PLL to produce SSCG(spread spectrum clock generation). SSCG is used to reduce peak EMI. SSMODE can use internal or external wave table, and the external wave table are controlled by CRU registers.

2.3.2 Reset Generate Unit(RGU)

RGU generate the reset of all module in SoC. Almost all module have 6 reset source as the following figure shows. The 'xxx' in the figure is the module name.

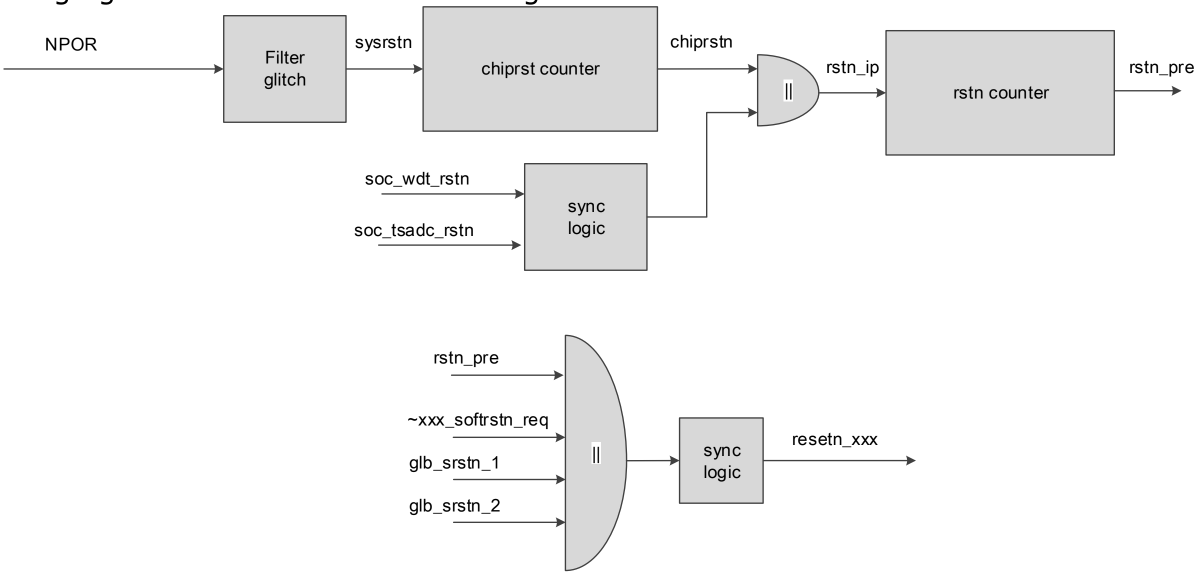


Fig. 2-3 Reset Architecture Diagram

The reset source are:

- NPOR(Negative Power On Reset): hardware reset from IO.
- oc_wdt_rstn: reset from WDT IP in SoC.
- soc_tsadc_rstn: reset from TS-ADC IP in SoC.
- xxx_softrstn_req: software reset request by programming CRU register SOFTRST_CONx. each bit of SOFTRST_CONx is separated to different module.
- glb_srstn_1: global software first reset by programming CRU register

CRU_GLB_SRST_FST as 0xfdb9

- glb_srstn_2: global software second reset by programming CRU register CRU_GLB_SRST_SND as 0xeca8

The two global software resets will be self-cleared by hardware. glb_srstn_1 will reset the all logic, and glb_srstn_2 will reset the all logic except GRF, SGRF and all GPIOs.

2.3.3 Clock Generate Unit(CGU)

CGU generate the clock of all modules in SoC. A full architecture with basic unit and clock source is as following figure. For each module clock, it may be part of full architecture base the requirement.

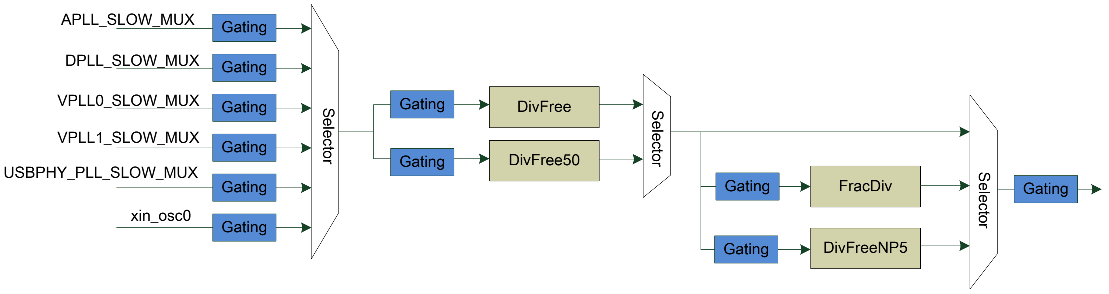


Fig. 2-4 Clock Architecture Diagram

The basic units are:

- Gating
- Selector
- Divfree(Glitch free divider)
  - clk_out_freq=clk_in_freq/divisor
  - When divisor is even, the clock duty cycle of clk_out is 50%
  - When divisor is odd, the clock duty cycle of clk_out is not 50%
- Fracdiv(Fractional divider)
  - clk_out_freq=clk_in_freq*numerator/denominator, both numerator and denominator are 16 bits
- Divfree50(Glitch free divider for duty cycle 50%)
  - clk_out_freq=clk_in_freq/divisor
  - When divisor is even or odd, the clock duty cycle of clk_out is 50%
- DivFreeNP5(Glitch free divider for null point 5)
  - clk_out_freq= 2*clk_in_freq/(2*div_con+3)
  - The clock duty cycle of clk_out is not 50%

The setting of all basic units are controlled by CRU registers. Especial for core, the setting also can be controlled by submodule HWFFC, the actually setting can be read by some shadow registers when HWFFC mode is enabled.

2.4 Register Description

2.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_APLL_CON0	0x0000	W	0x00003064	APLL configuration register0
CRU_APLL_CON1	0x0004	W	0x00001041	APLL configuration register1
CRU_APLL_CON2	0x0008	W	0x00000001	APLL configuration register2
CRU_APLL_CON3	0x000c	W	0x00000007	APLL configuration register3
CRU_APLL_CON4	0x0010	W	0x00007f00	APLL configuration register4
CRU_DPLL_CON0	0x0020	W	0x00002064	DPLL configuration register0
CRU_DPLL_CON1	0x0024	W	0x00001041	DPLL configuration register1
CRU_DPLL_CON2	0x0028	W	0x00000001	DPLL configuration register2
CRU_DPLL_CON3	0x002c	W	0x00000007	DPLL configuration register3

Name	Offset	Size	Reset Value	Description
CRU_DPLL_CON4	0x0030	W	0x00007f00	DPLL configuration register4
CRU_VPLL0_CON0	0x0040	W	0x00001028	VPLL0 configuration register0
CRU_VPLL0_CON1	0x0044	W	0x00000041	VPLL0 configuration register1
CRU_VPLL0_CON2	0x0048	W	0x00f5c28f	VPLL0 configuration register2
CRU_VPLL0_CON3	0x004c	W	0x00000007	VPLL0 configuration register3
CRU_VPLL0_CON4	0x0050	W	0x00007f00	VPLL0 configuration register4
CRU_VPLL1_CON0	0x0060	W	0x00001021	VPLL1 configuration register0
CRU_VPLL1_CON1	0x0064	W	0x00000041	VPLL1 configuration register1
CRU_VPLL1_CON2	0x0068	W	0x00de69ad	VPLL1 configuration register2
CRU_VPLL1_CON3	0x006c	W	0x00000007	VPLL1 configuration register3
CRU_VPLL1_CON4	0x0070	W	0x00007f00	VPLL1 configuration register4
CRU_MODE	0x00a0	W	0x00000000	MODE register
CRU_MISC	0x00a4	W	0x00000000	MISC register
CRU_GLB_CNT_TH	0x00b0	W	0x00000000	Global reset counter threshold register
CRU_GLB_RST_ST	0x00b4	W	0x00000000	Global reset state register
CRU_GLB_SRST_FST	0x00b8	W	0x00000000	Global software first reset register
CRU_GLB_SRST_SND	0x00bc	W	0x00000000	Global software second reset register
CRU_GLB_RST_CON	0x00c0	W	0x00000000	Global reset control register
CRU_GLB_PLL_LOCK	0x00c4	W	0x00003a98	Global PLL lock register
CRU_HWFFC_CON0	0x00e0	W	0x00000000	Hardware fast frequency change control register0
CRU_HWFFC_TH	0x00e8	W	0x0000030f	Hardware fast frequency change threshold register
CRU_HWFFC_INTSTS	0x00ec	W	0x00000000	Hardware fast frequency change interrupt status register
CRU_APLL_CON0_S	0x00f0	W	0x00003064	APLL configuration shadow register0
CRU_APLL_CON1_S	0x00f4	W	0x00001041	APLL configuration shadow register1
CRU_CLKSEL_CON0_S	0x00f8	W	0x00001300	Internal clock select and divide shadow register0
CRU_CLKSEL_CON0	0x0100	W	0x00001300	Internal clock select and divide register0
CRU_CLKSEL_CON1	0x0104	W	0x00000080	Internal clock select and divide register1
CRU_CLKSEL_CON2	0x0108	W	0x00000300	Internal clock select and divide register2
CRU_CLKSEL_CON3	0x010c	W	0x00000000	Internal clock select and divide register3
CRU_CLKSEL_CON4	0x0110	W	0x00007530	Internal clock select and divide register4

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON5	0x0114	W	0x00000005	Internal clock select and divide register5
CRU_CLKSEL_CON6	0x0118	W	0x00000b0b	Internal clock select and divide register6
CRU_CLKSEL_CON7	0x011c	W	0x00000305	Internal clock select and divide register7
CRU_CLKSEL_CON8	0x0120	W	0x00000007	Internal clock select and divide register8
CRU_CLKSEL_CON9	0x0124	W	0x00000000	Internal clock select and divide register9
CRU_CLKSEL_CON10	0x0128	W	0x0000000b	Internal clock select and divide register10
CRU_CLKSEL_CON11	0x012c	W	0x0000000b	Internal clock select and divide register11
CRU_CLKSEL_CON12	0x0130	W	0x00000000	Internal clock select and divide register12
CRU_CLKSEL_CON13	0x0134	W	0x0000000b	Internal clock select and divide register13
CRU_CLKSEL_CON14	0x0138	W	0x0000000b	Internal clock select and divide register14
CRU_CLKSEL_CON15	0x013c	W	0x00000000	Internal clock select and divide register15
CRU_CLKSEL_CON16	0x0140	W	0x0000000b	Internal clock select and divide register16
CRU_CLKSEL_CON17	0x0144	W	0x0000000b	Internal clock select and divide register17
CRU_CLKSEL_CON18	0x0148	W	0x00000000	Internal clock select and divide register18
CRU_CLKSEL_CON19	0x014c	W	0x0000000b	Internal clock select and divide register19
CRU_CLKSEL_CON20	0x0150	W	0x0000000b	Internal clock select and divide register20
CRU_CLKSEL_CON21	0x0154	W	0x00000000	Internal clock select and divide register21
CRU_CLKSEL_CON22	0x0158	W	0x0000000b	Internal clock select and divide register22
CRU_CLKSEL_CON23	0x015c	W	0x0000000b	Internal clock select and divide register23
CRU_CLKSEL_CON24	0x0160	W	0x00000000	Internal clock select and divide register24
CRU_CLKSEL_CON25	0x0164	W	0x00000005	Internal clock select and divide register25

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON26	0x0168	W	0x00000005	Internal clock select and divide register26
CRU_CLKSEL_CON27	0x016c	W	0x00000005	Internal clock select and divide register27
CRU_CLKSEL_CON28	0x0170	W	0x00000005	Internal clock select and divide register28
CRU_CLKSEL_CON29	0x0174	W	0x0000000b	Internal clock select and divide register29
CRU_CLKSEL_CON30	0x0178	W	0x0000000b	Internal clock select and divide register30
CRU_CLKSEL_CON31	0x017c	W	0x0000000b	Internal clock select and divide register31
CRU_CLKSEL_CON32	0x0180	W	0x0000000b	Internal clock select and divide register32
CRU_CLKSEL_CON33	0x0184	W	0x00000001	Internal clock select and divide register33
CRU_CLKSEL_CON34	0x0188	W	0x00000017	Internal clock select and divide register34
CRU_CLKSEL_CON35	0x018c	W	0x00000010	Internal clock select and divide register35
CRU_CLKSEL_CON36	0x0190	W	0x00000005	Internal clock select and divide register36
CRU_CLKSEL_CON37	0x0194	W	0x00000b0b	Internal clock select and divide register37
CRU_CLKSEL_CON38	0x0198	W	0x00000007	Internal clock select and divide register38
CRU_CLKSEL_CON39	0x019c	W	0x00000003	Internal clock select and divide register39
CRU_CLKSEL_CON40	0x01a0	W	0x00000003	Internal clock select and divide register40
CRU_CLKSEL_CON41	0x01a4	W	0x00000003	Internal clock select and divide register41
CRU_CLKSEL_CON42	0x01a8	W	0x00004004	Internal clock select and divide register42
CRU_CLKSEL_CON43	0x01ac	W	0x00008017	Internal clock select and divide register43
CRU_CLKSEL_CON44	0x01b0	W	0x0000009d	Internal clock select and divide register44
CRU_CLKSEL_CON45	0x01b4	W	0x00000909	Internal clock select and divide register45
CRU_CLKSEL_CON46	0x01b8	W	0x0000000f	Internal clock select and divide register46

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON47	0x01bc	W	0x00000000	Internal clock select and divide register47
CRU_CLKSEL_CON48	0x01c0	W	0x00000009	Internal clock select and divide register48
CRU_CLKSEL_CON49	0x01c4	W	0x00000000	Internal clock select and divide register49
CRU_CLKSEL_CON50	0x01c8	W	0x00000003	Internal clock select and divide register50
CRU_CLKSEL_CON51	0x01cc	W	0x00000000	Internal clock select and divide register51
CRU_CLKSEL_CON52	0x01d0	W	0x00000111	Internal clock select and divide register52
CRU_CLKSEL_CON53	0x01d4	W	0x00000000	Internal clock select and divide register53
CRU_CLKSEL_CON54	0x01d8	W	0x00000013	Internal clock select and divide register54
CRU_CLKSEL_CON55	0x01dc	W	0x00000000	Internal clock select and divide register55
CRU_CLKSEL_CON56	0x01e0	W	0x00000111	Internal clock select and divide register56
CRU_CLKSEL_CON57	0x01e4	W	0x00000000	Internal clock select and divide register57
CRU_CLKSEL_CON58	0x01e8	W	0x00000013	Internal clock select and divide register58
CRU_CLKSEL_CON59	0x01ec	W	0x00000000	Internal clock select and divide register59
CRU_CLKSEL_CON60	0x01f0	W	0x00000111	Internal clock select and divide register60
CRU_CLKSEL_CON61	0x01f4	W	0x00000000	Internal clock select and divide register61
CRU_CLKSEL_CON62	0x01f8	W	0x00000013	Internal clock select and divide register62
CRU_CLKSEL_CON63	0x01fc	W	0x00000000	Internal clock select and divide register63
CRU_CLKSEL_CON64	0x0200	W	0x00000111	Internal clock select and divide register64
CRU_CLKSEL_CON65	0x0204	W	0x00000000	Internal clock select and divide register65
CRU_CLKSEL_CON66	0x0208	W	0x00000013	Internal clock select and divide register66
CRU_CLKSEL_CON67	0x020c	W	0x00000000	Internal clock select and divide register67

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON68	0x0210	W	0x00000013	Internal clock select and divide register68
CRU_CLKSEL_CON69	0x0214	W	0x00000000	Internal clock select and divide register69
CRU_CLKSEL_CON70	0x0218	W	0x00000013	Internal clock select and divide register70
CRU_CLKSEL_CON71	0x021c	W	0x00000000	Internal clock select and divide register71
CRU_CLKSEL_CON72	0x0220	W	0x00000031	Internal clock select and divide register72
CRU_CLKSEL_CON73	0x0224	W	0x0000001f	Internal clock select and divide register73
CRU_CLKGATE_CON0	0x0300	W	0x00000000	Internal clock gating register0
CRU_CLKGATE_CON1	0x0304	W	0x00000000	Internal clock gating register1
CRU_CLKGATE_CON2	0x0308	W	0x00000000	Internal clock gating register2
CRU_CLKGATE_CON3	0x030c	W	0x00000000	Internal clock gating register3
CRU_CLKGATE_CON4	0x0310	W	0x00000000	Internal clock gating register4
CRU_CLKGATE_CON5	0x0314	W	0x00000000	Internal clock gating register5
CRU_CLKGATE_CON6	0x0318	W	0x00000000	Internal clock gating register6
CRU_CLKGATE_CON7	0x031c	W	0x00000000	Internal clock gating register7
CRU_CLKGATE_CON8	0x0320	W	0x00000000	Internal clock gating register8
CRU_CLKGATE_CON9	0x0324	W	0x00000000	Internal clock gating register9
CRU_CLKGATE_CON10	0x0328	W	0x00000000	Internal clock gating register10
CRU_CLKGATE_CON11	0x032c	W	0x00000000	Internal clock gating register11
CRU_CLKGATE_CON12	0x0330	W	0x00000000	Internal clock gating register12
CRU_CLKGATE_CON13	0x0334	W	0x00000000	Internal clock gating register13
CRU_CLKGATE_CON14	0x0338	W	0x00000000	Internal clock gating register14
CRU_SSCGTBL0_3	0x0380	W	0x00000000	SSCG external wave table register0
CRU_SSCGTBL4_7	0x0384	W	0x00000000	SSCG external wave table register1
CRU_SSCGTBL8_11	0x0388	W	0x00000000	SSCG external wave table register2
CRU_SSCGTBL12_15	0x038c	W	0x00000000	SSCG external wave table register3
CRU_SSCGTBL16_19	0x0390	W	0x00000000	SSCG external wave table register4
CRU_SSCGTBL20_23	0x0394	W	0x00000000	SSCG external wave table register5
CRU_SSCGTBL24_27	0x0398	W	0x00000000	SSCG external wave table register6
CRU_SSCGTBL28_31	0x039c	W	0x00000000	SSCG external wave table register7

Name	Offset	Size	Reset Value	Description
CRU_SSCGTBL32_35	0x03a0	W	0x00000000	SSCG external wave table register8
CRU_SSCGTBL36_39	0x03a4	W	0x00000000	SSCG external wave table register9
CRU_SSCGTBL40_43	0x03a8	W	0x00000000	SSCG external wave table register10
CRU_SSCGTBL44_47	0x03ac	W	0x00000000	SSCG external wave table register11
CRU_SSCGTBL48_51	0x03b0	W	0x00000000	SSCG external wave table register12
CRU_SSCGTBL52_55	0x03b4	W	0x00000000	SSCG external wave table register13
CRU_SSCGTBL56_59	0x03b8	W	0x00000000	SSCG external wave table register14
CRU_SSCGTBL60_63	0x03bc	W	0x00000000	SSCG external wave table register15
CRU_SSCGTBL64_67	0x03c0	W	0x00000000	SSCG external wave table register16
CRU_SSCGTBL68_71	0x03c4	W	0x00000000	SSCG external wave table register17
CRU_SSCGTBL72_75	0x03c8	W	0x00000000	SSCG external wave table register18
CRU_SSCGTBL76_79	0x03cc	W	0x00000000	SSCG external wave table register19
CRU_SSCGTBL80_83	0x03d0	W	0x00000000	SSCG external wave table register20
CRU_SSCGTBL84_87	0x03d4	W	0x00000000	SSCG external wave table register21
CRU_SSCGTBL88_91	0x03d8	W	0x00000000	SSCG external wave table register22
CRU_SSCGTBL92_95	0x03dc	W	0x00000000	SSCG external wave table register23
CRU_SSCGTBL96_99	0x03e0	W	0x00000000	SSCG external wave table register24
CRU_SSCGTBL100_103	0x03e4	W	0x00000000	SSCG external wave table register25
CRU_SSCGTBL104_107	0x03e8	W	0x00000000	SSCG external wave table register26
CRU_SSCGTBL108_111	0x03ec	W	0x00000000	SSCG external wave table register27
CRU_SSCGTBL112_115	0x03f0	W	0x00000000	SSCG external wave table register28

Name	Offset	Size	Reset Value	Description
CRU_SSCGTBL116_119	0x03f4	W	0x00000000	SSCG external wave table register29
CRU_SSCGTBL120_123	0x03f8	W	0x00000000	SSCG external wave table register30
CRU_SSCGTBL124_127	0x03fc	W	0x00000000	SSCG external wave table register31
CRU_SOFTRST_CON0	0x0400	W	0x00000000	Internal software reset control register0
CRU_SOFTRST_CON1	0x0404	W	0x00000000	Internal software reset control register1
CRU_SOFTRST_CON2	0x0408	W	0x00000000	Internal software reset control register2
CRU_SOFTRST_CON3	0x040c	W	0x00000000	Internal software reset control register3
CRU_SOFTRST_CON4	0x0410	W	0x00000000	Internal software reset control register4
CRU_SOFTRST_CON5	0x0414	W	0x00000000	Internal software reset control register5
CRU_SOFTRST_CON6	0x0418	W	0x00000000	Internal software reset control register6
CRU_SOFTRST_CON7	0x041c	W	0x00000000	Internal software reset control register7
CRU_SOFTRST_CON8	0x0420	W	0x00000000	Internal software reset control register8
CRU_SOFTRST_CON9	0x0424	W	0x00000000	Internal software reset control register9
CRU_SDMMC_CON0	0x0480	W	0x00000004	SDMMC control register0
CRU_SDMMC_CON1	0x0484	W	0x00000000	SDMMC control register1
CRU_SDIO_CON0	0x0488	W	0x00000004	SDIO control register0
CRU_SDIO_CON1	0x048c	W	0x00000000	SDIO control register1
CRU_EMMC_CON0	0x0490	W	0x00000004	EMMC control register0
CRU_EMMC_CON1	0x0494	W	0x00000000	EMMC control register1

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

2.4.2 Detail Register Description

CRU_APLL_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass REF is bypassed to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x3	postdiv1 PLL post divide 1 setting, 1-7 is valid postdiv1 should be greater than or equal to postdiv2
11:0	RW	0x064	fbdiv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: no plus one operation

CRU_APLL_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel 1'b0: PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted 1'b1: PLL can be power down only by pllpd1
14	RW	0x0	pllpd1 PLL power down 1 setting, it's valid when pllpdsel=1'b1 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL power down 0 setting, it's valid when pllpdsel=1'b0 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd Power down delta-sigma modulator 1'b0: no power down, PLL is fractional mode 1'b1: power down, PLL is integer mode
11	RO	0x0	reserved
10	RO	0x0	pll_lock 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x1	postdiv2 PLL post divide 2 setting, 1-7 is valid
5:0	RW	0x01	refdiv Reference divide value, 1-63 is valid

CRU_APLL_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down of 4 phase clock generator 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd VCO rate output clock power down 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Post divide power down 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down noise canceling DAC in FRAC mode 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional portion of feedback divide value

CRU_APLL_CON3

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread Set modulation depth(spread percentage) of SSMOD 5'h0: 0% 5'h1: 0.1% 5'h2: 0.2% ... 5'h1e: 3% 5'h1f: 3.1%

Bit	Attr	Reset Value	Description
7:4	WO	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	WO	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset SSMOD reset 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Disable sperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	WO	0x1	ssmod_bp SSMOD bypass 1'b0: no bypass 1'b1: bypass

CRU_APLL_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU_DPLL_CON0

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass REF is bypassed to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x2	postdiv1 PLL post divide 1 setting, 1-7 is valid postdiv1 should be greater than or equal to postdiv2
11:0	RW	0x064	fbdiv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: no plus one operation

CRU_DPLL_CON1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel 1'b0: PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted 1'b1: PLL can be power down only by pllpd1
14	RW	0x0	pllpd1 PLL power down 1 setting, it's valid when pllpdsel=1'b1 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL power down 0 setting, it's valid when pllpdsel=1'b0 1'b0: no power down 1'b1: power down
12	RW	0x1	dsmpd Power down delta-sigma modulator 1'b0: no power down, PLL is fractional mode 1'b1: power down, PLL is integer mode
11	RO	0x0	reserved
10	RO	0x0	pll_lock 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x1	postdiv2 PLL post divide 2 setting, 1-7 is valid
5:0	RW	0x01	refdiv Reference divide value, 1-63 is valid

CRU_DPLL_CON2

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down of 4 phase clock generator 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd VCO rate output clock power down 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Post divide power down 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down noise canceling DAC in FRAC mode 1'b0: no power down 1'b1: power down
23:0	RW	0x000001	fracdiv Fractional portion of feedback divide value

CRU_DPLL_CON3

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread Set modulation depth(spread percentage) of SSMOD 5'h0: 0% 5'h1: 0.1% 5'h2: 0.2% ... 5'h1e: 3% 5'h1f: 3.1%

Bit	Attr	Reset Value	Description
7:4	WO	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	WO	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset SSMOD reset 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Disable sperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	WO	0x1	ssmod_bp SSMOD bypass 1'b0: no bypass 1'b1: bypass

CRU_DPLL_CON4

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU_VPLL0_CON0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass REF is bypassed to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 PLL post divide 1 setting, 1-7 is valid postdiv1 should be greater than or equal to postdiv2
11:0	RW	0x028	fbdiv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: no plus one operation

CRU_VPLL0_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel 1'b0: PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted 1'b1: PLL can be power down only by pllpd1
14	RW	0x0	pllpd1 PLL power down 1 setting, it's valid when pllpdsel=1'b1 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL power down 0 setting, it's valid when pllpdsel=1'b0 1'b0: no power down 1'b1: power down
12	RW	0x0	dsmpd Power down delta-sigma modulator 1'b0: no power down, PLL is fractional mode 1'b1: power down, PLL is integer mode
11	RO	0x0	reserved
10	RO	0x0	pll_lock 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x1	postdiv2 PLL post divide 2 setting, 1-7 is valid
5:0	RW	0x01	refdiv Reference divide value, 1-63 is valid

CRU_VPLL0_CON2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down of 4 phase clock generator 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd VCO rate output clock power down 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Post divide power down 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down noise canceling DAC in FRAC mode 1'b0: no power down 1'b1: power down
23:0	RW	0xf5c28f	fracdiv Fractional portion of feedback divide value

CRU_VPLL0_CON3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread Set modulation depth(spread percentage) of SSMOD 5'h0: 0% 5'h1: 0.1% 5'h2: 0.2% ... 5'h1e: 3% 5'h1f: 3.1%

Bit	Attr	Reset Value	Description
7:4	WO	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	WO	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset SSMOD reset 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Disable sperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	WO	0x1	ssmod_bp SSMOD bypass 1'b0: no bypass 1'b1: bypass

CRU_VPLL0_CON4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU_VPLL1_CON0

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	bypass REF is bypassed to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RW	0x1	postdiv1 PLL post divide 1 setting, 1-7 is valid postdiv1 should be greater than or equal to postdiv2
11:0	RW	0x021	fbdiv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: no plus one operation

CRU_VPLL1_CON1

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pllpdsel 1'b0: PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted 1'b1: PLL can be power down only by pllpd1
14	RW	0x0	pllpd1 PLL power down 1 setting, it's valid when pllpdsel=1'b1 1'b0: no power down 1'b1: power down
13	RW	0x0	pllpd0 PLL power down 0 setting, it's valid when pllpdsel=1'b0 1'b0: no power down 1'b1: power down
12	RW	0x0	dsmpd Power down delta-sigma modulator 1'b0: no power down, PLL is fractional mode 1'b1: power down, PLL is integer mode
11	RO	0x0	reserved
10	RO	0x0	pll_lock 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x1	postdiv2 PLL post divide 2 setting, 1-7 is valid
5:0	RW	0x01	refdiv Reference divide value, 1-63 is valid

CRU_VPLL1_CON2

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	fout4phasepd Power down of 4 phase clock generator 1'b0: no power down 1'b1: power down
26	RW	0x0	foutvcopd VCO rate output clock power down 1'b0: no power down 1'b1: power down
25	RW	0x0	foutpostdivpd Post divide power down 1'b0: no power down 1'b1: power down
24	RW	0x0	dacpd Power down noise canceling DAC in FRAC mode 1'b0: no power down 1'b1: power down
23:0	RW	0xde69ad	fracdiv Fractional portion of feedback divide value

CRU_VPLL1_CON3

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	WO	0x00	ssmod_spread Set modulation depth(spread percentage) of SSMOD 5'h0: 0% 5'h1: 0.1% 5'h2: 0.2% ... 5'h1e: 3% 5'h1f: 3.1%

Bit	Attr	Reset Value	Description
7:4	WO	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	WO	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: down spread 1'b1: center spread
2	WO	0x1	ssmod_reset SSMOD reset 1'b0: no reset 1'b1: reset
1	WO	0x1	ssmod_disable_sscg Disable sperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	WO	0x1	ssmod_bp SSMOD bypass 1'b0: no bypass 1'b1: bypass

CRU_VPLL1_CON4

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	WO	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid
7:1	RO	0x0	reserved
0	WO	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU_MODE

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13	RW	0x0	vpll1_clk_sel VPLL1 clock source select 1'b0: PLL clock output with level shift 1'b1: PLL clock output without level shift
12	RW	0x0	vpll0_clk_sel VPLL0 clock source select 1'b0: PLL clock output with level shift 1'b1: PLL clock output without level shift
11	RW	0x0	dpll_clk_sel DPLL clock source select 1'b0: PLL clock output with level shift 1'b1: PLL clock output without level shift
10	RW	0x0	apll_clk_sel APLL clock source select 1'b0: PLL clock output with level shift 1'b1: PLL clock output without level shift
9:8	RW	0x0	usbphy480m_work_mode usbphy480m work mode select 2'b00: clock from xin_osc0_func_div 2'b01: clock from PLL 2'b10: clock from clk_rtc_32k 2'b11: reserved
7:6	RW	0x0	vpll1_work_mode VPLL1 work mode select 2'b00: clock from xin_osc0_func_div 2'b01: clock from PLL 2'b10: clock from clk_rtc_32k 2'b11: reserved
5:4	RW	0x0	vpll0_work_mode VPLL0 work mode select 2'b00: clock from xin_osc0_func_div 2'b01: clock from PLL 2'b10: clock from clk_rtc_32k 2'b11: reserved
3:2	RW	0x0	dpll_work_mode DPLL work mode select 2'b00: clock from xin_osc0_func_div 2'b01: clock from PLL 2'b10: clock from clk_rtc_32k 2'b11: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	apll_work_mode APLL work mode select 2'b00: clock from xin_osc0_func_div 2'b01: clock from PLL 2'b10: clock from clk_rtc_32k 2'b11: reserved

CRU_MISC

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RW	0x0	core_high_freq_rst_en 1'b0: disable high frequency reset function 1'b1: enable high frequency reset function Each bit for each core, eg. bit0 for core0
11:5	RO	0x0	reserved
4	RW	0x0	corepo_wrst_wfien 1'b0: core0/1/2/3 power on reset is asserted immediately after warm reset request, it has no relation to STANDBYWFI status 1'b1: core0/1/2/3 power on reset is not asserted immediately after warm reset request, it will wait until after entering STANDBYWFI
3	RW	0x0	corepo_srst_wfien 1'b0: core0/1/2/3 power on reset is asserted immediately after software reset request, it has no relation to STANDBYWFI status 1'b1: core0/1/2/3 power on reset is not asserted immediately after software reset request, it will wait until after entering STANDBYWFI
2	RW	0x0	core_wrst_wfien 1'b0: core0/1/2/3 reset is asserted immediately after warm reset request, it has no relation to STANDBYWFI status 1'b1: core0/1/2/3 reset is not asserted immediately after warm reset request, it will wait until after entering STANDBYWFI
1	RW	0x0	core_srst_wfien 1'b0: core0/1/2/3 reset is asserted immediately after software reset request, it has no relation to STANDBYWFI status 1'b1: core0/1/2/3 reset is not asserted immediately after software reset request, it will wait until after entering STANDBYWFI
0	RW	0x0	warmrst_en 1'b0: disable cpu warm reset 1'b1: enable cpu warm reset

CRU_GLB_CNT_TH

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	global_reset_counter_threshold Global soft reset, wdt reset or tsadc_shut reset asserted time counter threshold. Measured in OSC clock cycles

CRU_GLB_RST_ST

Address: Operational Base + offset (0x00b4)

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

CRU_GLB_SRST_FST

Address: Operational Base + offset (0x00b8)

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

CRU_GLB_SRST_SND

Address: Operational Base + offset (0x00bc)

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

CRU_GLB_RST_CON

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	chiprstn_pd_sel 1'b0: chiprstn to power domain can be triggered by npor and global soft reset 1'b1: chiprstn to power domain can be triggered by npor only
7	RW	0x0	wdt_reset_ext_en 1'b0: disable wdt reset extend 1'b1: enable wdt reset extend, reset extend time depend on bit15~0 of GLB_CNT_TH
6	RW	0x0	tsadc_shut_reset_ext_en 1'b0: disable tsadc_shut reset extend 1'b1: enable tsadc_shut reset extend, reset extend time depend on bit15~0 of GLB_CNT_TH
5	RO	0x0	reserved
4	RW	0x0	pmu_srst_wdt_en 1'b0: enable wdt reset as pmu reset source 1'b1: disable wdt reset as pmu reset source
3	RW	0x0	pmu_srst_glb_rst_en 1'b0: enable first or second global reset as pmu reset source 1'b1: disable first or second global reset as pmu reset source
2	RW	0x0	pmu_srst_ctrl 1'b0: first global reset trigger pmu reset 1'b1: second global reset trigger pmu reset
1	RW	0x0	wdt_glb_srst_ctrl 1'b0: wdt trigger second global reset 1'b1: wdt trigger first global reset
0	RW	0x0	tsadc_glb_srst_ctrl 1'b0: tsadc trigger second global reset 1'b1: tsadc trigger first global reset

CRU_GLB_PLL_LOCK

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x3a98	pll_lockperiod Measured in OSC clock cycles

CRU_HWFFC_CON0

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	hwffc_mode 1'b0: hwffc mode disable. Use APLL_CON0 and APLL_CON1 as APLL configurations 1'b1: hwffc mode enable. Use APLL_CON0_S and APLL_CON1_S as APLL configurations
8	RW	0x0	hwffc_req 1'b0: indicate hwffc is done 1'b1: set hwffc request or hwffc is on going, user should set hwffc_mode=1 first
7:6	RW	0x0	backuppll_sel 2'b00: reserved 2'b01: select VPLL0 as backup pll for HWFFC 2'b10: select VPLL1 as backup pll for HWFFC 2'b11: reserved
5:4	RO	0x0	reserved
3:0	RW	0x0	clk_core_div_con clk_core=pll_clk_src/(div_con+1)

CRU_HWFFC_TH

Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RW	0x03	switch_th clock switch threshold cycle. Unit is pclk
7:0	RW	0x0f	pll_pd_extend_th pll pd extend time threshold cycle. Unit is pclk

CRU_HWFFC_INTSTS

Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	hwffc_done_inten 1'b0: disable hwffc_done interrupt 1'b1: enable hwffc_done interrupt
7	RO	0x0	reserved
6:4	RW	0x0	hwffc_state 3'h0: hwffc_idle state 3'h1: hwffc_2_backup_pll_mux state 3'h2: hwffc_2_backup_pll_div state 3'h3: hwffc_cfg_apll state 3'h4: hwffc_back_apll_mux state 3'h5: hwffc_back_apll_div_state 3'h6: hwffc_done state others: reserved
3:1	RO	0x0	reserved
0	W1C	0x0	hwffc_done 1'b0: hwffc done interrupt not generated 1'b1: hwffc done interrupt generated

CRU_APLL_CON0_S

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	bypass REF is bypassed to FOUTPOSTDIV 1'b0: no bypass 1'b1: bypass
14:12	RO	0x3	postdiv1 PLL post divide 1 setting, 1-7 is valid postdiv1 should be greater than or equal to postdiv2
11:0	RO	0x064	fbdiv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: no plus one operation

CRU_APLL_CON1_S

Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	pllpdsel 1'b0: PLL is power down when any one of refdiv/fbdiv/fracdiv is changed or pllpd0 is asserted 1'b1: PLL can be power down only by pllpd1

Bit	Attr	Reset Value	Description
14	RO	0x0	pllpd1 PLL power down 1 setting, it's valid when pllpdsel=1'b1 1'b0: no power down 1'b1: power down
13	RO	0x0	pllpd0 PLL power down 0 setting, it's valid when pllpdsel=1'b0 1'b0: no power down 1'b1: power down
12	RO	0x1	dsmpd Power down delta-sigma modulator 1'b0: no power down, PLL is fractional mode 1'b1: power down, PLL is integer mode
11	RO	0x0	reserved
10	RO	0x0	pll_lock 1'b0: unlock 1'b1: lock
9	RO	0x0	reserved
8:6	RO	0x1	postdiv2 PLL post divide 2 setting, 1-7 is valid
5:0	RO	0x01	refdiv Reference divide value, 1-63 is valid

CRU_CLKSEL_CON0_S
Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:12	RO	0x1	aclk_core_div_con aclk_core=clk_core/(div_con+1)
11:8	RO	0x3	core_dbg_div_con pclk_dbg=clk_core/(div_con+1)
7:6	RO	0x0	core_clk_pll_sel 2'b00: APLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5:4	RO	0x0	reserved
3:0	RO	0x0	clk_core_div_con clk_core=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON0
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14:12	RW	0x1	aclk_core_div_con aclk_core=clk_core/(div_con+1)
11:8	RW	0x3	core_dbg_div_con pclk_dbg=clk_core/(div_con+1)
7:6	RW	0x0	core_clk_pll_sel 2'b00: APLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5:4	RO	0x0	reserved
3:0	RW	0x0	clk_core_div_con clk_core=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	ddr_stdby_clk_sel 1'b0: select ddrphy_dfi_clk1x_out as clk_ddr_stdby clock 1'b1: select ddr_phy_dfi_clk4x_div4 as clk_ddr_stdby clock
7:6	RW	0x2	ddrphy_dfi_clk4x_pll_clk_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5:3	RO	0x0	reserved
2:0	RW	0x0	ddrphy_dfi_clk4x_div_con ddrphy_dfi_clk4x=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON2

Address: Operational Base + offset (0x0108)

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

Bit	Attr	Reset Value	Description
10	RW	0x0	clk_32k_pll_sel 1'b0: VPLL0 1'b1: VPLL1
9:8	RW	0x3	clk_32k_sel 2'b00: clk_32k_from_io 2'b01: clk_32k_from_pmu_pvtm 2'b10: clk_32k_frac_div 2'b11: clk_32k_div
7:5	RO	0x0	reserved
4:0	RW	0x00	func_24m_div_con xin_osc0_func_div=xin_osc0_func/(div_con+1)

CRU_CLKSEL_CON3

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_32k_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is xin_osc0

CRU_CLKSEL_CON4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:0	RW	0x7530	clk_32k_div_con clk_32k_div=clk_32k_src/(div_con+1)

CRU_CLKSEL_CON5

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	aclk_hclk_pclk_bus_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5	RO	0x0	reserved
4:0	RW	0x05	aclk_bus_div_con aclk_bus=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON6

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x0b	pclk_bus_div_con pclk_bus=pll_clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x0b	hclk_bus_div_con hclk_bus=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON7

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_crypto_apk_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
13	RO	0x0	reserved
12:8	RW	0x03	clk_crypto_apk_div_con clk_crypto_apk=pll_clk_src/(div_con+1)
7:6	RW	0x0	clk_crypto_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5	RO	0x0	reserved
4:0	RW	0x05	clk_crypto_div_con clk_crypto=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON8

Address: Operational Base + offset (0x0120)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	dclk_vop_sel 2'b00: select dclk_vop 2'b01: select dclk_vop_frac_out 2'b10: select xin_osc0 2'b11: reserved
13:12	RO	0x0	reserved
11:10	RW	0x0	dclk_vop_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
9:8	RO	0x0	reserved
7:0	RW	0x07	dclk_vop_div_con dclk_vop=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON9

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dclk_vop_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is dclk_vops

CRU_CLKSEL_CON10

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	clk_uart0_pll_sel 3'b000: DPLL 3'b001: VPLL0 3'b010: VPLL1 3'b011: usbphy480m 3'b100: xin_osc0 others: reserved
12:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart0_div_con clk_uart0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON11

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart0_sel 2'b00: select clk_uart0 2'b01: select clk_uart0_np5 2'b10: select clk_uart0_frac_out 2'b11: reserved
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart0_divnp5_div_con clk_uart0_np5=2*clk_uart0/(2*div_con+3)

CRU_CLKSEL_CON12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart0_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart0

CRU_CLKSEL_CON13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	clk_uart1_pll_sel 3'b000: DPLL 3'b001: VPLL0 3'b010: VPLL1 3'b011: usbphy480m 3'b100: xin_osc0 others: reserved
12:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart1_div_con clk_uart1=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON14

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart1_sel 2'b00: select clk_uart1 2'b01: select clk_uart1_np5 2'b10: select clk_uart1_frac_out 2'b11: reserved
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart1_divnp5_div_con clk_uart1_np5=2*clk_uart1/(2*div_con+3)

CRU_CLKSEL_CON15

Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart1_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart1

CRU_CLKSEL_CON16

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	clk_uart2_pll_sel 3'b000: DPLL 3'b001: VPLL0 3'b010: VPLL1 3'b011: usbphy480m 3'b100: xin_osc0 others: reserved
12:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart2_div_con clk_uart2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON17

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart2_sel 2'b00: select clk_uart2 2'b01: select clk_uart2_np5 2'b10: select clk_uart2_frac_out 2'b11: reserved
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart2_divnp5_div_con clk_uart2_np5=2*clk_uart2/(2*div_con+3)

CRU_CLKSEL_CON18
Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart2_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart2

CRU_CLKSEL_CON19
Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	clk_uart3_pll_sel 3'b000: DPLL 3'b001: VPLL0 3'b010: VPLL1 3'b011: usbphy480m 3'b100: xin_osc0 others: reserved
12:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart3_div_con clk_uart3=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON20
Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart3_sel 2'b00: select clk_uart3 2'b01: select clk_uart3_np5 2'b10: select clk_uart3_frac_out 2'b11: reserved
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart3_divnp5_div_con clk_uart3_np5=2*clk_uart3/(2*div_con+3)

CRU_CLKSEL_CON21

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart3_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart3

CRU_CLKSEL_CON22

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RW	0x0	clk_uart4_pll_sel 3'b000: DPLL 3'b001: VPLL0 3'b010: VPLL1 3'b011: usbphy480m 3'b100: xin_osc0 others: reserved
12:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart4_div_con clk_uart4=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON23

Address: Operational Base + offset (0x015c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_uart4_sel 2'b00: select clk_uart4 2'b01: select clk_uart4_np5 2'b10: select clk_uart4_frac_out 2'b11: reserved
13:5	RO	0x0	reserved
4:0	RW	0x0b	clk_uart4_divnp5_div_con clk_uart4_np5=2*clk_uart4/(2*div_con+3)

CRU_CLKSEL_CON24

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart4_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_uart4

CRU_CLKSEL_CON25

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_i2c0_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x05	clk_i2c0_div_con clk_i2c0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON26

Address: Operational Base + offset (0x0168)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	clk_i2c1_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x05	clk_i2c1_div_con clk_i2c1=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON27
Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_i2c2_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x05	clk_i2c2_div_con clk_i2c2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON28
Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_i2c3_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x05	clk_i2c3_div_con clk_i2c3=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON29
Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_pwm_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x0b	clk_pwm_div_con clk_pwm=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON30

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_spi0_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x0b	clk_spi0_div_con clk_spi0=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON31

Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_spi1_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x0b	clk_spi1_div_con clk_spi1=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON32

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_spi2_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: xin_osc0 2'b11: reserved
13:7	RO	0x0	reserved
6:0	RW	0x0b	clk_spi2_div_con clk_spi2=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON33

Address: Operational Base + offset (0x0184)

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

CRU_CLKSEL_CON34

Address: Operational Base + offset (0x0188)

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

CRU_CLKSEL_CON35

Address: Operational Base + offset (0x018c)

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

CRU_CLKSEL_CON36

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7:6	RW	0x0	aclk_hclk_pclk_peri_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5	RO	0x0	reserved
4:0	RW	0x05	aclk_peri_div_con aclk_peri=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON37

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x0b	pclk_peri_div_con pclk_peri=pll_clk_src/(div_con+1)
7:5	RO	0x0	reserved
4:0	RW	0x0b	hclk_peri_div_con hclk_peri=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON38

Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_nandc_sel50 1'b0: The duty ratio of clk_nandc is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_nandc is always 50%
14:8	RO	0x0	reserved
7:6	RW	0x0	clk_nandc_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved

Bit	Attr	Reset Value	Description
5	RO	0x0	reserved
4:0	RW	0x07	clk_nandc_div_con clk_nandc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON39

Address: Operational Base + offset (0x019c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sdmmc_sel50 1'b0: The duty ratio of clk_sdmmc is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_sdmmc is always 50%
14:10	RO	0x0	reserved
9:8	RW	0x0	clk_sdmmc_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: xin_osc0
7:0	RW	0x03	clk_sdmmc_div_con clk_sdmmc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON40

Address: Operational Base + offset (0x01a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_sdio_sel50 1'b0: The duty ratio of clk_sdio is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_sdio is always 50%
14:10	RO	0x0	reserved
9:8	RW	0x0	clk_sdio_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: xin_osc0
7:0	RW	0x03	clk_sdio_div_con clk_sdio=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON41

Address: Operational Base + offset (0x01a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_emmc_sel50 1'b0: The duty ratio of clk_emmc is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_emmc is always 50%
14:10	RO	0x0	reserved
9:8	RW	0x0	clk_emmc_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: xin_osc0
7:0	RW	0x03	clk_emmc_div_con clk_emmc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON42

Address: Operational Base + offset (0x01a8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x1	clk_sfc_pll_sel 2'b00:DPLL 2'b01:VPLL0 2'b10:VPLL1
13:7	RO	0x0	reserved
6:0	RW	0x04	clk_sfc_div_con clk_sfc=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON43

Address: Operational Base + offset (0x01ac)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x1	rmii_clk_sel 1'b0: 10M 1'b1: 100M
14	RW	0x0	rmii_extclksrc_sel 1'b0: select clk_mac as the clock of mac 1'b1: select external phy clock as the clock of mac
13:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	clk_mac_pll_sel 2'b00: DPLL 2'b01: VPLL0 2'b10: VPLL1 2'b11: reserved
5	RO	0x0	reserved
4:0	RW	0x17	clk_mac_div_con clk_mac=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON44
Address: Operational Base + offset (0x01b0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x1	clk_wifi_sel 1'b0: select xin_osc0 as clk_wifi_out 1'b1: select clk_wifi_div as clk_wifi_out
6	RW	0x0	clk_wifi_pll_sel 1'b0: DPLL 1'b1: VPLL0
5:0	RW	0x1d	clk_wifi_div_con clk_wifi_div=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON45
Address: Operational Base + offset (0x01b4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x09	pclk_audio_div_con pclk_audio=pll_clk_src/(div_con+1)
7:6	RW	0x0	hclk_pclk_audio_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
5	RO	0x0	reserved
4:0	RW	0x09	hclk_audio_div_con hclk_audio=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON46

Address: Operational Base + offset (0x01b8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_pdm_sel 1'b0: select clk_pdm 1'b1: select clk_pdm_frac_out
14:10	RO	0x0	reserved
9:8	RW	0x0	clk_pdm_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x0f	clk_pdm_div_con clk_pdm=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON47

Address: Operational Base + offset (0x01bc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_pdm_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_pdm

CRU_CLKSEL_CON48

Address: Operational Base + offset (0x01c0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RW	0x0	clk_spdiftx_sel 2'b00: select clk_spdiftx 2'b01: select clk_spdiftx_frac_out 2'b10: select mclk_from_io
13	RO	0x0	reserved
12	RW	0x0	clk_spdiftx_sel50 1'b0: The duty ratio of clk_spdiftx is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_spdiftx is always 50%
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	clk_spdiftx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x09	clk_spdiftx_div_con clk_spdiftx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON49

Address: Operational Base + offset (0x01c4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_spdiftx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_spdiftx

CRU_CLKSEL_CON50

Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_spdifrx_sel 1'b0: select clk_spdifrx 1'b1: select clk_spdifrx_frac_out
14	RW	0x0	clk_spdifrx_sel50 1'b0: The duty ratio of clk_spdifrx is 50% only when div_con+1 is even 1'b1: The duty ratio of clk_spdifrx is always 50%
13:10	RO	0x0	reserved
9:8	RW	0x0	clk_spdifrx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x03	clk_spdifrx_div_con clk_spdifrx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON51

Address: Operational Base + offset (0x01cc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_spdifrx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_spdifrx

CRU_CLKSEL_CON52

Address: Operational Base + offset (0x01d0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s0_8ch_tx_out_mclk_sel 1'b0: select selected clock by clk_i2s0_8ch_tx_rx_clk_sel 1'b1: select xin_osc0_half
14:13	RO	0x0	reserved
12	RW	0x0	clk_i2s0_8ch_tx_rx_clk_sel 1'b0: select clk_i2s0_8ch_tx_clk 1'b1: select clk_i2s0_8ch_rx_clk
11:10	RW	0x0	clk_i2s0_8ch_tx_sel 2'b00: select clk_i2s0_8ch_tx 2'b01: select clk_i2s0_8ch_tx_frac_out 2'b10: select mclk_i2s0_8ch_tx_in 2'b11: reserved
9:8	RW	0x1	clk_i2s0_8ch_tx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x11	clk_i2s0_8ch_tx_div_con clk_i2s0_8ch_tx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON53

Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s0_8ch_tx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s0_8ch_tx

CRU_CLKSEL_CON54

Address: Operational Base + offset (0x01d8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_i2s0_8ch_rx_tx_clk_sel 1'b0: select clk_i2s0_8ch_rx_clk 1'b1: select clk_i2s0_8ch_tx_clk
11:10	RW	0x0	clk_i2s0_8ch_rx_sel 2'b00: select clk_i2s0_8ch_rx 2'b01: select clk_i2s0_8ch_rx_frac_out 2'b10: select mclk_i2s0_8ch_rx_in 2'b11: reserved
9:8	RW	0x0	clk_i2s0_8ch_rx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s0_8ch_rx_div_con clk_i2s0_8ch_rx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON55

Address: Operational Base + offset (0x01dc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s0_8ch_rx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s0_8ch_rx

CRU_CLKSEL_CON56

Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s1_8ch_tx_out_mclk_sel 1'b0: select selected clock by clk_i2s1_8ch_tx_rx_clk_sel 1'b1: select xin_osc0_half
14:13	RO	0x0	reserved
12	RW	0x0	clk_i2s1_8ch_tx_rx_clk_sel 1'b0: select clk_i2s1_8ch_tx_clk 1'b1: select clk_i2s1_8ch_rx_clk

Bit	Attr	Reset Value	Description
11:10	RW	0x0	clk_i2s1_8ch_tx_sel 2'b00: select clk_i2s1_8ch_tx 2'b01: select clk_i2s1_8ch_tx_frac_out 2'b10: select mclk_i2s1_8ch_tx_in 2'b11: reserved
9:8	RW	0x1	clk_i2s1_8ch_tx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x11	clk_i2s1_8ch_tx_div_con clk_i2s1_8ch_tx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON57

Address: Operational Base + offset (0x01e4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s1_8ch_tx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s1_8ch_tx

CRU_CLKSEL_CON58

Address: Operational Base + offset (0x01e8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_i2s1_8ch_rx_tx_clk_sel 1'b1: select clk_i2s1_8ch_rx_clk 1'b1: select clk_i2s1_8ch_tx_clk
11:10	RW	0x0	clk_i2s1_8ch_rx_sel 2'b00: select clk_i2s1_8ch_rx 2'b01: select clk_i2s1_8ch_rx_frac_out 2'b10: select mclk_i2s1_8ch_rx_in 2'b11: reserved
9:8	RW	0x0	clk_i2s1_8ch_rx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s1_8ch_rx_div_con clk_i2s1_8ch_rx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON59

Address: Operational Base + offset (0x01ec)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s1_8ch_rx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s1_8ch_rx

CRU_CLKSEL_CON60

Address: Operational Base + offset (0x01f0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s2_8ch_tx_out_mclk_sel 1'b0: select selected clock by clk_i2s2_8ch_tx_rx_clk_sel 1'b1: select xin_osc0_half
14:13	RO	0x0	reserved
12	RW	0x0	clk_i2s2_8ch_tx_rx_clk_sel 1'b0: select clk_i2s2_8ch_tx_clk 1'b1: select clk_i2s2_8ch_rx_clk
11:10	RW	0x0	clk_i2s2_8ch_tx_sel 2'b00: select clk_i2s2_8ch_tx 2'b01: select clk_i2s2_8ch_tx_frac_out 2'b10: select mclk_i2s2_8ch_tx_in 2'b11: reserved
9:8	RW	0x1	clk_i2s2_8ch_tx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x11	clk_i2s2_8ch_tx_div_con clk_i2s2_8ch_tx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON61

Address: Operational Base + offset (0x01f4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s2_8ch_tx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s2_8ch_tx

CRU_CLKSEL_CON62

Address: Operational Base + offset (0x01f8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_i2s2_8ch_rx_tx_clk_sel 1'b0: select clk_i2s2_8ch_rx_clk 1'b1: select clk_i2s2_8ch_tx_clk
11:10	RW	0x0	clk_i2s2_8ch_rx_sel 2'b00: select clk_i2s2_8ch_rx 2'b01: select clk_i2s2_8ch_rx_frac_out 2'b10: select mclk_i2s2_8ch_rx_in 2'b11: reserved
9:8	RW	0x0	clk_i2s2_8ch_rx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s2_8ch_rx_div_con clk_i2s2_8ch_rx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON63

Address: Operational Base + offset (0x01fc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s2_8ch_rx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s2_8ch_rx

CRU_CLKSEL_CON64

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s3_8ch_tx_out_mclk_sel 1'b0: select selected clock by clk_i2s3_8ch_tx_rx_clk_sel 1'b1: select xin_osc0_half
14:13	RO	0x0	reserved
12	RW	0x0	clk_i2s3_8ch_tx_rx_clk_sel 1'b0: select clk_i2s3_8ch_tx_clk 1'b1: select clk_i2s3_8ch_rx_clk

Bit	Attr	Reset Value	Description
11:10	RW	0x0	clk_i2s3_8ch_tx_sel 2'b00: select clk_i2s3_8ch_tx 2'b01: select clk_i2s3_8ch_tx_frac_out 2'b10: select mclk_i2s3_8ch_tx_in 2'b11: reserved
9:8	RW	0x1	clk_i2s3_8ch_tx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x11	clk_i2s3_8ch_tx_div_con clk_i2s3_8ch_tx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON65
Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s3_8ch_tx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s3_8ch_tx

CRU_CLKSEL_CON66
Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	clk_i2s3_8ch_rx_tx_clk_sel 1'b0: select clk_i2s3_8ch_rx_clk 1'b1: select clk_i2s3_8ch_tx_clk
11:10	RW	0x0	clk_i2s3_8ch_rx_sel 2'b00: select clk_i2s3_8ch_rx 2'b01: select clk_i2s3_8ch_rx_frac_out 2'b10: select mclk_i2s3_8ch_rx_in 2'b11: reserved
9:8	RW	0x0	clk_i2s3_8ch_rx_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s3_8ch_rx_div_con clk_i2s3_8ch_rx=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON67

Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s3_8ch_rx_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s3_8ch_rx

CRU_CLKSEL_CON68

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s0_2ch_out_mclk_sel 1'b0: select selected clock by clk_i2s0_2ch_sel 1'b1: select xin_osc0_half
14:12	RO	0x0	reserved
11:10	RW	0x0	clk_i2s0_2ch_sel 2'b00: select clk_i2s0_2ch 2'b01: select clk_i2s0_2ch_frac_out 2'b10: select mclk_i2s0_2ch_in 2'b11: reserved
9:8	RW	0x0	clk_i2s0_2ch_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s0_2ch_div_con clk_i2s0_2ch=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON69

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s0_2ch_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s0_2ch

CRU_CLKSEL_CON70

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s1_2ch_out_mclk_sel 1'b0: select selected clock by clk_i2s1_2ch_sel 1'b1: select xin_osc0_half
14:12	RO	0x0	reserved
11:10	RW	0x0	clk_i2s1_2ch_sel 2'b00: select clk_i2s1_2ch 2'b01: select clk_i2s1_2ch_frac_out 2'b10: select mclk_i2s1_2ch_in 2'b11: reserved
9:8	RW	0x0	clk_i2s1_2ch_pll_sel 2'b00: VPLL0 2'b01: VPLL1 2'b10: xin_osc0 2'b11: reserved
7	RO	0x0	reserved
6:0	RW	0x13	clk_i2s1_2ch_div_con clk_i2s1_2ch=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON71

Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s1_2ch_frac_div_con High 16-bit for numerator, Low 16-bit for denominator, clock source is clk_i2s1_2ch

CRU_CLKSEL_CON72

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:8	RO	0x0	reserved
7	RW	0x0	usbphy_ref_sel 1'b0: select xin_osc0 as usbphy reference clock 1'b1: select clk_ref24m as usbphy reference clock
6	RW	0x0	clk_ref24m_pll_sel 1'b0: DPLL 1'b1: VPLL0
5:0	RW	0x31	clk_ref24m_div_con clk_ref24m=pll_clk_src/(div_con+1)

CRU_CLKSEL_CON73

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12:8	RW	0x00	testclk_sel 5'h00: xin_osc0 5'h01: clk_core 5'h02: clk_ddrphy4x 5'h03: clk_ddrphy1x 5'h04: aclk_peri 5'h05: hclk_peri 5'h06: clk_nandc 5'h07: clk_sdmmc 5'h08: clk_sdio 5'h09: clk_emmc 5'h0a: clk_sfc 5'h0b: clk_mac_ref 5'h0c: aclk_bus 5'h0d: hclk_bus 5'h0e: clk_crypto 5'h0f: clk_crypto_apk 5'h10: dclk_vop 5'h11: clk_uart0 5'h12: clk_i2c0 5'h13: clk_spi0 5'h14: clk_tsadc 5'h15: clk_saradc 5'h16: clk_otp 5'h17: hclk_audio 5'h18: clk_pdm 5'h19: clk_spdifrx 5'h1a: clk_i2s0_8ch_tx 5'h1b: clk_i2s0_8ch_rx 5'h1c: clk_i2s0_2ch 5'h1d: clk_rtc32k 5'h1e: clk_usbphy_ref24m 5'h1f: otp_isp_osc_out
7:5	RO	0x0	reserved
4:0	RW	0x1f	test_div_con clk_test_out=test_clk_src/(div_con+1)

CRU_CLKGATE_CON0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:14	RO	0x0	reserved
13	RW	0x0	ddr_phy_dfi_clk4x_div4_clk_en When HIGH, disable clock
12	RW	0x0	ddr_monitor_timer_clk_en When HIGH, disable clock
11	RW	0x0	ddr_phy_dfi_clk4x_en When HIGH, disable clock
10	RW	0x0	ddr_phy_pll_clk_en When HIGH, disable clock
9	RW	0x0	pclk_core_grf_clk_en When HIGH, disable clock
8	RW	0x0	aclk_core_perf_clk_en When HIGH, disable clock
7	RW	0x0	pclk_core_dbg_daplite_clk_en When HIGH, disable clock
6	RW	0x0	pclk_core_dbg_niu_clk_en When HIGH, disable clock
5	RW	0x0	aclk_core_niu_clk_en When HIGH, disable clock
4	RW	0x0	clk_core_pvtm_clk_en When HIGH, disable clock
3	RW	0x0	clk_jtag_core_clk_en When HIGH, disable clock
2	RW	0x0	pclk_core_dbg_src_clk_en When HIGH, disable clock
1	RW	0x0	aclk_core_src_clk_en When HIGH, disable clock
0	RW	0x0	core_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_uart1_divfrac_clk_en When HIGH, disable clock
14	RW	0x0	clk_uart1_divnp5_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
13	RW	0x0	clk_uart1_pll_clk_en When HIGH, disable clock
12	RW	0x0	clk_uart0_clk_en When HIGH, disable clock
11	RW	0x0	clk_uart0_divfrac_clk_en When HIGH, disable clock
10	RW	0x0	clk_uart0_divnp5_clk_en When HIGH, disable clock
9	RW	0x0	clk_uart0_pll_clk_en When HIGH, disable clock
8	RW	0x0	dclkvop_clk_en When HIGH, disable clock
7	RW	0x0	dclkvop_fracdiv_clk_en When HIGH, disable clock
6	RW	0x0	dclkvop_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_crypto_apk_pll_clk_en When HIGH, disable clock
4	RW	0x0	clk_crypto_pll_clk_en When HIGH, disable clock
3	RW	0x0	pclk_bus_clk_en When HIGH, disable clock
2	RW	0x0	hclk_bus_clk_en When HIGH, disable clock
1	RW	0x0	aclk_bus_clk_en When HIGH, disable clock
0	RW	0x0	logic_bus_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON2

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2c2_pll_clk_en When HIGH, disable clock
14	RW	0x0	clk_i2c1_pll_clk_en When HIGH, disable clock
13	RW	0x0	clk_i2c0_pll_clk_en When HIGH, disable clock
12	RW	0x0	clk_uart4_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
11	RW	0x0	clk_uart4_divfrac_clk_en When HIGH, disable clock
10	RW	0x0	clk_uart4_divnp5_clk_en When HIGH, disable clock
9	RW	0x0	clk_uart4_pll_clk_en When HIGH, disable clock
8	RW	0x0	clk_uart3_clk_en When HIGH, disable clock
7	RW	0x0	clk_uart3_divfrac_clk_en When HIGH, disable clock
6	RW	0x0	clk_uart3_divnp5_clk_en When HIGH, disable clock
5	RW	0x0	clk_uart3_pll_clk_en When HIGH, disable clock
4	RW	0x0	clk_uart2_clk_en When HIGH, disable clock
3	RW	0x0	clk_uart2_divfrac_clk_en When HIGH, disable clock
2	RW	0x0	clk_uart2_divnp5_clk_en When HIGH, disable clock
1	RW	0x0	clk_uart2_pll_clk_en When HIGH, disable clock
0	RW	0x0	clk_uart1_clk_en When HIGH, disable clock

CRU_CLKGATE_CON3

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_timer5_en When HIGH, disable clock
14	RW	0x0	clk_timer4_en When HIGH, disable clock
13	RW	0x0	clk_timer3_en When HIGH, disable clock
12	RW	0x0	clk_timer2_en When HIGH, disable clock
11	RW	0x0	clk_timer1_en When HIGH, disable clock
10	RW	0x0	clk_timer0_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_cpu_boost_clk_en When HIGH, disable clock
8	RW	0x0	clk_otp_usr_clk_en When HIGH, disable clock
7	RW	0x0	clk_otp_pll_clk_en When HIGH, disable clock
6	RW	0x0	clk_saradc_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_tsadc_pll_clk_en When HIGH, disable clock
4	RW	0x0	clk_spi2_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_spi1_pll_clk_en When HIGH, disable clock
2	RW	0x0	clk_spi0_pll_clk_en When HIGH, disable clock
1	RW	0x0	clk_pwm_pll_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2c3_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON4

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	clk_ddr_stdby_en When HIGH, disable clock
13	RW	0x0	clk_ddr_msch_peri_en When HIGH, disable clock, and cannot access the peripheral bus
12	RW	0x0	clk_ddr_msch_en When HIGH, disable clock
11	RW	0x0	clk_ddr_upctrl_en When HIGH, disable clock
10	RW	0x0	clk_ddr_mon_en When HIGH, disable clock
9	RW	0x0	testclk_en When HIGH, disable clock
8	RW	0x0	usbphy_ref24m_clk_en When HIGH, disable clock
7	RW	0x0	clk_ref24m_pll_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_pmu_clk_en When HIGH, disable clock
5	RW	0x0	pclk_pmu_clk_en When HIGH, disable clock
4	RW	0x0	clk_pmu_pvtm_clk_en When HIGH, disable clock
3	RW	0x0	clk_div32k_frac_out_clk_en When HIGH, disable clock
2	RW	0x0	clk_div32k_out_clk_en When HIGH, disable clock
1	RW	0x0	clk_wifi_clk_en When HIGH, disable clock
0	RW	0x0	clk_wifi_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON5

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_i2c0_en When HIGH, disable clock
14	RW	0x0	pclk_uart4_en When HIGH, disable clock
13	RW	0x0	pclk_uart3_en When HIGH, disable clock
12	RW	0x0	pclk_uart2_en When HIGH, disable clock
11	RW	0x0	pclk_uart1_en When HIGH, disable clock
10	RW	0x0	pclk_uart0_en When HIGH, disable clock
9	RW	0x0	pclk_bus_niu_en When HIGH, disable clock
8	RW	0x0	hclk_vop_en When HIGH, disable clock
7	RW	0x0	hclk_crypto_en When HIGH, disable clock
6	RW	0x0	hclk_rom_en When HIGH, disable clock
5	RW	0x0	hclk_bus_niu_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
4	RW	0x0	aclk_gic_en When HIGH, disable clock
3	RW	0x0	aclk_vop_en When HIGH, disable clock
2	RW	0x0	aclk_crypto_en When HIGH, disable clock
1	RW	0x0	aclk_intmem_en When HIGH, disable clock
0	RW	0x0	aclk_bus_niu_en When HIGH, disable clock

CRU_CLKGATE_CON6

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_gpio3_en When HIGH, disable clock
14	RW	0x0	pclk_gpio2_en When HIGH, disable clock
13	RW	0x0	pclk_gpio1_en When HIGH, disable clock
12	RW	0x0	pclk_gpio0_en When HIGH, disable clock
11	RO	0x0	reserved
10	RW	0x0	pclk_otp_ns_en When HIGH, disable clock
9	RW	0x0	pclk_timer_en When HIGH, disable clock
8	RW	0x0	pclk_tsadc_en When HIGH, disable clock
7	RW	0x0	pclk_saradc_en When HIGH, disable clock
6	RW	0x0	pclk_spi2_en When HIGH, disable clock
5	RW	0x0	pclk_spi1_en When HIGH, disable clock
4	RW	0x0	pclk_spi0_en When HIGH, disable clock
3	RW	0x0	pclk_pwm_en When HIGH, disable clock
2	RW	0x0	pclk_i2c3_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
1	RW	0x0	pclk_i2c2_en When HIGH, disable clock
0	RW	0x0	pclk_i2c1_en When HIGH, disable clock

CRU_CLKGATE_CON7

Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:12	RO	0x0	reserved
11	RW	0x0	pclk_cpu_boost_en When HIGH, disable clock
10	RW	0x0	pclk_otp_phy_en When HIGH, disable clock
9	RW	0x0	pclk_cru_en When HIGH, disable clock
8	RW	0x0	pclk_usbgrf_en When HIGH, disable clock
7	RW	0x0	pclk_ddr_stdby_en When HIGH, disable clock
6	RW	0x0	pclk_ddr_phy_en When HIGH, disable clock
5	RW	0x0	pclk_ddr_mon_en When HIGH, disable clock
4	RW	0x0	pclk_ddr_upctrl_en When HIGH, disable clock
3	RW	0x0	pclk_usbsdmmc_det_en When HIGH, disable clock
2	RW	0x0	pclk_grf_en When HIGH, disable clock
1	RW	0x0	pclk_sgrf_en When HIGH, disable clock
0	RW	0x0	pclk_gpio4_en When HIGH, disable clock

CRU_CLKGATE_CON8

Address: Operational Base + offset (0x0320)

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

Bit	Attr	Reset Value	Description
14	RW	0x0	clk_mac_pll_clk_en When HIGH, disable clock
13	RW	0x0	clk_otg_adp_clk_en When HIGH, disable clock
12	RW	0x0	clk_sfc_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_emmc_clk_en When HIGH, disable clock
10	RW	0x0	clk_emmc_pll_clk_en When HIGH, disable clock
9	RW	0x0	clk_sdio_clk_en When HIGH, disable clock
8	RW	0x0	clk_sdio_pll_clk_en When HIGH, disable clock
7	RW	0x0	clk_sdmmc_clk_en When HIGH, disable clock
6	RW	0x0	clk_sdmmc_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_nandc_clk_en When HIGH, disable clock
4	RW	0x0	clk_nandc_pll_clk_en When HIGH, disable clock
3	RW	0x0	pclk_peri_clk_en When HIGH, disable clock
2	RW	0x0	hclk_peri_clk_en When HIGH, disable clock
1	RW	0x0	aclk_peri_clk_en When HIGH, disable clock
0	RW	0x0	logic_peri_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON9

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	pclk_mac_en When HIGH, disable clock
14	RW	0x0	pclk_peri_niu_en When HIGH, disable clock
13	RW	0x0	hclk_host_arb_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
12	RW	0x0	hclk_host_en When HIGH, disable clock
11	RW	0x0	hclk_otg_en When HIGH, disable clock
10	RW	0x0	hclk_sfc_en When HIGH, disable clock
9	RW	0x0	hclk_emmc_en When HIGH, disable clock
8	RW	0x0	hclk_sdio_en When HIGH, disable clock
7	RW	0x0	hclk_sdmmc_en When HIGH, disable clock
6	RW	0x0	hclk_nanc_en When HIGH, disable clock
5	RW	0x0	hclk_peri_niu_en When HIGH, disable clock
4	RW	0x0	aclk_mac_en When HIGH, disable clock
3	RW	0x0	aclk_peribus_niu_en When HIGH, disable clock
2	RW	0x0	aclk_peri_niu_en When HIGH, disable clock
1	RW	0x0	clk_mac_ref_en When HIGH, disable clock
0	RW	0x0	clk_mac_txrx_en When HIGH, disable clock

CRU_CLKGATE_CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s0_8ch_tx_out_clk_en When HIGH, disable clock
14	RW	0x0	clk_i2s0_8ch_tx_clk_en When HIGH, disable clock
13	RW	0x0	clk_i2s0_8ch_tx_divfrac_clk_en When HIGH, disable clock
12	RW	0x0	clk_i2s0_8ch_tx_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_spdifrx_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
10	RW	0x0	clk_spdifrx_divfrac_clk_en When HIGH, disable clock
9	RW	0x0	clk_spdifrx_pll_clk_en When HIGH, disable clock
8	RW	0x0	clk_spdiftx_clk_en When HIGH, disable clock
7	RW	0x0	clk_spdiftx_divfrac_clk_en When HIGH, disable clock
6	RW	0x0	clk_spdiftx_pll_clk_en When HIGH, disable clock
5	RW	0x0	clk_pdm_clk_en When HIGH, disable clock
4	RW	0x0	clk_pdm_divfrac_clk_en When HIGH, disable clock
3	RW	0x0	clk_pdm_pll_clk_en When HIGH, disable clock
2	RW	0x0	pclk_audio_clk_en When HIGH, disable clock
1	RW	0x0	hclk_audio_clk_en When HIGH, disable clock
0	RW	0x0	logic_audio_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON11

Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s2_8ch_tx_out_clk_en When HIGH, disable clock
14	RW	0x0	clk_i2s2_8ch_tx_clk_en When HIGH, disable clock
13	RW	0x0	clk_i2s2_8ch_tx_divfrac_clk_en When HIGH, disable clock
12	RW	0x0	clk_i2s2_8ch_tx_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_i2s1_8ch_rx_out_clk_en When HIGH, disable clock
10	RW	0x0	clk_i2s1_8ch_rx_clk_en When HIGH, disable clock
9	RW	0x0	clk_i2s1_8ch_rx_divfrac_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
8	RW	0x0	clk_i2s1_8ch_rx_pll_clk_en When HIGH, disable clock
7	RW	0x0	clk_i2s1_8ch_tx_out_clk_en When HIGH, disable clock
6	RW	0x0	clk_i2s1_8ch_tx_clk_en When HIGH, disable clock
5	RW	0x0	clk_i2s1_8ch_tx_divfrac_clk_en When HIGH, disable clock
4	RW	0x0	clk_i2s1_8ch_tx_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_i2s0_8ch_rx_out_clk_en When HIGH, disable clock
2	RW	0x0	clk_i2s0_8ch_rx_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2s0_8ch_rx_divfrac_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2s0_8ch_rx_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON12

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	clk_i2s0_2ch_out_clk_en When HIGH, disable clock
14	RW	0x0	clk_i2s0_2ch_clk_en When HIGH, disable clock
13	RW	0x0	clk_i2s0_2ch_divfrac_clk_en When HIGH, disable clock
12	RW	0x0	clk_i2s0_2ch_pll_clk_en When HIGH, disable clock
11	RW	0x0	clk_i2s3_8ch_rx_out_clk_en When HIGH, disable clock
10	RW	0x0	clk_i2s3_8ch_rx_clk_en When HIGH, disable clock
9	RW	0x0	clk_i2s3_8ch_rx_divfrac_clk_en When HIGH, disable clock
8	RW	0x0	clk_i2s3_8ch_rx_pll_clk_en When HIGH, disable clock
7	RW	0x0	clk_i2s3_8ch_tx_out_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
6	RW	0x0	clk_i2s3_8ch_tx_clk_en When HIGH, disable clock
5	RW	0x0	clk_i2s3_8ch_tx_divfrac_clk_en When HIGH, disable clock
4	RW	0x0	clk_i2s3_8ch_tx_pll_clk_en When HIGH, disable clock
3	RW	0x0	clk_i2s2_8ch_rx_out_clk_en When HIGH, disable clock
2	RW	0x0	clk_i2s2_8ch_rx_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2s2_8ch_rx_divfrac_clk_en When HIGH, disable clock
0	RW	0x0	clk_i2s2_8ch_rx_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON13

Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	clk_i2s0_2ch_out_ioe 1'b0: enable clk_i2s0_2ch_out to PAD MCLK 1'b1: disable clk_i2s0_2ch_out to PAD MCLK
7	RW	0x0	clk_i2s1_8ch_rx_out_ioe 1'b0: enable clk_i2s1_8ch_rx_out to PAD MCLK 1'b1: disable clk_i2s1_8ch_rx_out to PAD MCLK
6	RW	0x0	clk_i2s1_8ch_tx_out_ioe 1'b0: enable clk_i2s1_8ch_tx_out to PAD MCLK 1'b1: disable clk_i2s1_8ch_tx_out to PAD MCLK
5	RW	0x0	clk_i2s0_8ch_rx_out_ioe 1'b0: enable clk_i2s0_8ch_rx_out to PAD MCLK 1'b1: disable clk_i2s0_8ch_rx_out to PAD MCLK
4	RW	0x0	clk_i2s0_8ch_tx_out_ioe 1'b0: enable clk_i2s0_8ch_tx_out to PAD MCLK 1'b1: disable clk_i2s0_8ch_tx_out to PAD MCLK
3	RW	0x0	clk_i2s1_2ch_out_clk_en When HIGH, disable clock
2	RW	0x0	clk_i2s1_2ch_clk_en When HIGH, disable clock
1	RW	0x0	clk_i2s1_2ch_divfrac_clk_en When HIGH, disable clock

Bit	Attr	Reset Value	Description
0	RW	0x0	clk_i2s1_2ch_pll_clk_en When HIGH, disable clock

CRU_CLKGATE_CON14

Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	pclk_acodec_en When HIGH, disable clock
11	RW	0x0	pclk_audio_niu_en When HIGH, disable clock
10	RW	0x0	hclk_vad_en When HIGH, disable clock
9	RW	0x0	hclk_i2s1_2ch_en When HIGH, disable clock
8	RW	0x0	hclk_i2s0_2ch_en When HIGH, disable clock
7	RW	0x0	hclk_i2s3_8ch_en When HIGH, disable clock
6	RW	0x0	hclk_i2s2_8ch_en When HIGH, disable clock
5	RW	0x0	hclk_i2s1_8ch_en When HIGH, disable clock
4	RW	0x0	hclk_i2s0_8ch_en When HIGH, disable clock
3	RW	0x0	hclk_spdifrx_en When HIGH, disable clock
2	RW	0x0	hclk_spdiftx_en When HIGH, disable clock
1	RW	0x0	hclk_pdm_en When HIGH, disable clock
0	RW	0x0	hclk_audio_niu_en When HIGH, disable clock

CRU_SSCGTBL0_3

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl0_3 7-0: table0 15-8: table1 23-16: table2 31-24: table3

CRU_SSCGTBL4_7

Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl4_7 7-0: table4 15-8: table5 23-16: table6 31-24: table7

CRU_SSCGTBL8_11

Address: Operational Base + offset (0x0388)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl8_11 7-0: table8 15-8: table9 23-16: table10 31-24: table11

CRU_SSCGTBL12_15

Address: Operational Base + offset (0x038c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl12_15 7-0: table12 15-8: table13 23-16: table14 31-24: table15

CRU_SSCGTBL16_19

Address: Operational Base + offset (0x0390)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl16_19 7-0: table16 15-8: table17 23-16: table18 31-24: table19

CRU_SSCGTBL20_23

Address: Operational Base + offset (0x0394)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl20_23 7-0: table20 15-8: table21 23-16: table22 31-24: table23

CRU_SSCGTBL24_27

Address: Operational Base + offset (0x0398)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl24_27 7-0: table24 15-8: table25 23-16: table26 31-24: table27

CRU_SSCGTBL28_31

Address: Operational Base + offset (0x039c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl28_31 7-0: table28 15-8: table29 23-16: table30 31-24: table31

CRU_SSCGTBL32_35

Address: Operational Base + offset (0x03a0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl32_35 7-0: table32 15-8: table33 23-16: table34 31-24: table35

CRU_SSCGTBL36_39

Address: Operational Base + offset (0x03a4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl36_39 7-0: table36 15-8: table37 23-16: table38 31-24: table39

CRU_SSCGTBL40_43

Address: Operational Base + offset (0x03a8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl40_43 7-0: table40 15-8: table41 23-16: table42 31-24: table43

CRU_SSCGTBL44_47

Address: Operational Base + offset (0x03ac)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl44_47 7-0: table44 15-8: table45 23-16: table46 31-24: table47

CRU_SSCGTBL48_51

Address: Operational Base + offset (0x03b0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl48_51 7-0: table48 15-8: table49 23-16: table50 31-24: table51

CRU_SSCGTBL52_55

Address: Operational Base + offset (0x03b4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl52_55 7-0: table52 15-8: table53 23-16: table54 31-24: table55

CRU_SSCGTBL56_59

Address: Operational Base + offset (0x03b8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl56_59 7-0: table56 15-8: table57 23-16: table58 31-24: table59

CRU_SSCGTBL60_63

Address: Operational Base + offset (0x03bc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl60_63 7-0: table60 15-8: table61 23-16: table62 31-24: table63

CRU_SSCGTBL64_67

Address: Operational Base + offset (0x03c0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl64_67 7-0: table64 15-8: table65 23-16: table66 31-24: table67

CRU_SSCGTBL68_71

Address: Operational Base + offset (0x03c4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl68_71 7-0: table68 15-8: table69 23-16: table70 31-24: table71

CRU_SSCGTBL72_75

Address: Operational Base + offset (0x03c8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl72_75 7-0: table72 15-8: table73 23-16: table74 31-24: table75

CRU_SSCGTBL76_79

Address: Operational Base + offset (0x03cc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl76_79 7-0: table76 15-8: table77 23-16: table78 31-24: table79

CRU_SSCGTBL80_83

Address: Operational Base + offset (0x03d0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl80_83 7-0: table80 15-8: table81 23-16: table82 31-24: table83

CRU_SSCGTBL84_87

Address: Operational Base + offset (0x03d4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl84_87 7-0: table84 15-8: table85 23-16: table86 31-24: table87

CRU_SSCGTBL88_91

Address: Operational Base + offset (0x03d8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl88_91 7-0: table88 15-8: table89 23-16: table90 31-24: table91

CRU_SSCGTBL92_95

Address: Operational Base + offset (0x03dc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl92_95 7-0: table92 15-8: table93 23-16: table94 31-24: table95

CRU_SSCGTBL96_99

Address: Operational Base + offset (0x03e0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl96_99 7-0: table96 15-8: table97 23-16: table98 31-24: table99

CRU_SSCGTBL100_103
Address: Operational Base + offset (0x03e4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl100_103 7-0: table100 15-8: table101 23-16: table102 31-24: table103

CRU_SSCGTBL104_107
Address: Operational Base + offset (0x03e8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl104_107 7-0: table104 15-8: table105 23-16: table106 31-24: table107

CRU_SSCGTBL108_111
Address: Operational Base + offset (0x03ec)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl108_111 7-0: table108 15-8: table109 23-16: table110 31-24: table111

CRU_SSCGTBL112_115
Address: Operational Base + offset (0x03f0)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl112_115 7-0: table112 15-8: table113 23-16: table114 31-24: table115

CRU_SSCGTBL116_119

Address: Operational Base + offset (0x03f4)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl116_119 7-0: table116 15-8: table117 23-16: table118 31-24: table119

CRU_SSCGTBL120_123

Address: Operational Base + offset (0x03f8)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl120_123 7-0: table120 15-8: table121 23-16: table122 31-24: table123

CRU_SSCGTBL124_127

Address: Operational Base + offset (0x03fc)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	sscgtbl124_127 7-0: table124 15-8: table125 23-16: table126 31-24: table127

CRU_SOFTRST_CON0

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	R/W SC	0x0	resetn_l2_req When HIGH, will assert core nL2RESET(Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
14	R/W SC	0x0	aresetn_bus_niu_req When HIGH, reset relative logic
13	R/W SC	0x0	resetn_core_noc_req When HIGH, reset relative logic
12	RW	0x0	resetn_topdbg_req When HIGH, will assert core nPRESETDBG(Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
11:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	resetn_core3_req When HIGH, will assert core3 nCORERESET[3](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
6	RW	0x0	resetn_core2_req When HIGH, will assert core2 nCORERESET[2](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
5	RW	0x0	resetn_core1_req When HIGH, will assert core1 nCORERESET[1](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
4	R/W SC	0x0	resetn_core0_req When HIGH, will assert core0 nCORERESET[0](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
3	RW	0x0	resetn_corepo3_req When HIGH, will assert core3 nCPUPORESET[3](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
2	RW	0x0	resetn_corepo2_req When HIGH, will assert core2 nCPUPORESET[2](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
1	RW	0x0	resetn_corepo1_req When HIGH, will assert core1 nCPUPORESET[1](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)
0	R/W SC	0x0	resetn_corepo0_req When HIGH, will assert core0 nCPUPORESET[0](Please refer to ARM Cortex-A35 Processor Revision: r0p2 Technical Reference Manual)

CRU_SOFTRST_CON1

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RO	0x0	reserved
14	RW	0x0	presetn_ddrphy_req When HIGH, reset relative logic
13	RW	0x0	resetn_ddrphy_clkdiv_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
12	RW	0x0	resetn_ddrphy_req When HIGH, reset relative logic
11	RW	0x0	resetn_ddr_stdby_req When HIGH, reset relative logic
10	RW	0x0	presetn_ddr_stdby_req When HIGH, reset relative logic
9	RW	0x0	presetn_ddr_monitor_req When HIGH, reset relative logic
8	RO	0x0	reserved
7	RW	0x0	resetn_msch_req When HIGH, reset relative logic
6	RW	0x0	presetn_ddrupctl_req When HIGH, reset relative logic
5	RO	0x0	reserved
4	RW	0x0	resetn_ddrupctl_req When HIGH, reset relative logic
3	RW	0x0	presetn_core_grf_req When HIGH, reset relative logic
2	RW	0x0	aresetn_core_prf_req When HIGH, reset relative logic
1	RW	0x0	resetn_core_pvtm_req When HIGH, reset relative logic
0	RW	0x0	resetn_dap_req When HIGH, reset relative logic

CRU_SOFTRST_CON2

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_uart1_req When HIGH, reset relative logic
14	RW	0x0	presetn_uart1_req When HIGH, reset relative logic
13	RW	0x0	resetn_uart0_req When HIGH, reset relative logic
12	RW	0x0	presetn_uart0_req When HIGH, reset relative logic
11	RW	0x0	hresetn_rom_req When HIGH, reset relative logic
10	RW	0x0	aresetn_gic_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
9	RW	0x0	aresetn_intmem_req When HIGH, reset relative logic
8	RW	0x0	dresetn_vop_req When HIGH, reset relative logic
7	RW	0x0	hresetn_vop_req When HIGH, reset relative logic
6	RW	0x0	aresetn_vop_req When HIGH, reset relative logic
5	RW	0x0	resetn_crypto_apk_req When HIGH, reset relative logic
4	RW	0x0	resetn_crypto_core_req When HIGH, reset relative logic
3	RW	0x0	hresetn_crypto_req When HIGH, reset relative logic
2	RW	0x0	aresetn_crypto_req When HIGH, reset relative logic
1	RW	0x0	presetn_bus_niu_req When HIGH, reset relative logic
0	RW	0x0	hresetn_bus_niu_req When HIGH, reset relative logic

CRU_SOFTRST_CON3
Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_pwm_req When HIGH, reset relative logic
14	RW	0x0	presetn_pwm_req When HIGH, reset relative logic
13	RW	0x0	resetn_i2c3_req When HIGH, reset relative logic
12	RW	0x0	presetn_i2c3_req When HIGH, reset relative logic
11	RW	0x0	resetn_i2c2_req When HIGH, reset relative logic
10	RW	0x0	presetn_i2c2_req When HIGH, reset relative logic
9	RW	0x0	resetn_i2c1_req When HIGH, reset relative logic
8	RW	0x0	presetn_i2c1_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
7	RW	0x0	resetn_i2c0_req When HIGH, reset relative logic
6	RW	0x0	presetn_i2c0_req When HIGH, reset relative logic
5	RW	0x0	resetn_uart4_req When HIGH, reset relative logic
4	RW	0x0	presetn_uart4_req When HIGH, reset relative logic
3	RW	0x0	resetn_uart3_req When HIGH, reset relative logic
2	RW	0x0	presetn_uart3_req When HIGH, reset relative logic
1	RW	0x0	resetn_uart2_req When HIGH, reset relative logic
0	RW	0x0	presetn_uart2_req When HIGH, reset relative logic

CRU_SOFTRST_CON4
Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	resetn_timer5_req When HIGH, reset relative logic
14	RW	0x0	resetn_timer4_req When HIGH, reset relative logic
13	RW	0x0	resetn_timer3_req When HIGH, reset relative logic
12	RW	0x0	resetn_timer2_req When HIGH, reset relative logic
11	RW	0x0	resetn_timer1_req When HIGH, reset relative logic
10	RW	0x0	resetn_timer0_req When HIGH, reset relative logic
9	RW	0x0	presetn_timer0_req When HIGH, reset relative logic
8	RW	0x0	resetn_tsadc_req When HIGH, reset relative logic
7	RW	0x0	presetn_tsadc_req When HIGH, reset relative logic
6	RW	0x0	presetn_saradc_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
5	RW	0x0	resetn_spi2_req When HIGH, reset relative logic
4	RW	0x0	presetn_spi2_req When HIGH, reset relative logic
3	RW	0x0	resetn_spi1_req When HIGH, reset relative logic
2	RW	0x0	presetn_spi1_req When HIGH, reset relative logic
1	RW	0x0	resetn_spi0_req When HIGH, reset relative logic
0	RW	0x0	presetn_spi0_req When HIGH, reset relative logic

CRU_SOFTRST_CON5

Address: Operational Base + offset (0x0414)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	presetn_usb_grf_req When HIGH, reset relative logic
14	RW	0x0	resetn_pmu_pvtm_req When HIGH, reset relative logic
13	RW	0x0	pmu_srstn_req When HIGH, reset relative logic
12	RW	0x0	presetn_usbsdmmc_det_req When HIGH, reset relative logic
11	RW	0x0	presetn_grf_req When HIGH, reset relative logic
10	RW	0x0	presetn_gpio4_req When HIGH, reset relative logic
9	RW	0x0	presetn_gpio3_req When HIGH, reset relative logic
8	RW	0x0	presetn_gpio2_req When HIGH, reset relative logic
7	RW	0x0	presetn_gpio1_req When HIGH, reset relative logic
6	RW	0x0	presetn_gpio0_req When HIGH, reset relative logic
5	RO	0x0	reserved
4	RW	0x0	otp_phy_srstn_req When HIGH, reset relative logic
3	RW	0x0	presetn_otp_phy_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	resetn_otp_ns_user_req When HIGH, reset relative logic
1	RW	0x0	resetn_otp_ns_sbpi_req When HIGH, reset relative logic
0	RW	0x0	presetn_otp_ns_req When HIGH, reset relative logic

CRU_SOFTRST_CON6

Address: Operational Base + offset (0x0418)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	hresetn_usb2host_arb_req When HIGH, reset relative logic
14	RW	0x0	hresetn_usb2host_req When HIGH, reset relative logic
13	RW	0x0	resetn_usb2otg_adp_req When HIGH, reset relative logic
12	RW	0x0	resetn_usb2otg_req When HIGH, reset relative logic
11	RW	0x0	hresetn_usb2otg_req When HIGH, reset relative logic
10	RW	0x0	presetn_peri_niu_req When HIGH, reset relative logic
9	RW	0x0	hresetn_peri_niu_req When HIGH, reset relative logic
8	RW	0x0	aresetn_peri_niu_req When HIGH, reset relative logic
7:2	RO	0x0	reserved
1	RW	0x0	presetn_cpu_boost_req When HIGH, reset relative logic
0	RW	0x0	resetn_cpu_boost_req When HIGH, reset relative logic

CRU_SOFTRST_CON7

Address: Operational Base + offset (0x041c)

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

Bit	Attr	Reset Value	Description
12	RW	0x0	nresetn_nandc_req When HIGH, reset relative logic
11	RW	0x0	hresetn_nandc_req When HIGH, reset relative logic
10	RW	0x0	hresetn_sdmmc_req When HIGH, reset relative logic
9	RW	0x0	resetn_sfc_req When HIGH, reset relative logic
8	RW	0x0	hresetn_sfc_req When HIGH, reset relative logic
7	RW	0x0	hresetn_emmc_req When HIGH, reset relative logic
6	RW	0x0	hresetn_sdio_req When HIGH, reset relative logic
5	RW	0x0	utmi1_srst_req When HIGH, reset relative logic
4	RW	0x0	utmi0_srst_req When HIGH, reset relative logic
3	RW	0x0	usbphypor_rst_req When HIGH, reset relative logic
2	RW	0x0	resetn_usb2host_utmi_req When HIGH, reset relative logic
1	RW	0x0	resetn_usb2host_ehciphy_req When HIGH, reset relative logic
0	RW	0x0	hresetn_usb2host_aux_req When HIGH, reset relative logic

CRU_SOFTRST_CON8

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	mresetn_i2s2_8ch_tx_req When HIGH, reset relative logic
14	RW	0x0	hresetn_i2s2_8ch_req When HIGH, reset relative logic
13	RW	0x0	mresetn_i2s1_8ch_rx_req When HIGH, reset relative logic
12	RW	0x0	mresetn_i2s1_8ch_tx_req When HIGH, reset relative logic
11	RW	0x0	hresetn_i2s1_8ch_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
10	RW	0x0	mresetn_i2s0_8ch_rx_req When HIGH, reset relative logic
9	RW	0x0	mresetn_i2s0_8ch_tx_req When HIGH, reset relative logic
8	RW	0x0	hresetn_i2s0_8ch_req When HIGH, reset relative logic
7	RW	0x0	mresetn_spdifrx_req When HIGH, reset relative logic
6	RW	0x0	hresetn_spdifrx_req When HIGH, reset relative logic
5	RW	0x0	mresetn_spdiftx_req When HIGH, reset relative logic
4	RW	0x0	hresetn_spdiftx_req When HIGH, reset relative logic
3	RW	0x0	mresetn_pdm_req When HIGH, reset relative logic
2	RW	0x0	hresetn_pdm_req When HIGH, reset relative logic
1	RW	0x0	presetn_audio_niu_req When HIGH, reset relative logic
0	RW	0x0	hresetn_audio_niu_req When HIGH, reset relative logic

CRU_SOFTRST_CON9

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:10	RO	0x0	reserved
9	RW	0x0	presetn_acodec_req When HIGH, reset relative logic
8	RW	0x0	hresetn_vad_req When HIGH, reset relative logic
7	RW	0x0	mresetn_i2s1_2ch_req When HIGH, reset relative logic
6	RW	0x0	hresetn_i2s1_2ch_req When HIGH, reset relative logic
5	RW	0x0	mresetn_i2s0_2ch_req When HIGH, reset relative logic
4	RW	0x0	hresetn_i2s0_2ch_req When HIGH, reset relative logic
3	RW	0x0	mresetn_i2s3_8ch_rx_req When HIGH, reset relative logic

Bit	Attr	Reset Value	Description
2	RW	0x0	mresetn_i2s3_8ch_tx_req When HIGH, reset relative logic
1	RW	0x0	hresetn_i2s3_8ch_req When HIGH, reset relative logic
0	RW	0x0	mresetn_i2s2_8ch_rx_req When HIGH, reset relative logic

CRU_SDMMC_CON0

Address: Operational Base + offset (0x0480)

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

CRU_SDMMC_CON1

Address: Operational Base + offset (0x0484)

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

CRU_SDIO_CON0

Address: Operational Base + offset (0x0488)

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

CRU_SDIO_CON1

Address: Operational Base + offset (0x048c)

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

CRU_EMMC_CON0

Address: Operational Base + offset (0x0490)

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

CRU_EMMC_CON1

Address: Operational Base + offset (0x0494)

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

2.5 Application Notes

2.5.1 PLL usage

FBDIV, POSTDIV1, BYPASS can be configured by programming CRU_xPLL_CON0.
DSMPD, REFDIV, POSTDIV2 can be configured by programming CRU_xPLL_CON1.
FRAC can be configured by programming CRU_xPLL_CON2. If DSMPD = 1, PLL is in integer mode. If DSMPD = 0, PLL is in fractional mode.

A. PLL integer mode configuration

FOUTVCO = (FREF / REFDIV) * FBDIV
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

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

And then
FOUTVCO = (FREF / REFDIV) * FBDIV = 24/6*175=700
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)=700/1/1=700

B. PLL fractional mode configuration

FOUTVCO = (FREF / REFDIV) * (FBDIV + FRAC / (2^24))
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)
When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

```
        DSMPD = 0
        REFDIV = 1
        FBDIV   = 40
        FRAC    = 24'hf5c28f
        POSTDIV1=2
        POSTDIV2=1
```

And then
FOUTVCO = (FREF / REFDIV) * (FBDIV + FRAC / (2^24)) = 983.04
FOUTPOSTDIV = FOUTVCO / (POSTDIV1*POSTDIV2)=983.04/(2*1)=491.52

C. PLL setting consideration

- VCO output clock from 800MHz to 3.2GHz
- The value of POSTDIV1 should always be greater than or equal to POSTDIV2
- For lowest power operation, the minimum VCO and FREF frequencies should be used. For minimum jitter operation, the highest VCO and FREF frequencies should be used.
- The supply rejection will be worse at the low end of the VCO range so care should be

taken to keep the supply clean for low power applications.

2.5.2 PLL frequency change and lock check

PLL lock state can be checked in CRU_xPLL_CON1[10] register. The lock state is high when both original hardware PLL lock and PLL counter lock are high. The max lock time for PLL is 500 REF_CLK.

The PLL counter lock initial value is GLB_PLL_LOCK [15:0], the default value is 15000. It recommends configure GLB_PLL_LOCK [15:0] to a smaller value after power up(for example: 500).

User should configure as following steps to change the PLL output frequency

- Change PLL from normal to slow mode by programming CRU_MODE[9:0].
- Change PLL setting.
- Wait until PLL is lock state by checking CRU_xPLL_CON1[10] register or after delay about 500 REF_CLK
- Change PLL into normal mode.

By default, user don't have to configure the register to power down PLL when change the PLL output frequency. When any RFDIV/FBDIV/FRACDIV is changed, CRU will assert PLL power down and release after 1us automatic. In addition, user also can configure CRU_xPLL_CON1[13] to power down PLL.

User can set CRU_XPLL_CON1[15] to 1'b1 to disable the PLL power down automatic. In this case, user should configure CRU_xPLL_CON1[14] to power down PLL before changing PLL setting, and release at least 1us after valid settings, referring to the following figure.

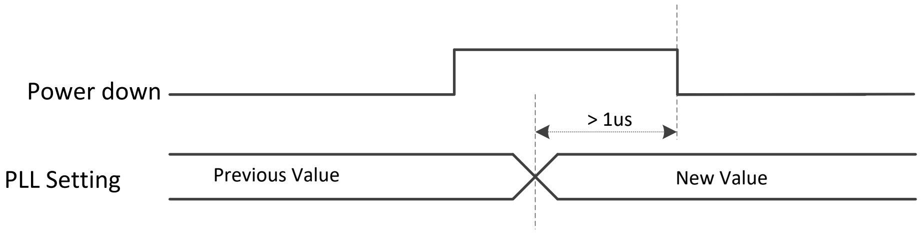


Fig. 2-5 PLL setting change timing

User also can use HWFFC (hardware fast frequency change) to change the APLL frequency as following steps:

- Set the CRU_HWFFC_INTST(interrupt related) and CRU_HWFFC_TH.
- Enable hwffc_mode(CRU_HWFFC_CON0[9]).
- Set clk_core_div for backup PLL (CRU_HWFFC_CON0[3:0]).
- Set the new APLL setting to CRU_APLL_CON0, CRU_APLL_CON1, this setting will not take effect until hwffc_req is asserted.
- Trigger hwffc_req (CRU_HWFFC_CON0[8]), and Poll CRU_HWFFC_CON0[8]=0x1.
- HWFFC is done when CRU_HWFFC_CON0[8] is cleared and HWFFC interrupt happened (or interrupt status is asserted).

When HWFFC mode is enabled, some setting of APLL and core clock generate unit (configured by CRU_APLL_CON0, CRU_APLL_CON1 and CRU_CLKSEL_CON0) are also controlled by HWFFC module. The actually setting can be read by registers CRU_APLL_CON0_S, CRU_APLL_CON1_S and CRU_CLKSEL_CON0_S.

2.5.3 Fractional divider usage

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

Table 2-1 Source Clock Limitation of Fractional Divider

Clock Name	Fractional divider source clock limitation
dclk_vop	270Mhz
clk_pdm	800Mhz

Clock Name	Fractional divider source clock limitation
clk_spdiftx	800Mhz
clk_spdifrx	800Mhz
clk_i2s0/1/2/3_8ch_tx	800Mhz
clk_i2s0/1/2/3_8ch_rx	800Mhz
clk_i2s0/1_2ch	800Mhz

2.5.4 Divfree50 divider usage

Some IPs, such as NAND, EMMC, SDIO and SDMMC need clock of 50% duty cycle, then Divfree50 divider is used.

2.5.5 DivfreeNP5 divider usage

Some IPs, such as UART needs some special frequency, then DivfreeNP5 is used. UART with baud rate of 4Mbps need use this divider to generate 64MHz clock from 480MHz of USBPHY PLL.

2.5.6 Global software reset usage

Two global software resets are designed in the chip, you can program CRU_GLB_SRST_FST [15:0] as 0xfdb9 to assert the first global software reset glb_srstn_1 and program CRU_GLB_SRST_SND [15:0] as 0xeca8 to assert the second global software reset glb_srstn_2. These two software resets are self-de-asserted by hardware. Resetting hold timing of global software reset (glb_srstn_1, glb_srstn_2, soc_wdt_rstn, soc_tsadc_rstn) can be programmable up to 178.9s.

Glb_srstn_1 resets almost all logic.

Glb_srstn_2 resets almost all logic except GRF and GPIOs.

2.5.7 Software reset usage

Almost all software reset are controlled by CRU registers. Especially for core, when it assert warm reset request, the corresponding bit of nCPUPORESET and nCORERESET will also be asserted.

For implementation issue, Some IPs clock have following frequency limitation. User should reduce the frequency to less than or equal to the limitation when assert the software reset.

Table 2-2 Clock Limitation of reset

Reset request	Clock Name	limitation
resetn_msch_req(CRU_SOFTRST_CON1[7])	ddrphy_dfi_clk4x	1200Mhz
resetn_crypto_apk_req(CRU_SOFTRST_CON2[5])	clk_crypto_apk	275Mhz
aresetn_dmac1_req(SGRF_SOFTRST_CON[12])	aclk_bus	200Mhz
resetn_ddrphy_clkdiv_req(CRU_SOFTRST_CON1[13])	ddrphy_dfi_clk4x	300Mhz
CPU warm reset	clk_core	800Mhz

2.5.8 SSCG usage

There are some scenes where SSCG should not be enabled. One scene is in communication where a fixed frequency is required. Another scene is a system requiring a clock with low long-term jitter.

When SSCG is usage, the PLL should be configured to fractional mode firstly for spread spectrum capability.

A. SSCG use Internal Point Table

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

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

The modulation amplitude is controlled by the value of ssmod_spread. A ssmod_spread value of 5'd0 turns off the modulation. A ssmod_spread value of 5'd31 (5'b11111) gives maximum

modulation while a value of 5'd1 gives minimum modulation.

The modulation amplitude can be calculated from the value of modulation by:

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

The modulation direction is determined by the ssmod_downspread bit.

- ■ ssmod_downspread=1'b1, then down spread mode is used. If ssmod_spread =

  5’d29. Then, the maximum PLL frequency is the nominally programmed value FNOM, and the minimum value is given by FNOM* (1–0.029).
  - ssmod_downspread=1’b0, then center spread mode is used. If ssmod_spread = 5’d29. Then, the maximum PLL frequency would be determined by FNOM*(1+0.029) and the minimum frequency by FNOM* (1–0.029).

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

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

When use internal point table. The frequency will change as following during 128 point:

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

Spread spectrum modulator is implemented by repeating as above.

- Setting ssmod_divval (CRU_XPLL_CON3[7:4])

The frequency of modulation FMOD= FREF / (Point number*REFDIV*ssmod_divval). The FMOD is typically set above 32kHz and below the maximum frequency for modulation fidelity, which is determined by the PLL bandwidth. The maximum modulation frequency is conservatively set at FREF/(200*REFDIV).

When FREF=24Mhz and REFDIV= 1, the value of ssmod_divval can be 5, then the FMOD is 37.5Khz.

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

A. SSCG use External Point Table

In addition to the internal shape table, tables external to the SSMOD can be accessed. This enables customization tables in both shape and the number of sample points for the envelope wave form up to 128 data points. The external table of 128 data points can be configured from CRU_SSCGTBL0_3~CRU_SSCGTBL124_127.

User can use SSMOD with external point table as following steps:

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

Same with internal point table usage.

- Setting ssmod_sel_ext_wave (CRU_XPLL_CON4[0])=1’b1
- Setting ssmod_ext_maxaddr(CRU_XPLL_CON4[15:8]) and table0~table127(CRU_SSCGTBL0_3~CRU_SSCGTBL124_127)

ssmod_ext_maxaddr is the maximum table address. For example, if the number of points describing the envelope shape is 128, the ssmod_ext_maxaddr should be configured to 127. The table address circulate over the range 0 to 127.

The table0~table127 must be 8 bit numbers in the form of sign and magnitude

  - 1.00 is represented by 8’b01111111, it is corresponded to maximum frequency.
  - -1.00 in represented in the table by 8’b11111111, it is corresponded to minimum frequency.
  - 0.5 in represented in the table by 8’b00111111.
  - -0.5 in represented in the table by 8’b10111111.

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

- Setting ssmod_divval (CRU_XPLL_CON3[7:4])

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

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

Chapter 3 Cortex-A35

3.1 Overview

The 3308 has a quad-core Cortex-A35 cluster with 256K L2 memory. Cortex-A35 processor, which is a mid-range, low-power processor that implements the ARMv8-A architecture.
The Cortex-A35 processor includes following features:

- Full implementation of the ARMv8-A A64, A32, and T32 instruction sets
- Both the AArch32 and AArch64 execution states at all Exception levels (EL0 to EL3)
- In-order pipeline with direct and indirect branch prediction
- Separate Level 1 (L1) data and instruction side memory systems with a Memory Management Unit(MMU)
- Level 2 (L2) memory system that provides cluster memory coherency
- L2 cache
- TrustZone
- Support data engine that implements the Advanced SIMD and floating-point architecture support
- Support Cryptographic Extension
- ARMv8 debug logic with v7 Debug memory map
- Support Generic Interrupt Controller (GIC) CPU interface to connect to an external distributor
- Generic Timers supporting 64-bit count input from an external system counter

The configuration details are shown in following tables:

Table 3-1 CPU Configuration

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

3.2 Block Diagram

The Cortex-A35 sub system is shown in Figure 1-1. As illustrated, quad-core Cortex-A35 connects to system bus through SCU-L2 which can handle with CDC (clock domain crossing) issue.

The Cortex-A35 is connected with system counter, which can run under a constant frequency clock, for PPI interrupt generation.

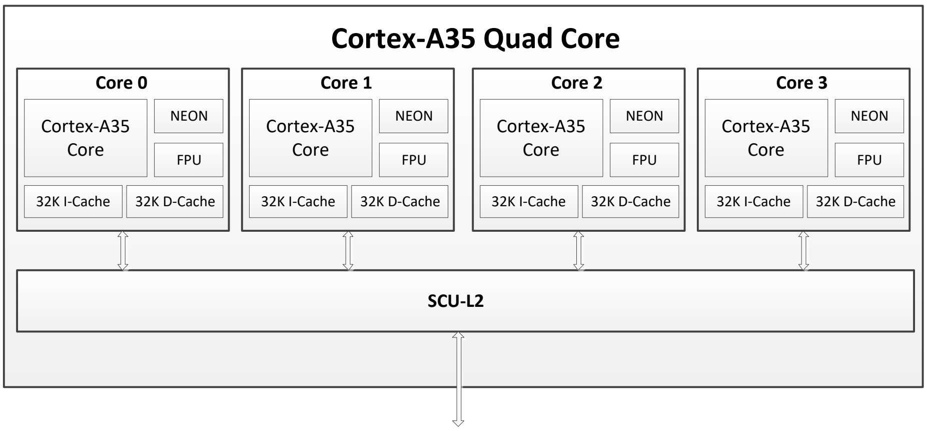


Fig. 3-1 Block Diagram

3.3 Function Description

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

Chapter 4 AXI PERF

4.1 Overview

One AXI_PERF blocks are used in the system: CPU_AXI_PERF for performance statistic on AXI bus of CPU. Following is the features of AXI_PERF IP:

- AXI read statistic
  - Support max read latency of one ID
  - Support the statistic of the average latency of one ID
  - Support count the burst which the read latency is more than A value
  - Support read bandwidth statistic for one or all ID
  - Support read real bandwidth statistic and DDR align bandwidth statistic
- AXI write statistic
  - Support write bandwidth statistic for one or all ID
  - Support write real bandwidth statistic and DDR align bandwidth statistic
- AXI read and write address monitor
  - Support monitor read from appointed address area for one or some ID
  - Support monitor write to appointed address area for one or some ID
  - Interrupt generation

4.2 Block Diagram

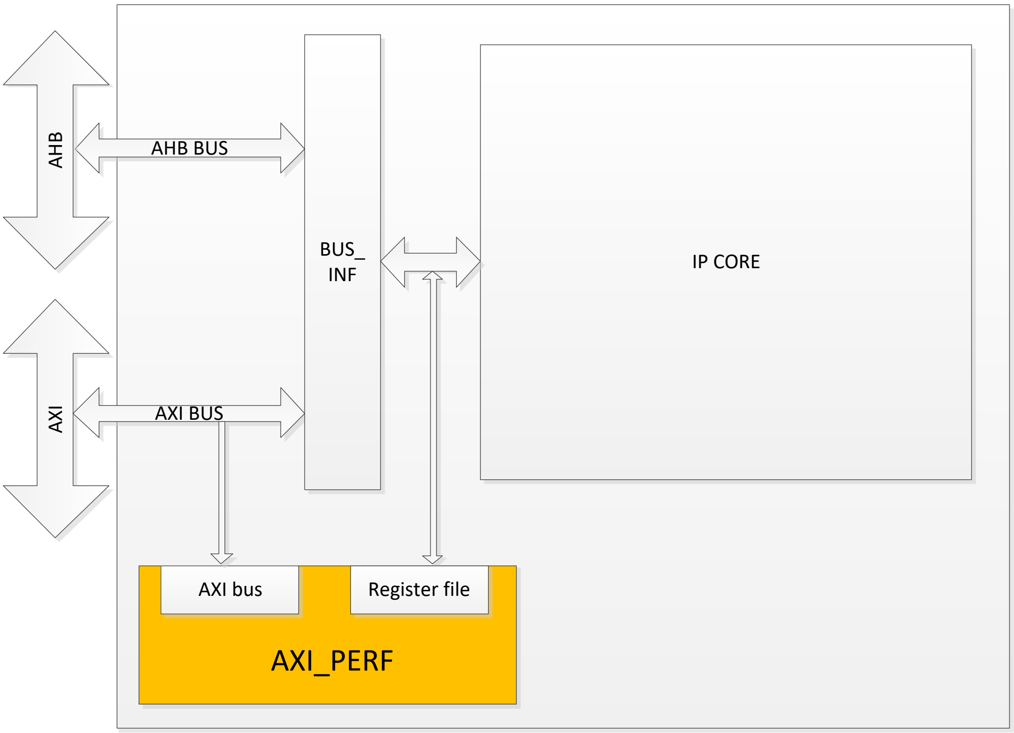


Fig. 4-1 AXI_PERF block diagram

4.3 Register Description of CPU_AXI_PERF

Please refer to Chapter GRF/CORE GRF.

4.4 Application Notes

4.4.1 Start and finish flow

- Assert sw_axi_perf_clr_e 1，and then assert sw_axi_perf_clr_e 0
- Config all the signals which need to be configured (sw_axi_perf_work_e keeps 0)
- Assert sw_axi_perf_work_e 1 after getting A frame end signal to start AXI_PERF
- Read back the statistic values after one or more frames
- Next loop

4.4.2 Read latency statistic case

The Read latency statistic is fit for one ID specify by sw_rd_latency_id only

Case1：The user wants to get the average read latency and max read latency of one frame and ID 1 at CPU. Here are the steps:

- Finish CPU AXI PERF clear step
- Assert sw_rd_latency_id 1, sw_rd_latency_thr 0, sw_axi_perf_frm_type 1（PERF_LATENCY_CTRL0=0x14）
- Start CPU AXI PERF
- Start CPU and wait for finish interrupt
- Read back PERF_WORKING_CNT. This value means how many cycles are used by this frame.
- Read back PERF_RD_MAX_LATENCY_NUM. This value means the max latency of one frame for ID 1.
- Read back PERF_RD_LATENCY_SAMP_NUM and PERF_RD_LATENCY_ACC_SUM. Use the formulation perf_rd_latency_average=perf_rd_latency_acc_sum/perf_rd_latency_samp_num；perf_rd_latency_average is the average latency of ID 1.
- Clear CPU AXI_PERF and wait for next statistic

Case2：The user wants to statistic the read average latency of every ID (assume ID from 0 to 9) and 60 frames at CPU. The burst to be counted need bigger than n. Here are the steps:

- Finish CPU AXI PERF clear step
- Assert sw_rd_latency_id 0, sw_rd_latency_thr n, sw_axi_perf_frm_type 0（PERF_LATENCY_CTRL0=0xn00）
- Start CPU AXI PERF
- Start CPU and run 60 frames continually
- Read back PERF_WORKING_CNT, this value which is divided by60 means the average cycle uses of running this scenario
- Read back PERF_RD_LATENCY_SAMP_NUM and PERF_RD_LATENCY_ACC_SUM. Use the formulation perf_rd_latency_average=perf_rd_latency_acc_sum/perf_rd_latency_samp_num；perf_rd_latency_average is the average latency of ID 0 by 60 frames.
- Clear CPU AXI_PERF
- Repeat the above steps and finish id1 to id9 statistic

4.4.3 Bandwidth statistic case

Case3：The user wants to get the single frame read and write bandwidth of ID 1 at CPU. Here are the steps:

- Finish CPU AXI PERF clear step
- Assert sw_ar_cnt_id_type sw_aw_cnt_id_type 1, sw_ar_count_id sw_aw_count_id 1，sw_axi_cnt_type 0, sw_axi_perf_frm_type 1(PERF_LATENCY_CTRL0=0x40 PERF_LATENCY_CTRL1=0x11c)
- Start CPU AXI PERF
- Start CPU and wait for finish interrupt
- Read back PERF_RD_AXI_TOTAL_BYTE, this value means the total read bytes of ID 1
- Read back PERF_WR_AXI_TOTAL_BYTE, this value means the total write bytes of ID 1
- Clear CPU AXI_PERF and wait for next statistic

Case4：The user wants to get the 60 frames read and write average bandwidth of all ID at CPU (assume the DDR type is LPDDR4). Here are the steps:

- Finish CPU AXI PERF clear step
- Assert sw_ar_cnt_id_type sw_aw_cnt_id_type 0，sw_addr_align_type 2 , sw_axi_cnt_type 1, sw_axi_perf_frm_type 0 (PERF_LATENCY_CTRL0=0x00 PERF_LATENCY_CTRL1=0x2)
- Start CPU AXI PERF
- Start CPU and run 60 frames continually
- Read back PERF_RD_AXI_TOTAL_BYTE, this value which is divided by60 means the average total read bytes of all ID
- Read back PERF_WR_AXI_TOTAL_BYTE, this value which is divided by60 means the average total write bytes of all ID
- Clear CPU AXI_PERF and wait for next statistic

Note:In wrap mode, DDR is always aligned. So, we must solve the deviation of bandwidth statics when the sw_axi_cnt_type assert to 1 by asserting the configured bit sw_axi_cnt_type_wrap to 1.

4.4.4 CPU address read and write monitor case

Case5：The user wants to monitor CPU of ID 127 read from or write to the address area (from 0x4000 to 0x8000) action of some scenario. Here are the steps:

- Finish CPU AXI PERF clear step, assert mon_id_msk 0
- Assert mon_id_type 1, mon_id 127, AXI_PERFE_RD_MON_ST AXI_PERFE_WR_MON_ST 0x4000, AXI_PERFE_RD_MON_END，AXI_PERFE_WR_MON_END 0x8000（AXI_PERFE_CON3=0xfffffe02）
- Start CPU AXI PERF
- Start CPU, finish this scenario and wait for the AXI PERF interrupt during this period
- If CPU get AXI PERF interrupt, then read back AXI_PERFE_INT_STATUS, get ar_mon_axi_hit_flag and aw_mon_axi_hit_flag, these bits can make sure whether ID 127 read from the specify address area or write to the specify address area
- If the AXI PERF interrupt is not asserted, means ID 127’s action does not hit the event
- Clear CPU AXI_PERF and wait for next statistic

Case6：The user wants to monitor CPU of ID0 - ID127 read from or write to the address area (from 0x14000 to 0x18000) action of some scenario. Here are the steps:

- Finish CPU AXI PERF clear step, assert mon_id_msk 0
- Assert mon_id_type 0, mon_id 0, AXI_PERFE_RD_MON_ST AXI_PERFE_WR_MON_ST 0x14000，AXI_PERFE_RD_MON_END, AXI_PERFE_WR_MON_END 0x18000（AXI_PERFE_CON3=0xffff01fc）
- Start CPU AXI PERF
- Start CPU, finish this scenario and wait for the AXI PERF interrupt during this period
- If CPU get AXI PERF interrupt, then read back AXI_PERFE_INT_STATUS, get ar_mon_axi_hit_flag and aw_mon_axi_hit_flag, these bits can make sure whether these IDs read from the specify address area or write to the specify address area. We can make sure which ID hit the event by aw_mon_axi_id_status or ar_mon_axi_id_status
- If the AXI PERF interrupt is not asserted, means no ID’s action hit the event
- Clear CPU AXI_PERF and wait for next statistic

Chapter 5 CPU_BOOST

5.1 Overview

The CPU_BOOST accelerates the CPU frequency when only one core working in the multi-core system.

The CPU_BOSOT supports following features:

- When only one core is alive (3 core are in WFI), CPU clock will turn to high freq config. The trigger condition is CRU_APLL configuration equal to CPU_BOOST_APLL_CON.
- VPLL0/VPLL1 can be use as freq borrow PLL, and also can be use as target PLL.
- When CPU_BOOST FSM is working, the WFI cores' wakeup interrupts(include FIQ & IRQ) will be blocked by CPU_BOOST, until the FSM back to idle.
- If the next boost activity condition is too near from last boost (the interval time is less than switch threshold). The boost condition will not trigger boost activity, until interval time is bigger than switch threshold.
- The boost activity can be counted. Including boost switch times and high freq counter.

5.2 Block Diagram

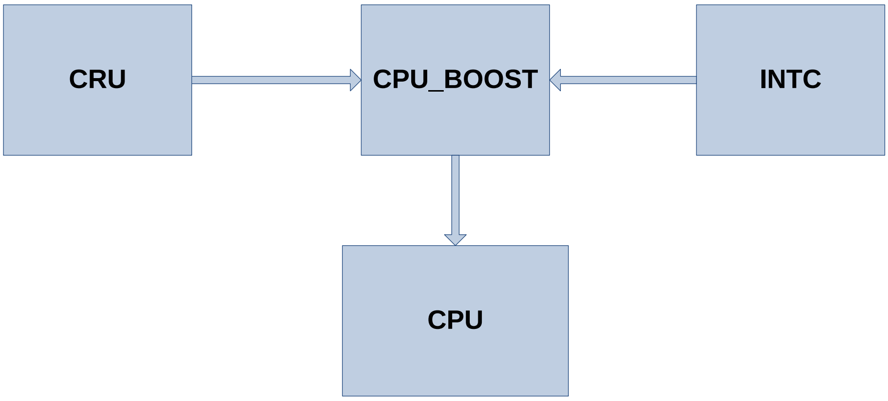


Fig. 5-1 CPU_BOOST block diagram

5.3 Register Description

5.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
BOOST_APLL_CON0	0x0000	W	0x0000215e	ARM PLL control register0
BOOST_APLL_CON1	0x0004	W	0x00001046	ARM PLL control register1
BOOST_CLK_CON	0x0008	W	0x00000000	CPU clock select and divide control register
BOOST_BOOST_CON	0x000c	W	0x00000000	CPU boost control register
BOOST_SWITCH_CNT	0x0010	W	0x00000000	Switch counter register
BOOST_HIGH_PERF_CNT0	0x0014	W	0x00000000	High performance time counter register0
BOOST_HIGH_PERF_CNT1	0x0018	W	0x00000000	High performance time counter register1
BOOST_STATIS_THRESHOLD	0x001c	W	0x00000100	Switch counter threshold for statistics
BOOST_SHORT_SWITCH_CNT	0x0020	W	0x00000000	Short switch counter register

Name	Offset	Size	Reset Value	Description
BOOST_SWTICH_THRESHOLD	0x0024	W	0x00000100	Frequency change switch threshold
BOOST_FSM_STATUS	0x0028	W	0x00000000	FSM status register
BOOST_APLL_LOW_CON0	0x002c	W	0x0000415e	ARM PLL low freq control register0
BOOST_APLL_LOW_CON1	0x0030	W	0x00001046	ARM PLL low freq control register1

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

5.3.2 Detail Register Description

BOOST_APLL_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask 16 bit write mask for lsb 15-0
15	RO	0x0	reserved
14:12	RW	0x2	postdiv1 PLL factor postdiv1
11:0	RW	0x15e	fbdiv PLL factor fbdiv

BOOST_APLL_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask 16 bit write mask for lsb 15-0
15:13	RO	0x0	reserved
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11:9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

BOOST_CLK_CON

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:13	RO	0x0	reserved
12	RW	0x0	backup_pll_usage_sel 1'b0: select backup PLL for freq borrow 1'b1: select backup PLL for target PLL
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	cpu_pll_sel 2'b00: APLL 2'b01: VPLL0 2'b10: VPLL1
7:5	RO	0x0	reserved
4:0	RW	0x00	core_div_con clk=clk_src/(div_con+1)

BOOST_BOOST_CON
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask 16 bit write mask for lsb 15-0
15:13	RO	0x0	reserved
12	W1C	0x0	clear_all Clear all statistics configuration, including HIGH_PERF_CNT0/1, SWITCH_CNT, FAST_SWITCH_CNT. 1'b0: normal 1'b1: clear all statistics configuration
11:5	RO	0x0	reserved
4	RW	0x0	statis_enalbe 1'b0: disable 1'b1: enable
3	RW	0x0	apll_low_freq_en Control apll to low frequency when boost_enable is valid: 1'b0: disable 1'b1: enable
2	RW	0x0	apll_sw_ctrl 1'b0: disable 1'b1: software control cpu_pll_sel, core_div_con, apll_slowmode, apll_pd
1	RW	0x0	boost_recovery 1'b0: normal 1'b1: recovery
0	RW	0x0	boost_enalbe 1'b0: disable 1'b1: enable

BOOST_SWITCH_CNT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	switch_cnt This 32bit counter is for accumulating switch activity

BOOST_HIGH_PERF_CNT0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	high_perf_cnt This 32bit counter [31:0] in 64bit high performance time counter, which is for accumulating CPU run in high frequency

BOOST_HIGH_PERF_CNT1

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	high_perf_cnt This 32bit counter [63:32] in 64bit high performance time counter, which is for accumulating CPU run in high frequency

BOOST_STATIS_THRESHOLD

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000100	statis_threshold If the high frequency sustain time is less than threshold, SHORT_SWITCH_CNT will count. This threshold is based on 24M clock cycle

BOOST_SHORT_SWITCH_CNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	short_swtich_cnt If the high frequency sustain time is less than threshold, SHORT_SWITCH_CNT will count

BOOST_SWTICH_THRESHOLD

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000100	switch_threshold If switch activity interval time is less than threshold, freq will not change

BOOST_FSM_STATUS

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	idle_state 1'b0: normal 1'b1: idle

Bit	Attr	Reset Value	Description
7:4	RO	0x0	req_fsm 4'b0000: initial state. others state is busy state
3:0	RO	0x0	boost_fsm 4'b0000: initial state. others state is busy state

BOOST_APLL_LOW_CON0

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	bit_write_mask 16 bit write mask for lsb 15-0
15	RO	0x0	reserved
14:12	RW	0x4	postdiv1 PLL factor postdiv1
11:0	RW	0x15e	fbdiv PLL factor fbdiv

BOOST_APLL_LOW_CON1

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	bit_write_mask 16 bit write mask for lsb 15-0
15:13	RO	0x0	reserved
12	RW	0x1	dsmpd when 1, PLL work at interger mode when 0, PLL work at frac mode
11:9	RO	0x0	reserved
8:6	RW	0x1	postdiv2 PLL factor postdiv2
5:0	RW	0x06	refdiv PLL factor refdiv

5.4 Application Notes

The CPU_BOOST config flow is as follow.

Before changing frequency:

a. Config BOOST_APLL_CON0/1 and BOOST_APLL_LOW_CON0/1 for single core boost frequency and multi-core normal frequency.
b. Config BOOST_CLK_CON for choosing VPLL0 usage and VPLL0 div_con.
c. Config BOOST_BOOST_CON[0] for enable boost function.

Frequency change flow:

a. Config BOOST_BOOST_CON[1] for enable recovery, and wait for the CPU_BOOST back to idle state by reading BOOST_FSM_STATUS[8].
b. Config BOOST_BOOST_CON[3:2] for enable sw_ctrl and low_freq_en.
c. Config BOOST_CLK_CON for choosing VPLL0 usage and VPLL0 div_con.
d. Config CRU_CLKSEL0_CON for changing CPU PLL source to VPLL0.
e. Config CRU_MODE_CON[1:0] for changing APLL to slowmode.

f. Config CRU_APLL_CON0/1 for re-config APLL and power_down APLL.
g. Config BOOST_BOOST_CON[3] for disable low_freq_en.
h. Config CRU_APLL_CON1 for deassert APLL power_down.
i. Waiting for APLL lock.
j. Config CRU_MODE_CON [1:0] for changing APLL to normalmode.
k. Config CRU_CLKSEL0_CON for changing CPU PLL source back to APLL.
l. Config BOOST_BOOST_CON[1] for disable recovery.
m. Config BOOST_BOOST_CON[2] for disable sw_ctrl.

Chapter 6 Audio Subsystem

6.1 Overview

Audio Subsystem is embedded with rich audio interfaces such as I2S, PCM, TDM, PDM, SPDIF and so on. It's also support a VAD(Voice Activity Detection) for human voice detection which can respond to human voice request timely and fast setup intelligent voice interaction application.
Audio Subsystem supports the following features:

- I2S with 2 channel
  - Support 2 I2S_2CH components
  - I2S_2CH_0 support master tx/rx mode and slave tx/rx mode
  - I2S_2CH_0 is connected to chip IO
  - I2S_2CH_1 support slave rx mode
  - I2S_2CH_1 is connected with Audio Codec inside chip
  - Sample rate up to 192KHz
  - Support common LRCK signal for receiving and transmitting when the sample rate are same
- I2S with 8 channel
  - Support 4 I2S_8CH components
  - I2S_8CH_0 support master tx/rx mode and slave tx/rx mode
  - I2S_8CH_1 support master tx/rx mode and slave tx/rx mode
  - I2S_8CH_0/1 are connected to chip IO
  - I2S_8CH_0 support max 8ch in and max 8ch out simultaneously
  - I2S_8CH_1 support tx plus rx max 10ch simultaneously
  - I2S_8CH_2 support master tx/rx mode and slave tx/rx mode
  - I2S_8CH_3 support slave rx mode, can only works as 4CH mode
  - I2S_8CH_2/3 are connected with Audio Codec inside chip
  - Sample rate up to 192KHz
  - Support independent SCLK and LRCK signals for receiving and transmitting
  - Support common SCLK and LRCK signal for receiving and transmitting when the sample rate are same
- I2S with 16 channel
  - Support one I2S_16CH by gathering I2S_8CH_0 and I2S_8CH_1
  - Support master tx/rx mode and slave tx/rx mode
  - Sample rate up to 192KHz
- PDM with 8 channel
  - Sample rate up to 192KHz
- TDM with 8 channel
  - Support 4 TDM_8CH, share same I2S_8CH controller accordingly
  - Sample rate up to 192KHz@2CH and 48KHz@8CH
  - Support independent SCLK and LRCK signals for receiving and transmitting
  - Support common SCLK and LRCK signal for receiving and transmitting when the sample rate are same
- SPDIF
  - Support SPDIF TX x 1
  - Support SPDIF RX x 1
  - Support HDMI ARC
  - Sample rate up to 192KHz
  - Support SPDIF RX is bypassed to SPDIF TX directly
- Voice Activity Detection(VAD)
- Embedded Audio Codec

6.2 Block Diagram

N/A

6.3 Function Description

6.3.1 I2S_2CH_0 Application Topology

I2S_2CH_0 support master transmitting/receiving mode and slave transmitting/receiving mode. IO LRCK_TX is shared for transmitting and receiving modes, user should configure the I2S0_2CH0 register CKR[29:28] to 0x1. If user wants to use both transmitting and receiving in master mode, user should enable transmitting and receiving at the same time. The sample rate should be same when transmitting and receiving at the same time. The direction of IO MCLK is controlled by CRU. The direction of IO SCLK and IO LRCK_TX is controlled by I2S mode, it is output when I2S is master mode, and input when I2S is slave mode.

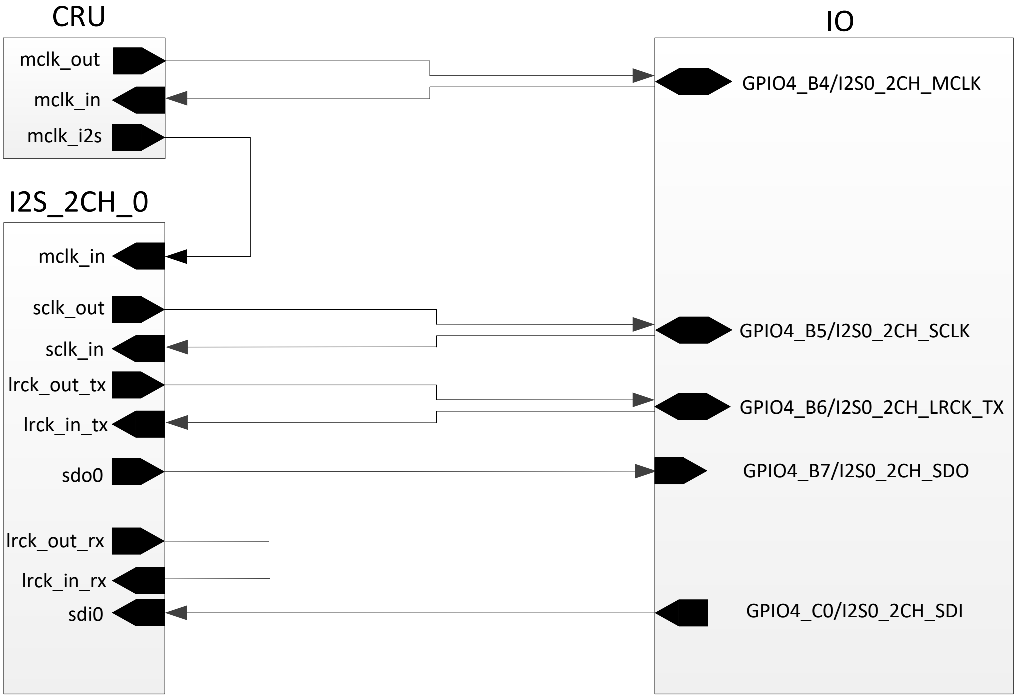


Fig. 6-1 I2S_2CH_0 application topology

6.3.2 I2S_2CH_1 Application Topology

I2S_2CH_1 only support slave receiving mode. The input data of I2S can select from any one of Audio Codec data outputs as follow figure shows, it is configured by GRF_SOC_CON1[13:12]. pin_adc_mclk source from mclk_i2s_rx of I2S_8H_2.

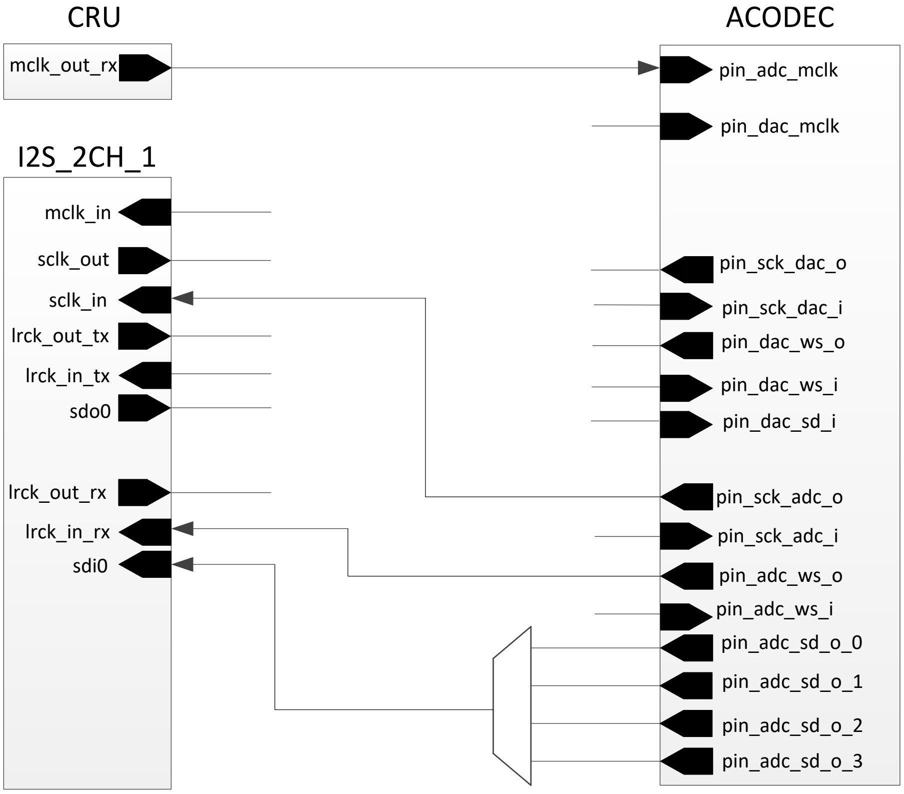


Fig. 6-2 I2S_8CH_1 application topology

6.3.3 I2S_8CH_0 Application Topology

I2S_8CH_0 support master transmitting/receiving mode and slave transmitting/receiving mode. It supports maximum 8 channels transmitting and 8 channels receiving simultaneously.

IO LRCK_TX/LRCK_RX can be shared for transmitting and receiving modes. User can configure the I2S_8CH_0 register CKR[29:28] to select LRCK_TX or LRCK_RX as common. The sharing will reduce the IO usage. After the sharing configuration, if user wants to use both transmitting and receiving in master mode, user should enable transmitting and receiving at the same time.

IO SCLK_TX/SCLK_RX can be shared for transmitting and receiving modes. User can configure the GRF_SOC_CON2[9:8] to select SCLK_TX or SCLK_RX as common. The sharing will reduce the IO usage. The LRCK_TX/LRCK_RX must be shared when SCLK_TX/SCLK_RX is shared, the reverse is the same.

The direction of IO SCLK and IO LRCK_TX is controlled by I2S mode, it is output when I2S is master mode, and input when I2S is slave mode. The source selection of I2S1_MCLK output is controlled by GRF_SOC_CON2[10].

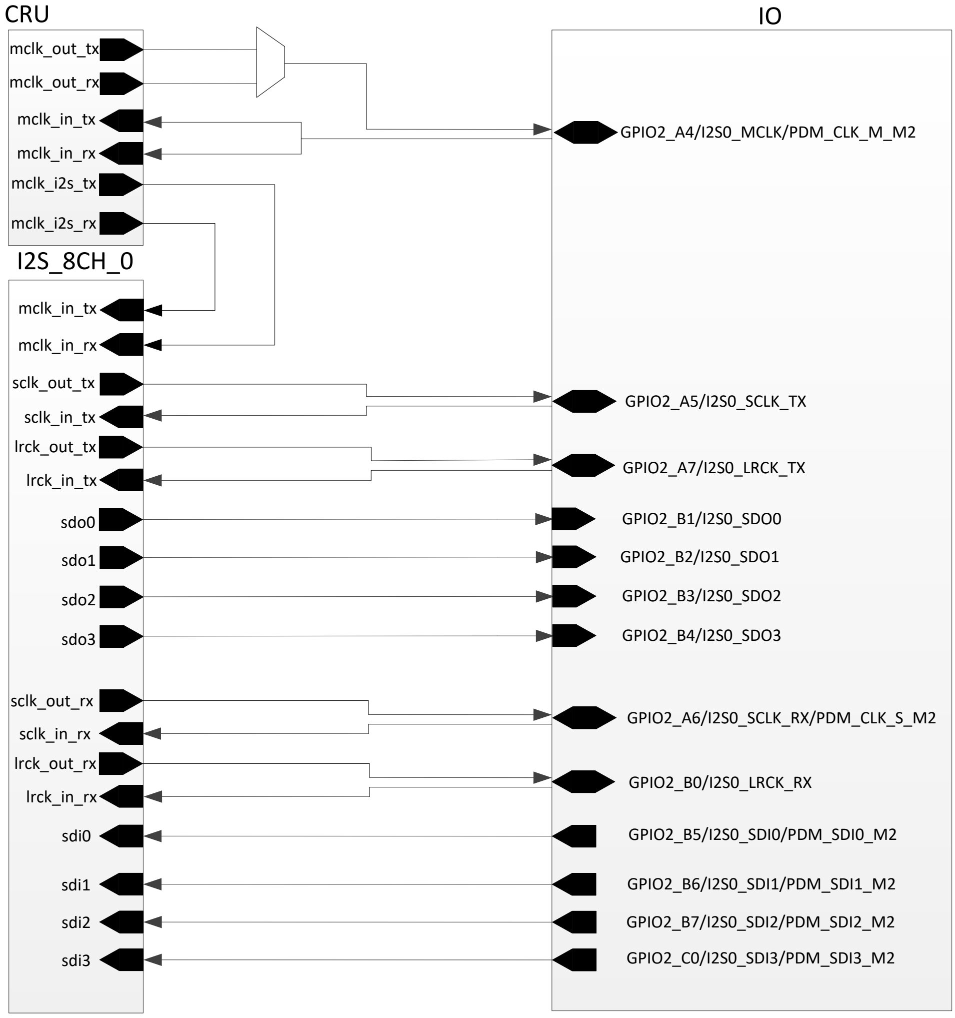


Fig. 6-3 I2S_8CH_0 application topology

6.3.4 I2S_8CH_1 Application Topology

I2S_2CH_1 support master transmitting/receiving mode and slave transmitting/receiving mode. It supports maximum 10 channels (transmitting + receiving) simultaneously, eg 8 channels transmitting and 2 channels receiving.

I2S_2CH_1 support two group IOMUX scheme which is controlled by GRF_SOC_CON2[3].

IO LRCK_TX/LRCK_RX can be shared for transmitting and receiving modes. User can configure the I2S_8CH_1 register CKR[29:28] to select LRCK_TX or LRCK_RX as common. The sharing will reduce the IO usage. After the sharing configuration, if user wants to use both transmitting and receiving in master mode, user should enable transmitting and receiving at the same time.

IO SCLK_TX/SCLK_RX can be shared for transmitting and receiving modes. User can configure the GRF_SOC_CON2[1:0] to select SCLK_TX or SCLK_RX as common. The sharing will reduce the IO usage. The LRCK_TX/LRCK_RX must be shared when SCLK_TX/SCLK_RX is shared, the reverse is the same.

The direction of IO SCLK and IO LRCK_TX is controlled by I2S mode, it is output when I2S is master mode, and input when I2S is slave mode.

The direction of I2S1_SDO1_SDI3_M0/1, I2S1_SDO2_SDI2_M0/1, I2S1_SDO3_SDI1_M0/1 is controlled by GRF_SOC_CON2[7:5]. The source selection of I2S1_MCLK output is controlled by GRF_SOC_CON2[2].

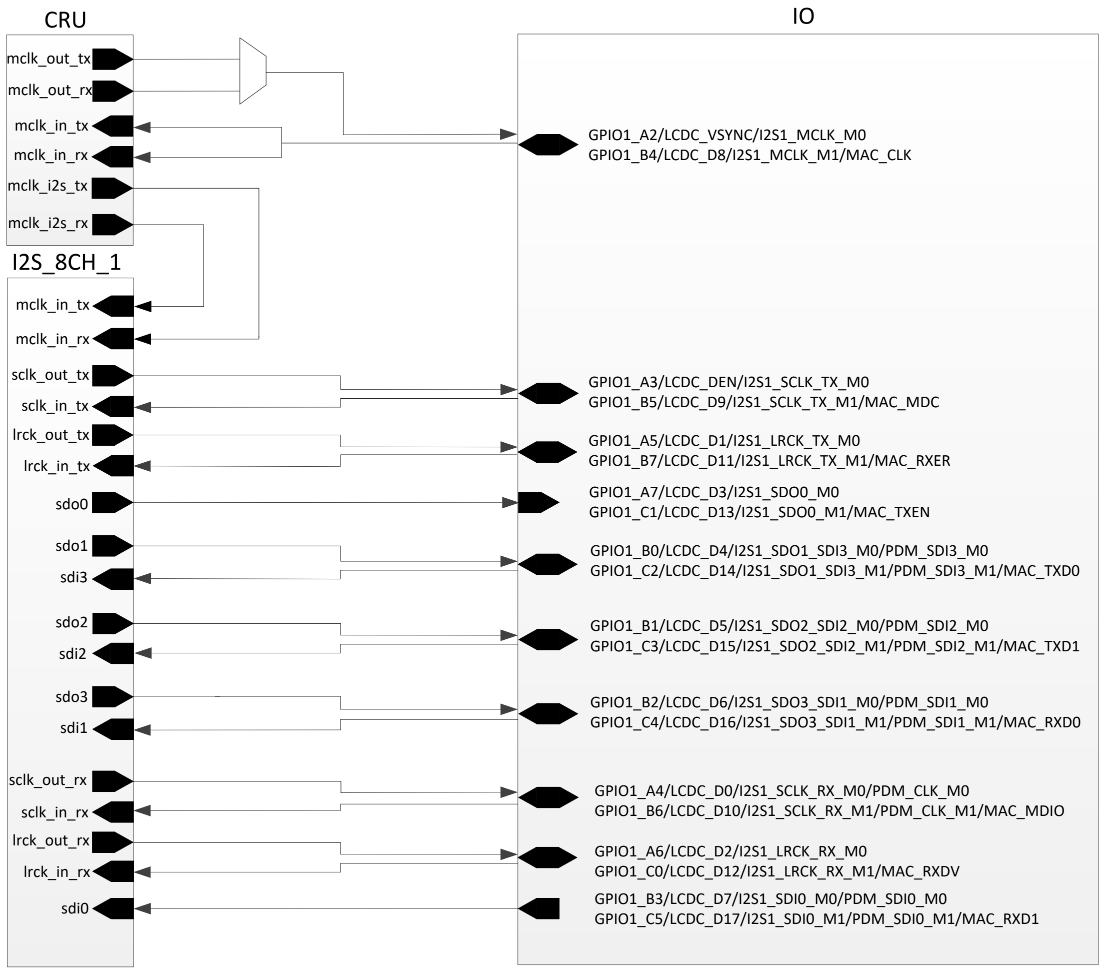

CRU
IO
mclk_out_tx
mclk_out_rx
mclk_in_tx
mclk_in_rx
mclk_i2s_tx
mclk_i2s_rx
I2S_8CH_1
mclk_in_tx
mclk_in_rx
sclk_out_tx
sclk_in_tx
lrck_out_tx
lrck_in_tx
sdo0
sdo1
sdi3
sdo2
sdi2
sdo3
sdi1
sclk_out_rx
sclk_in_rx
lrck_out_rx
lrck_in_rx
sdi0
GPIO1_A2/LCDC_VSYNC/I2S1_MCLK_M0
GPIO1_B4/LCDC_D8/I2S1_MCLK_M1/MAC_CLK
GPIO1_A3/LCDC_DEN/I2S1_SCLK_TX_M0
GPIO1_B5/LCDC_D9/I2S1_SCLK_TX_M1/MAC_MDC
GPIO1_A5/LCDC_D1/I2S1_LRCK_TX_M0
GPIO1_B7/LCDC_D11/I2S1_LRCK_TX_M1/MAC_RXER
GPIO1_A7/LCDC_D3/I2S1_SDO0_M0
GPIO1_C1/LCDC_D13/I2S1_SDO0_M1/MAC_TXEN
GPIO1_B0/LCDC_D4/I2S1_SDO1_SDI3_M0/PDM_SDI3_M0
GPIO1_C2/LCDC_D14/I2S1_SDO1_SDI3_M1/PDM_SDI3_M1/MAC_TXD0
GPIO1_B1/LCDC_D5/I2S1_SDO2_SDI2_M0/PDM_SDI2_M0
GPIO1_C3/LCDC_D15/I2S1_SDO2_SDI2_M1/PDM_SDI2_M1/MAC_TXD1
GPIO1_B2/LCDC_D6/I2S1_SDO3_SDI1_M0/PDM_SDI1_M0
GPIO1_C4/LCDC_D16/I2S1_SDO3_SDI1_M1/PDM_SDI1_M1/MAC_RXD0
GPIO1_A4/LCDC_D0/I2S1_SCLK_RX_M0/PDM_CLK_M0
GPIO1_B6/LCDC_D10/I2S1_SCLK_RX_M1/PDM_CLK_M1/MAC_MDIO
GPIO1_A6/LCDC_D2/I2S1_LRCK_RX_M0
GPIO1_C0/LCDC_D12/I2S1_LRCK_RX_M1/MAC_RXDV
GPIO1_B3/LCDC_D7/I2S1_SDI0_M0/PDM_SDI0_M0
GPIO1_C5/LCDC_D17/I2S1_SDI0_M1/PDM_SDI0_M1/MAC_RXD1

Fig. 6-4 I2S_8CH_1 application topology

6.3.5 I2S_16CH Application Topology

Support one I2S_16CH by gathering I2S_8CH_0 and I2S_8CH_1.
When the I2S_16CH is slave mode, the I2S_8CH_0 and I2S_8CH_1 are both slave mode, they can transmit or receive respective to gather.
When the I2S_16CH is master mode, one of I2S_8CH_0 and I2S_8CH_1 should be master mode and the other should be slave mode. In this case, user should enable GRF_SOC_CON2[14]. The figure below shows the topology when I2S_8CH_0 is master mode and I2S_8CH_1 is slave mode, the signal sclk/lrck_gather_tx/rx which derived by I2S_8CH_0 are input to I2S_8CH_1. User can also set I2S_8CH_1 to master mode and I2S_8CH_0 to slave mode, in this case, the signal sclk/lrck_gather_tx/rx which derived by I2S_8CH_1 are input to I2S_8CH_0.

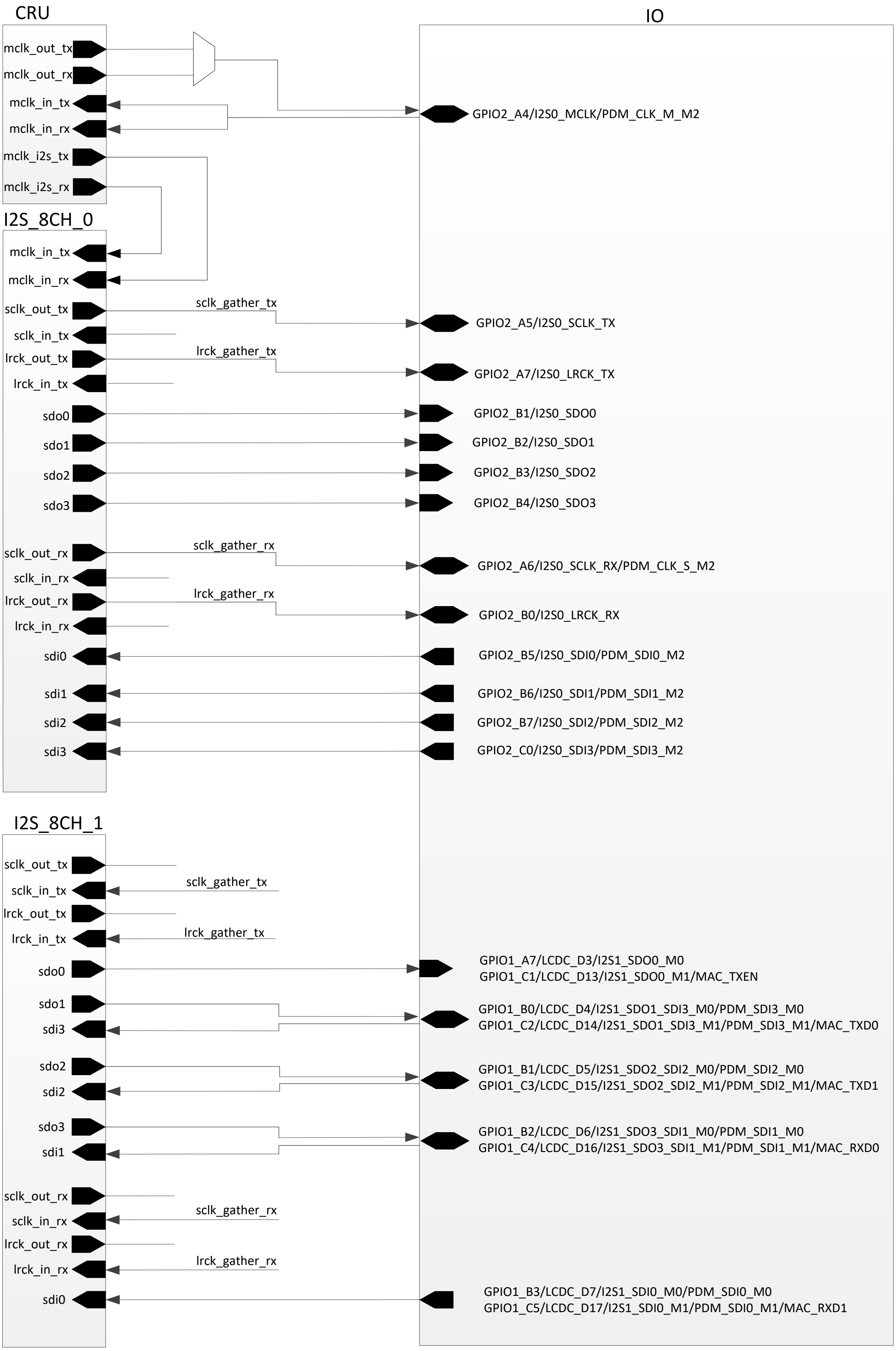

CRU
IO
mclk_out_tx
mclk_out_rx
mclk_in_tx
mclk_in_rx
mclk_i2s_tx
mclk_i2s_rx
GPIO2_A4/I2S0_MCLK/PDM_CLK_M_M2
I2S_8CH_0
mclk_in_tx
mclk_in_rx
sclk_out_tx
sclk_in_tx
lrck_out_tx
lrck_in_tx
sclk_gather_tx
lrck_gather_tx
GPIO2_A5/I2S0_SCLK_TX
GPIO2_A7/I2S0_LRCK_TX
sdo0
sdo1
sdo2
sdo3
GPIO2_B1/I2S0_SDO0
GPIO2_B2/I2S0_SDO1
GPIO2_B3/I2S0_SDO2
GPIO2_B4/I2S0_SDO3
sclk_out_rx
sclk_in_rx
lrck_out_rx
lrck_in_rx
sclk_gather_rx
lrck_gather_rx
GPIO2_A6/I2S0_SCLK_RX/PDM_CLK_S_M2
GPIO2_B0/I2S0_LRCK_RX
sdi0
sdi1
sdi2
sdi3
GPIO2_B5/I2S0_SDI0/PDM_SDI0_M2
GPIO2_B6/I2S0_SDI1/PDM_SDI1_M2
GPIO2_B7/I2S0_SDI2/PDM_SDI2_M2
GPIO2_C0/I2S0_SDI3/PDM_SDI3_M2
I2S_8CH_1
sclk_out_tx
sclk_in_tx
lrck_out_tx
lrck_in_tx
sclk_gather_tx
lrck_gather_tx
sdo0
GPIO1_A7/LCDC_D3/I2S1_SDO0_M0
GPIO1_C1/LCDC_D13/I2S1_SDO0_M1/MAC_TXEN
sdo1
sdi3
GPIO1_B0/LCDC_D4/I2S1_SDO1_SDI3_M0/PDM_SDI3_M0
GPIO1_C2/LCDC_D14/I2S1_SDO1_SDI3_M1/PDM_SDI3_M1/MAC_TXD0
sdo2
sdi2
GPIO1_B1/LCDC_D5/I2S1_SDO2_SDI2_M0/PDM_SDI2_M0
GPIO1_C3/LCDC_D15/I2S1_SDO2_SDI2_M1/PDM_SDI2_M1/MAC_TXD1
sdo3
sdi1
GPIO1_B2/LCDC_D6/I2S1_SDO3_SDI1_M0/PDM_SDI1_M0
GPIO1_C4/LCDC_D16/I2S1_SDO3_SDI1_M1/PDM_SDI1_M1/MAC_RXD0
sclk_out_rx
sclk_in_rx
lrck_out_rx
lrck_in_rx
sclk_gather_rx
lrck_gather_rx
sdi0
GPIO1_B3/LCDC_D7/I2S1_SDI0_M0/PDM_SDI0_M0
GPIO1_C5/LCDC_D17/I2S1_SDI0_M1/PDM_SDI0_M1/MAC_RXD1

Fig. 6-5 I2S_16CH application topology

6.3.6 I2S_8CH_2 Application topology

I2S_8CH_2 support master transmitting/receiving mode and slave transmitting/receiving mode. For each input data of I2S, it can select any one of Audio Codec output data, it is controlled by GRF_SOC_CON1[7:0]. pin_adc_mclk source from mclk_i2s_rx of I2S_8H_2, and pin_dac_mclk source from mclk_i2s_tx of I2S_8H_2.

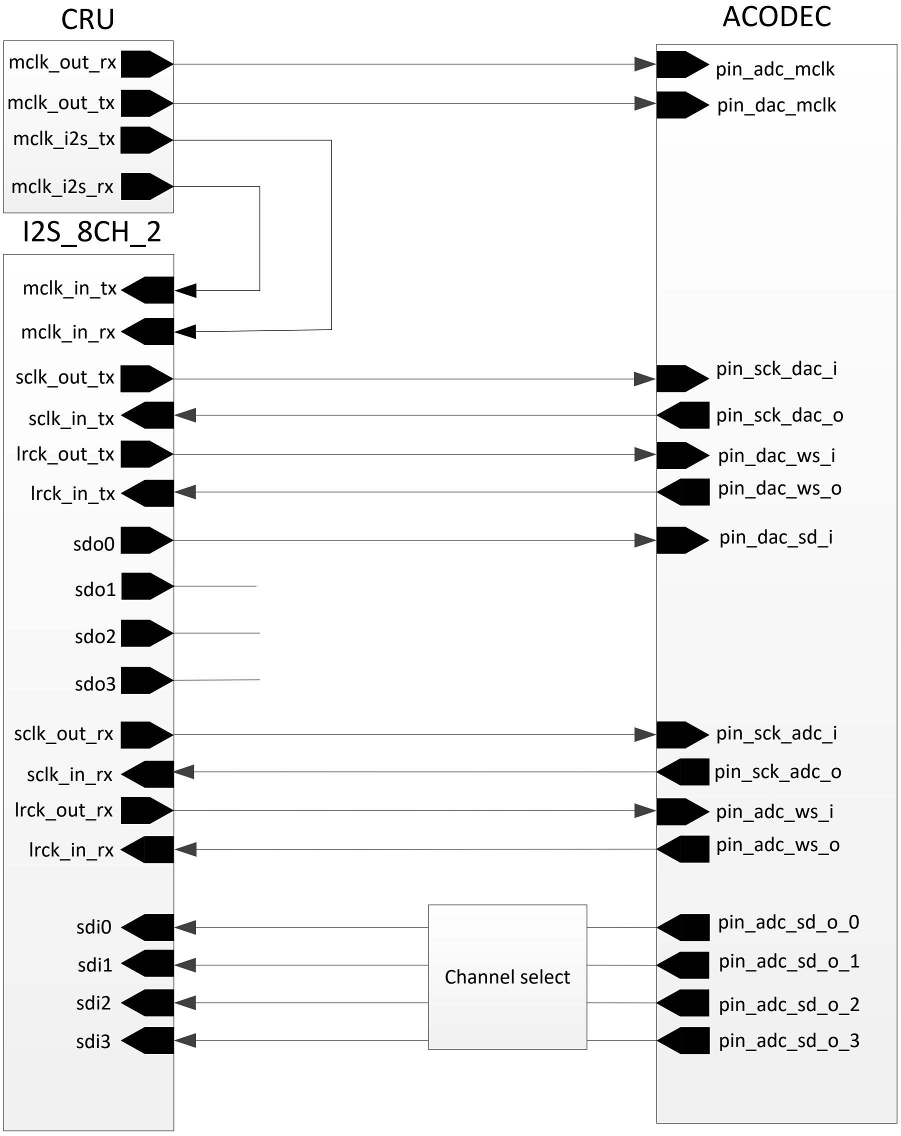


Fig. 6-6 I2S_8CH_2 application topology

6.3.7 I2S_8CH_3 Application Topology

I2S_8CH_3 only support slave receiving mode, and it can only works as 4 channels. For each valid input data of I2S, it can select any one of Audio Codec output data, it is controlled by GRF_SOC_CON1[11:8].

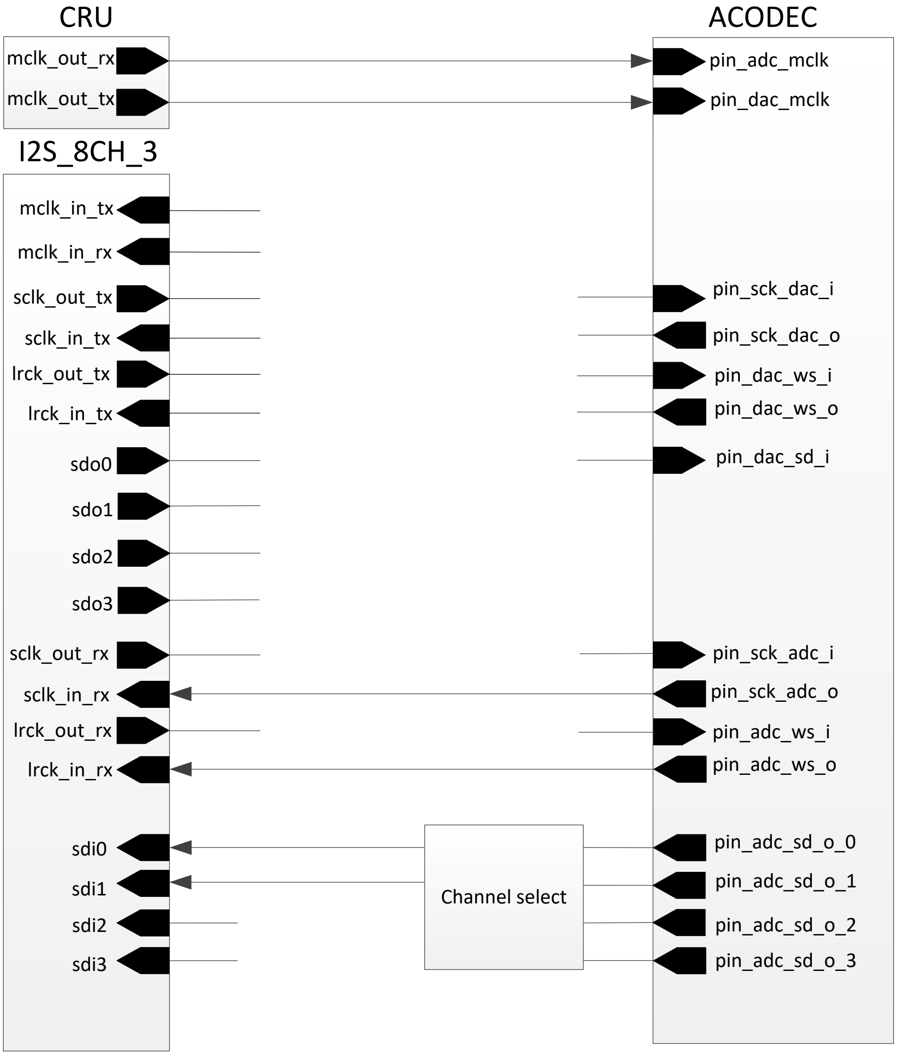


Fig. 6-7 I2S_8CH_3 application topology

6.3.8 VAD Application Topology

VAD support read voice data from I2S_8CH_0~3/PDM, and then use the data to detect the amplitude of voice. As the figure following shows, the data path is ①->②->③ when SOC work at normal mode. When SOC enable the VAD and enter low power mode, the data path is ④. User can configure to work at 1 channel when in low power mode. After the voice detection event, VAD can wake up SOC and configure to work at multichannel at the same time. When SOC return to normal mode, it can disable the VAD and return to normal data path, and it can read the backtracking data from the Internal SRAM.

When DMA is enabled, the share address range of Internal SRAM can be accessed by VAD only.

Considering the power consumption of low power mode, user can reduce the frequency of PLL. User also can power down all PLL and select the xin_osc0 as the clock source of I2S_8CH/PDM.

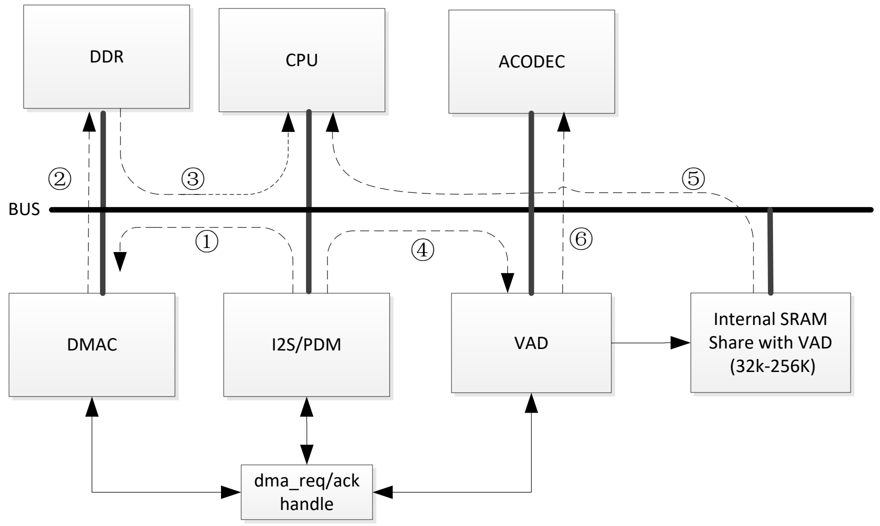


Fig. 6-8 I2S_2CH_3 application topology

In normal mode, the DMA request of I2S_8CH_0~3/PDM is handled by system DMA controller. When VAD is enabled, the DMA request is handled by VAD. The topology of I2S_8CH_0 DMA request is shows as following figure, I2S_8CH_1~3 and PDM are same.

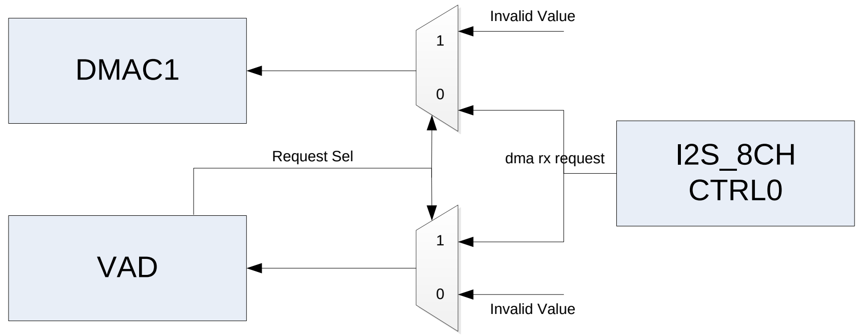


Fig. 6-9 I2S_8CH_0 DMA request and acknowledge

6.4 Register Description

6.4.1 Registers Summary

N/A

6.4.2 Detail Register Description

N/A

6.5 Application Notes

N/A

Chapter 7 Audio Serial Port Controller (ASPC)

7.1 Overview

The Audio Serial Port Controller (ASPC) is a PDM interface controller and decoder that support mono PDM format. It integrates a clock generator driving the PDM microphone and embeds filters which decimate the incoming bit stream to obtain most common audio rates. ASPC supports the following features:

- Support one internal 32-bit wide and 128-location deep FIFOs for receiving audio data
- Support receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of receive FIFO full interrupt
- Support combined interrupt output
- Support AHB bus slave interface
- Support DMA handshaking interface and configurable DMA water level
- Support PDM master receive mode
- Support 4 paths. Each path is composed of two digital microphone channels, the ASPC can be used with four stereo or eight mono microphones. Each path is enabled or disabled independently
- Support 16 ~24 bit sample resolution
- Support sample rate: 8khz,16khz,32kHz,64kHz,128khz,11.025khz,22.05khz,44.1khz,88.2khz,176.4khz,12khz ,24khz,48khz,96khz,192khz
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support programmable data sampling sensibility (rising or falling edge)

7.2 Block Diagram

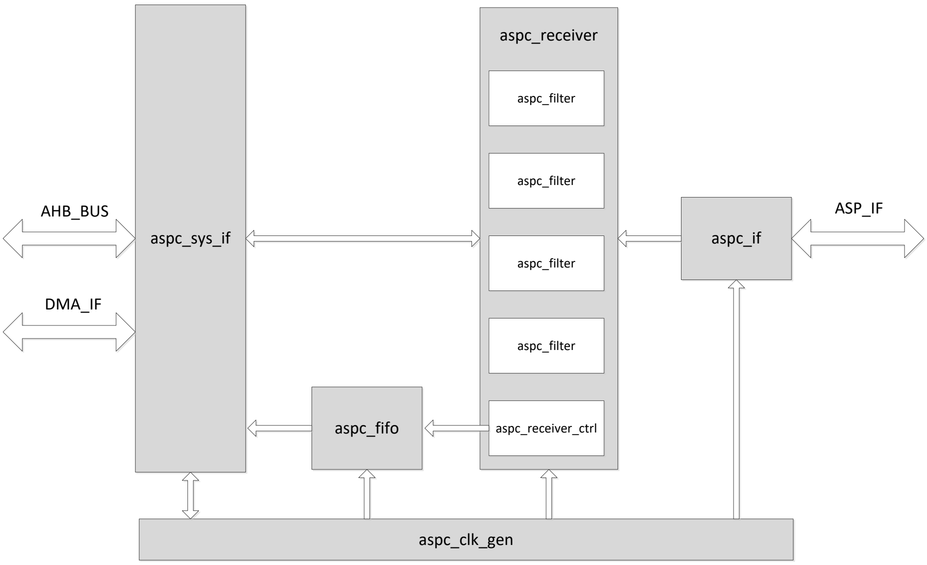


Fig. 7-1 ASPC Block Diagram

System Interface

The system interface implements the APB slave operation. It contains not only control registers of receiver inside but also interrupt and DMA handshaking interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK, and by the divider of the module, the clock generator generates CLK_PDM to receiver.

Receiver

The receiver can act as a decimation filter of PDM. And export PCM format data.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x

128.

ASP interface

The ASP interface implements PDM bit streams receive operation.

7.3 Function Description

7.3.1 AHB Interface

There is an AHB slave interface in ASPC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in 7.4.1.

7.3.2 PDM Interface

The PDM interface is a 5-wire interface. The ASPC module can support up to four external stereo and eight digital microphones.

Follow two figure show two cases of use of the ASPC, but all configurations are possible with stereo and mono digital microphones.

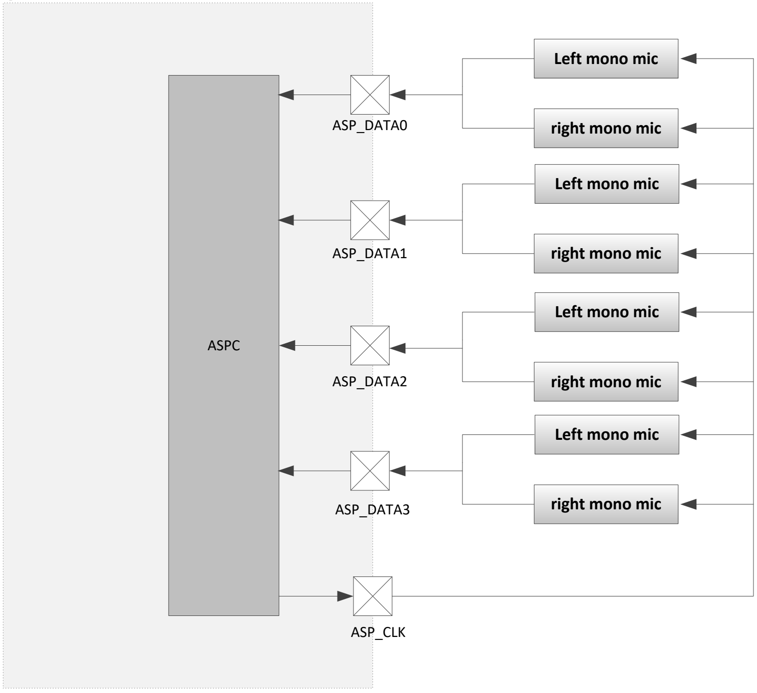


Fig. 7-2 ASPC with Eight Mono MIC

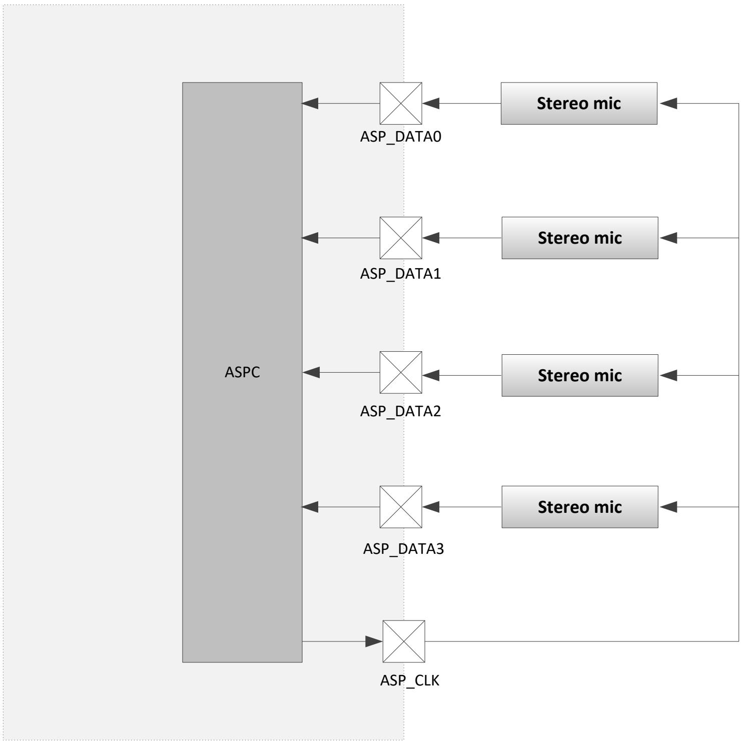


Fig. 7-3 ASPC with Four Stereo MIC

The PDM interface consists of a serial-data shift clock output (ASP_CLK) and a serial data input (ASP_DATA). The clock is fanned out to both digital mics, and both digital mics' data (left channel and right channel) outputs share a single signal line. To share a single line, the digital mics tristate their output during one phase of the clock(high or low part of cycle, depending on how they are configured via their L/R input).

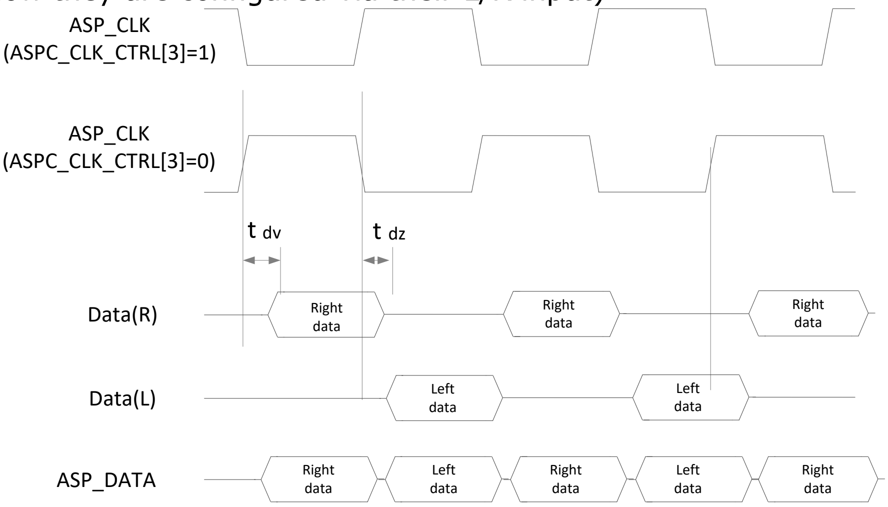


Fig. 7-4 ASPC interface diagram with external MIC

7.3.3 Digital Filter

The external PDMIC generates a PDM stream of bits and transfers it in one period or one half-period of the clock provided by the ASPC. The aim of the ASPC is to process data from the PDM interface, decimate and filter the data, and store the processed data in the FIFO. The four paths are identical. Each path is composed of a left and a right channel. The PDM interface delivers eight parallel data of 1bit. Each bit goes to a filter. The aim of the filter is to limit the noise and export PCM format audio data.

7.3.4 Frequency Configuration

MCLK is the source clock signal. ASP_CLK is the output clocks generated in the ASPC and is fed to the external microphones. They are also the internal clock of the external microphones. User must take care about the value of ASP_CLK when selecting the source clock (MCLK).

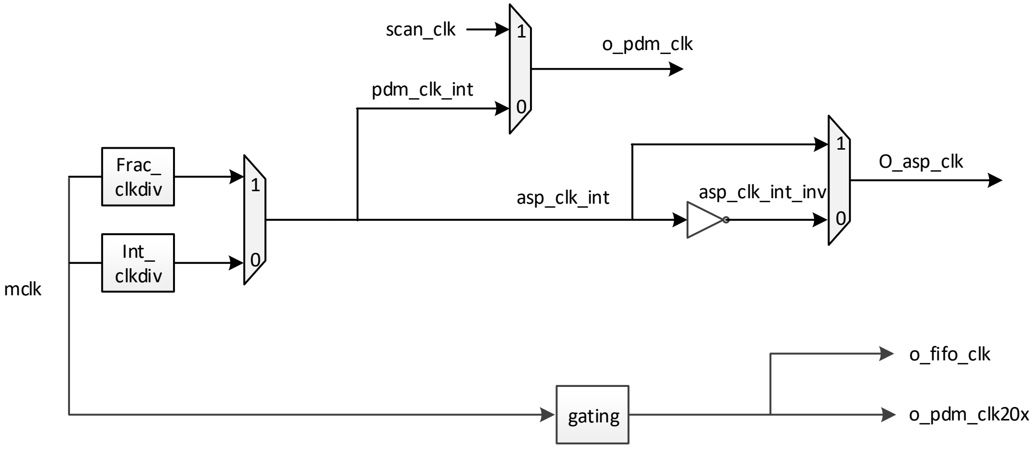


Fig. 7-5 ASPC Clock Structure

Table 7-1 Relation between ASP_CLK and sample rate

ASP_CLK	Sample rate
3.072Mhz	12khz,24khz,48khz,96khz,192khz
2.8224Mhz	11.025khz,22.05khz,44.1khz,88.2khz,176.4khz
2.048Mhz	8khz,16khz,32kHz,64kHz,128khz

User must configure the div_con depended on the frequency of MCLK. If MCLK/ASP_CLK is more than 40, ASPC_CLK_CTRL[6] should set to 0; if MCLK/ASP_CLK is less than 35, ASPC_CLK_CTRL[6] should set to 1.

7.4 Register Description

7.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
ASPC_SYSCONFIG	0x0000	W	0x00000000	ASPC system configure register
ASPC_CTRL0	0x0004	W	0x78000017	ASPC control register 0
ASPC_CTRL1	0x0008	W	0x0bb8ea60	ASPC control register 1
ASPC_CLK_CTRL	0x000c	W	0x00000000	ASPC clock control register
ASPC_HPF_CTRL	0x0010	W	0x00000000	ASPC high pass filter control register
ASPC_FIFO_CTRL	0x0014	W	0x00000000	ASPC FIFO control register
ASPC_DMA_CTRL	0x0018	W	0x0000001f	ASPC DMA control register
ASPC_INT_EN	0x001c	W	0x00000000	ASPC interrupt enable register
ASPC_INT_CLR	0x0020	W	0x00000000	ASPC interrupt clear register
ASPC_INT_ST	0x0024	W	0x00000000	ASPC interrupt status register
ASPC_RXFIFO_DATA_REG	0x0030	W	0x00000000	ASPC receive FIFO data register
ASPC_DATA0R_REG	0x0034	W	0x00000000	ASPC path0 right channel data register
ASPC_DATA0L_REG	0x0038	W	0x00000000	ASPC path0 left channel data register
ASPC_DATA1R_REG	0x003c	W	0x00000000	ASPC path1 right channel data register
ASPC_DATA1L_REG	0x0040	W	0x00000000	ASPC path1 left channel data register

Name	Offset	Size	Reset Value	Description
ASPC_DATA2R_REG	0x0044	W	0x00000000	ASPC path2 right channel data register
ASPC_DATA2L_REG	0x0048	W	0x00000000	ASPC path2 left channel data register
ASPC_DATA3R_REG	0x004c	W	0x00000000	ASPC path3 right channel data register
ASPC_DATA3L_REG	0x0050	W	0x00000000	ASPC path3 left channel data register
ASPC_DATA_VALID	0x0054	W	0x00000000	Path data valid register
ASPC_VERSION	0x0058	W	0x59313030	ASPC version register
ASPC_INCR_RXDR	0x0400	W	0x00000000	Increment address receive FIFO data register

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

7.4.2 Detail Register Description

ASPC_SYSCONFIG

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	rx_start RX Transfer start bit 1’b0:stop RX transfer 1’b1:start RX transfer
1	RO	0x0	reserved
0	RW	0x0	rx_clr ASPC RX logic clear This is a self cleard bit. High active Write 0x1: clear RX logic Write 0x0: no action Read 0x1: clear ongoing Read 0x0: clear done

ASPC_CTRL0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RW	0x0	sjm_sel Store justified mode: (Can be written only when SYSCONFIG[2] is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 1.Because if HWT is 0, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 1’b0:right justified 1’b1:left justified

Bit	Attr	Reset Value	Description
30	RW	0x1	path3_en Path 3 enable 1'b1: enable 1'b0: disable
29	RW	0x1	path2_en Path 2 enable 1'b1: enable 1'b0: disable
28	RW	0x1	path1_en Path 1 enable 1'b1: enable 1'b0: disable
27	RW	0x1	path0_en Path 0 enable 1'b1: enable 1'b0: disable
26	RW	0x0	hwt_en HWT Halfword word transform Only valid when VDW select 16bit data 1’b0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel 1’b1: low 16bit data valid to AHB/APB bus, high 16 bit data invalid
25:16	RO	0x0	reserved
15:8	RW	0x0	int_div_con integer divider can be written only when SYSCONFIG[2] is 0
7:5	RO	0x0	reserved
4:0	RW	0x17	data_vld_width (Can be written only when SYSCONFIG[2] is 0.) Valid Data width 0~14: reserved 15: 16bit 16: 17bit 17: 18bit 18: 19bit …… n: (n+1)bit …… 23: 24bit

ASPC_CTRL1
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0bb8	frac_div_numerator fraction divider numerator (Can be written only when SYSCONFIG[2] is 0.)
15:0	RW	0xea60	frac_div_denomonator fraction divider denominator (Can be written only when SYSCONFIG[2] is 0.)

ASPC_CLK_CTRL
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	frac_div_ratio_sel fraction clk divider ratio select: (Can be written only when SYSCONFIG[2] is 0.) 1’b0: ratio is more than 40 1’b1: ratio is less than 35
5	RW	0x0	pdm_clk_en Pdm clk enable.working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 1’b0: pdm clk disable 1’b1: pdm clk enable
4	RW	0x0	div_type_sel divider type select signal 1’b0: fraction divider 1’b1: integer divider (Can be written only when SYSCONFIG[2] is 0.)
3	RW	0x0	clk_polar ASP_CLK polarity selection (Can be written only when SYSCONFIG[2] is 0.) 1’b0: no inverted 1’b1: inverted
2:0	RW	0x0	pdm_ds_ratio DS_RATIO,working at PDM mode (Can be written only when SYSCONFIG[2] is 0.) 3'b000: sample rate 192k/176.5k/128k 3'b001: sample rate 96kk/88.2k/64k 3'b010: sample rate 48kk/44.1k/32k 3'b011: sample rate 24kk/22.05k/16k 3'b100: sample rate 12kk/11.025k/8k

ASPC_HPF_CTRL
Address: Operational Base + offset (0x0010)

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

Bit	Attr	Reset Value	Description
3	RW	0x0	hpfle HPFLE high pass filter enable for left channel 1'b0: high pass filter for right channel is disabled 1'b1: high pass filter for right channel is enabled
2	RW	0x0	hpfre HPFRE high pass filter enable for right channel 1'b0: high pass filter for right channel is disabled 1'b1: high pass filter for right channel is enabled
1:0	RW	0x0	hpf_cf HPF_CF high pass filter configure register 2'b00: 3.79Hz 2'b01: 60Hz 2'b10: 243Hz 2'b11: 493Hz

ASPC_FIFO_CTRL

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	rft Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO threshold interrupt is triggered
7:0	RO	0x00	rfl RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO

ASPC_DMA_CTRL

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	rde Receive DMA Enable 1’b0 : Receive DMA disabled 1’b1 : Receive DMA enabled
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x1f	rdl Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1

ASPC_INT_EN

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	rxoie RX overflow interrupt enable 1'b0: disable 1'b1: enable
0	RW	0x0	rxtie RX threshold interrupt enable 1'b0: disable 1'b1: enable

ASPC_INT_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	W1C	0x0	rxoic RX overflow interrupt clear, high active, auto clear
0	RO	0x0	reserved

ASPC_INT_ST

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	rxoi RX overflow interrupt 1'b0: inactive 1'b1: active
0	RO	0x0	rxfi RX full interrupt 1'b0: inactive 1'b1: active

ASPC_RXFIFO_DATA_REG

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdr Receive FIFO shadow Register When the register is read, data in the receive FIFO is accessed

ASPC_DATA0R_REG

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0r Data of the path 0 right channel

ASPC_DATA0L_REG

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data0l Data of the path 0 left channel

ASPC_DATA1R_REG

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	data1r Data of the path 1 right channel

ASPC_DATA1L_REG

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data1l Data of the path 1 left channel

ASPC_DATA2R_REG

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2r Data of the path 2 right channel

ASPC_DATA2L_REG

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data2l Data of the path 2 left channel

ASPC_DATA3R_REG

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3r Data of the path 3 right channel

ASPC_DATA3L_REG
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	data3l Data of the path 3 left channel

ASPC_DATA_VALID
Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RC	0x0	path0_vld 1’b0: DATA0R_REG, DATA0L_REG value is invalid 1’b1: DATA0R_REG, DATA0L_REG value is valid
2	RC	0x0	path1_vld 1’b0: DATA1R_REG, DATA1L_REG value is invalid 1’b1: DAT1R_REG, DATA1L_REG value is valid
1	RC	0x0	path2_vld 1’b0: DATA2R_REG, DATA2L_REG value is invalid 1’b1: DATA2R_REG, DATA2L_REG value is valid
0	RC	0x0	path3_vld 1’b0: DATA3R_REG, DATA3L_REG value is invalid 1’b1: DATA3R_REG, DATA3L_REG value is valid

ASPC_VERSION
Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x59313030	version ASPC version

ASPC_RXDR
Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	receive_fifo_data FIFO data can be read from these registers, This register is used when the access address is Increment

7.5 Interface Description

Table 7-2 ASPC Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
O_asp_clk	O	GPIO1_A4/LCDC_D0/I2S1_8CH_SCLK_RX_M0/PDM_8CH_CLK_M0	GRF_GPIO1A_IOMUX[9:8]=2’B11
		GPIO1_B6/LCDC_D10/I2S1_8CH_SCLK_RX_M1/PDM_8CH_CLK_M1/MAC_MDIO	GRF_GPIO1B_IOMUX_L[15:12]=4’B100
		GPIO2_A6/I2S0_8CH_SCLK_RX/PDM_8CH_CLK_S_M2	GRF_GPIO2A_IOMUX[13:12]=2’B10
		GPIO2_A4/I2S0_8CH_MCLK/PDM_8CH_CLK_M_M2	GRF_GPIO2A_IOMUX[13:12]=2’B10 GRF_SOC_CON12[2]=1’B1
I_asp_data0	I	GPIO1_B3/LCDC_D7/I2S1_8CH_SDI0_M0/PDM_8CH_SDI0_M0	GRF_GPIO1B_IOMUX_L[7:6]=2’B11 GRF_SOC_CON2[12]=2’B00
		GPIO1_C5/LCDC_D17/I2S1_8CH_SDI0_M1/PDM_8CH_SDI0_M1/MAC_RXD1	GRF_GPIO1C_IOMUX_H[3:0]=4’B100 GRF_SOC_CON2[12]=2’B01
		GPIO2_B5/I2S0_8CH_SDI0/PDM_8CH_SDI0_M2	GRF_GPIO2B_IOMUX[11:10]=2’B10 GRF_SOC_CON2[12]=2’B10
I_asp_data1	I	GPIO1_B2/LCDC_D6/I2S1_8CH_SDO3_SDI1_M0/PDM_8CH_SDI1_M0	GRF_GPIO1B_IOMUX_L[5:4]=2’B11 GRF_SOC_CON2[12]=2’B00
		GPIO1_C4/LCDC_D16/I2S1_8CH_SDO3_SDI1_M1/PDM_8CH_SDI1_M1/MAC_RXD0	GRF_GPIO1C_IOMUX_L[15:12]=4’B100 GRF_SOC_CON2[12]=2’B01
		GPIO2_B6/I2S0_8CH_SDI1/PDM_8CH_SDI1_M2	GRF_GPIO2B_IOMUX[13:12]=2’B10 GRF_SOC_CON2[12]=2’B10
I_asp_data2	I	GPIO1_B1/LCDC_D5/I2S1_8CH_SDO2_SDI2_M0/PDM_8CH_SDI2_M0	GRF_GPIO1B_IOMUX_L[3:2]=2’B11 GRF_SOC_CON2[12]=2’B00
		GPIO1_C3/LCDC_D15/I2S1_8CH_SDO2_SDI2_M1/PDM_8CH_SDI2_M1/MAC_TXD1	GRF_GPIO1C_IOMUX_L[11:8]=4’B100 GRF_SOC_CON2[12]=2’B01
		GPIO2_B7/I2S0_8CH_SDI2/PDM_8CH_SDI2_M2	GRF_GPIO2B_IOMUX[15:14]=2’B10 GRF_SOC_CON2[12]=2’B10
I_asp_data3	I	GPIO1_B0/LCDC_D4/I2S1_8CH_SDO1_SDI3_M0/PDM_8CH_SDI3_M0	GRF_GPIO1B_IOMUX_L[1:0]=2’B11 GRF_SOC_CON2[12]=2’B00
		GPIO1_C2/LCDC_D14/I2S1_8CH_SDO1_SDI3_M1/PDM_8CH_SDI3_M1/MAC_TXD0	GRF_GPIO1C_IOMUX_L[7:4]=4’B100 GRF_SOC_CON2[12]=2’B01
		GPIO2_C0/I2S0_8CH_SDI3/PDM_8CH_SDI3_M2	GRF_GPIO2C_IOMUX[1:0]=2’B10 GRF_SOC_CON2[12]=2’B10

Notes: I=input, O=output, I/O=input/output, bidirectional
When use GPIO2_A4/I2S0_8CH_MCLK/PDM_8CH_CLK_M_M2, the output enable is control by pmic_sleep

7.6 Application Notes

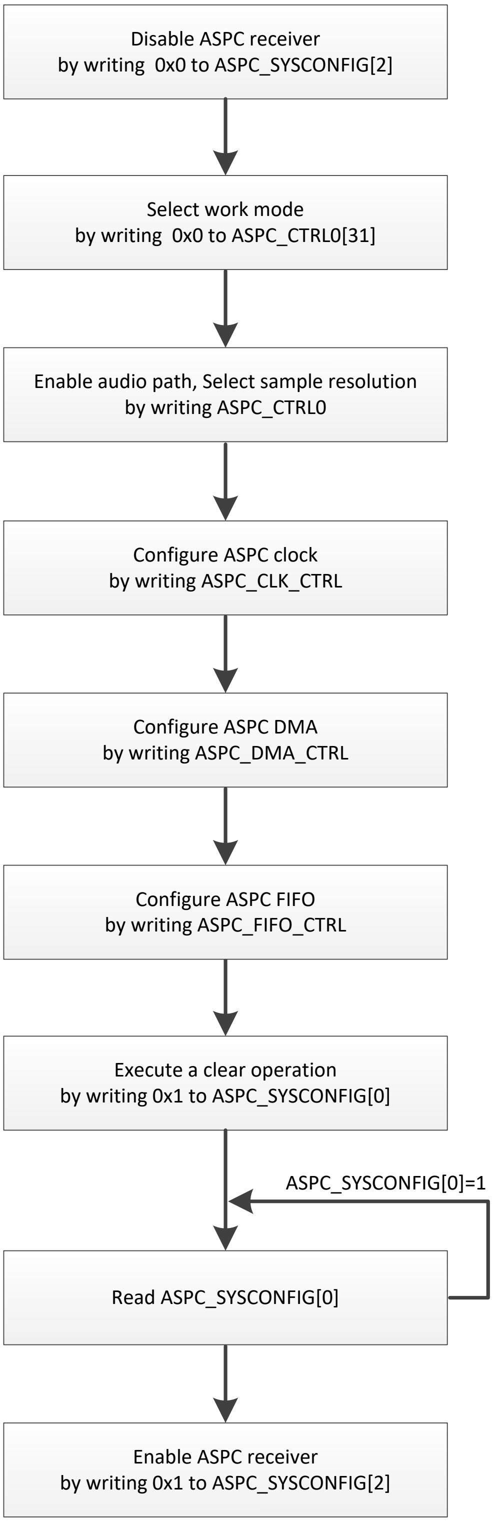

Disable ASPC receiver
by writing 0x0 to ASPC_SYSCONFIG[2]
Select work mode
by writing 0x0 to ASPC_CTRL0[31]
Enable audio path, Select sample resolution
by writing ASPC_CTRL0
Configure ASPC clock
by writing ASPC_CLK_CTRL
Configure ASPC DMA
by writing ASPC_DMA_CTRL
Configure ASPC FIFO
by writing ASPC_FIFO_CTRL
Execute a clear operation
by writing 0x1 to ASPC_SYSCONFIG[0]
ASPC_SYSCONFIG[0]=1
Read ASPC_SYSCONFIG[0]
Enable ASPC receiver
by writing 0x1 to ASPC_SYSCONFIG[2]

Fig. 7-6 ASPC operation flow

Chapter 8 Audio Codec

8.1 Overview

Audio Codec is a low power, high resolution, stereo CODEC solution which employs Sigma-Delta noise-shaping technique. The ADC, DAC and power amplifier are integrated to provide total solutions. With 24 bits resolution for DAC and 24 bits resolution for ADC, Audio Codec is suitable for applications in high end consumer digital audio systems, automobile audio, multimedia and digital systems. It supports following features:

- 24 bits DAC with 93dB(A-weighted) SNR
- Support 16~32Ω headphone out and 10kΩ line output
- 24 bits ADC with 92dB(A-weighted) SNR
- Support single-ended or differential microphone input
- Automatic Level Control (ALC) for smooth audio recording
- Pure logic process: no need for mixed signal layers and less mask cost
- Low power: DAC 3mA per channel, ADC 2.5mA per channel, less than 0.05mA for standby
- Support Mono, Stereo, 5.1 and 7.1 HiFi channel performance
- Programmable input and output analog gains
- Digital interpolation and decimation filter integrated
- Sampling rate of 8kHz/11.025kHz/12kHz/16kHz/22.05kHz/24kHz/32kHz/44.1kHz/48kHz/64kHz/88.2kHz/96kHz/128kHz/176.4kHz/192kHz
- 1.8V supply for analog and 1.0V supply for digital

8.2 Block Diagram

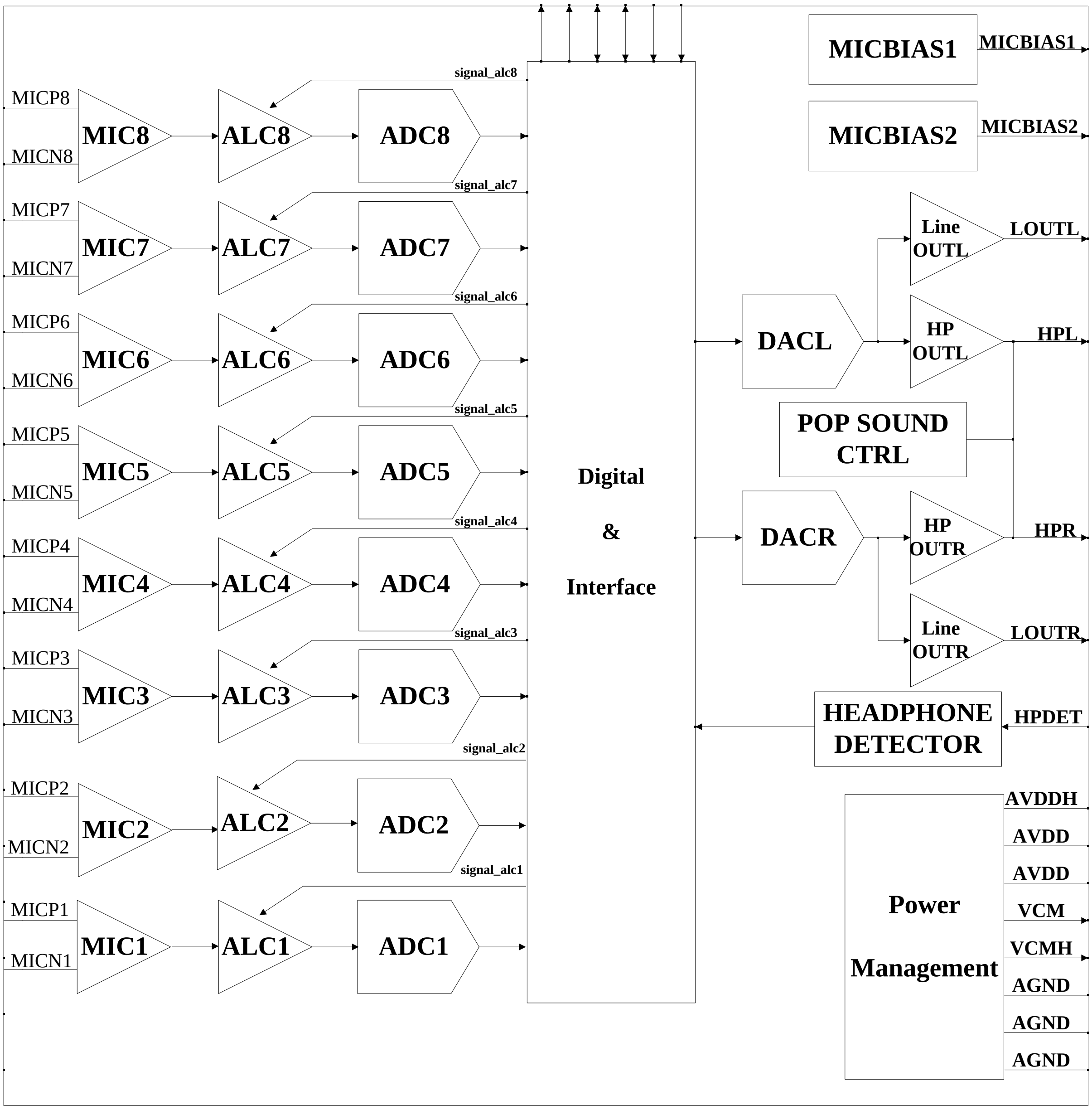


Fig. 8-1 Audio Codec overview

The default Audio Codec IP contains 2 DAC channels for stereo playback and 8 ADC channels for stereo recording from microphone.

8.3 Function description

8.3.1 Digital Interface

Audio Codec IP provides the I2S PCM interface of audio data stream which gets into DAC and out from ADC, both of which can be configured in master or slave mode. Different audio data formats are available for different operating modes. This is demonstrated in following table.

Table 8-1 Supported Data Formats in Different Modes

Data Formats	**ADC**		**DAC**	
	Master	**Slave**	**Master**	**Slave**
Left Justified	√	×	√	√
Right Justified	√	√	√	√
I2S	√	√	√	√
DSP/PCM mode A	√	√	√	√
DSP/PCM mode B	√	×	√	√

I2S_PCM interface supports five audio data formats: Left Justified mode, Right Justified mode, I2S mode, DSP/PCM mode A and mode B. They are valid when the device operates as a master or slave.

For Left Justified mode, the data format is illustrated in Fig. 8-2.The MSB is valid at the first rising edge of sck after ws transition is done. The other valid bits up to the LSB are transmitted sequentially. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear before every ws transition, which means the data in this period is invalid.

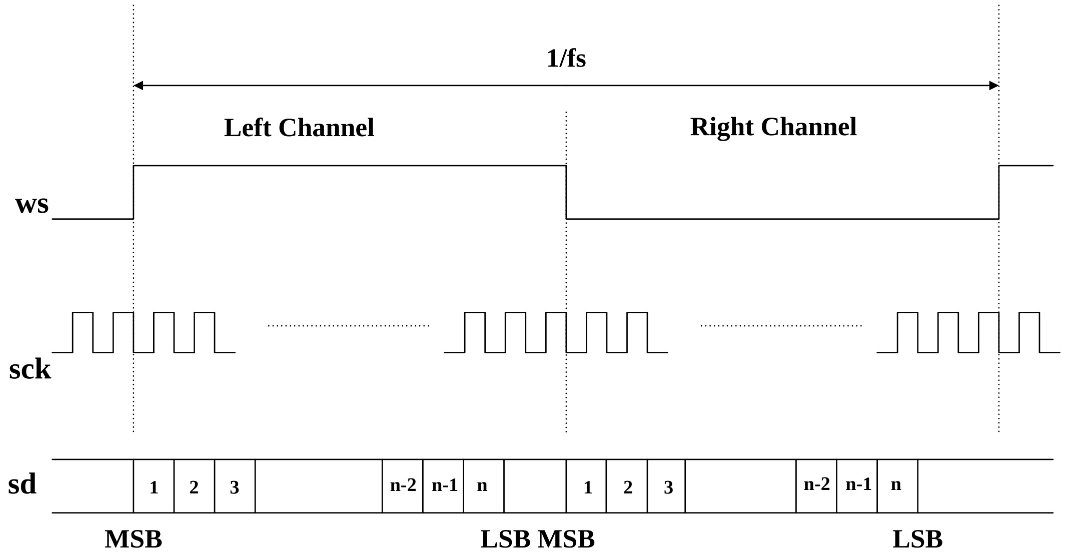


Fig. 8-2 Left Justified Mode (assuming n-bit word length)

For Right Justified mode, the data format is shown in Fig. 8-3. The LSB becomes valid at the last rising edge of sck before ws transition is done. As the MSB is transmitted first, the other valid bits up to the MSB are followed in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may exist after every ws transition, which means the data in this period is invalid.

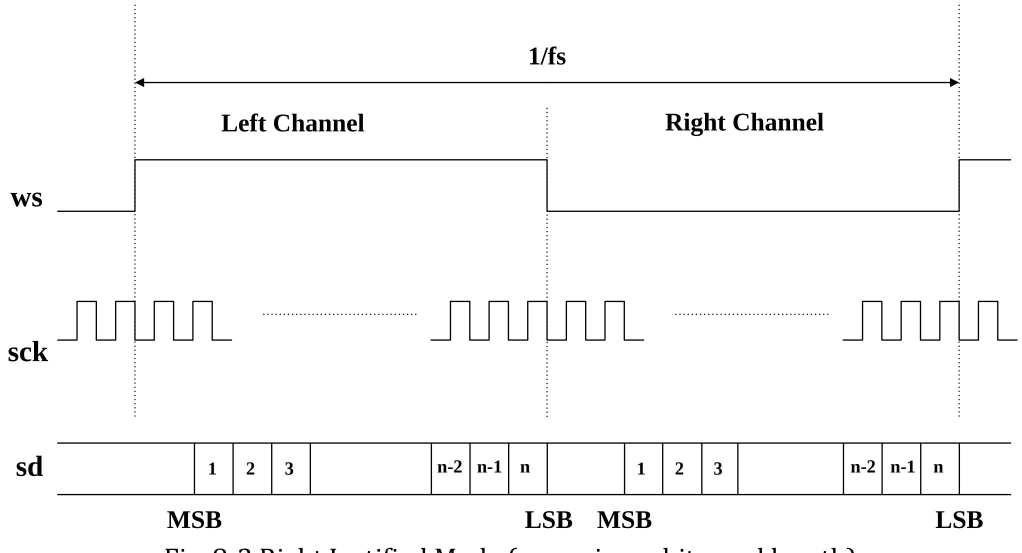


Fig. 8-3 Right Justified Mode (assuming n-bit word length)

For I2S mode, the data format is depicted in Fig. 8-4. The MSB becomes available at the second rising edge of sck when ws transition is done. The other valid bits up to the LSB are transmitted in order. Due to varied word length, different sck frequency and sample rate, some unused sck cycles may appear between the LSB of the current sample and the MSB of the next one, which means the data in this period can be ignored.

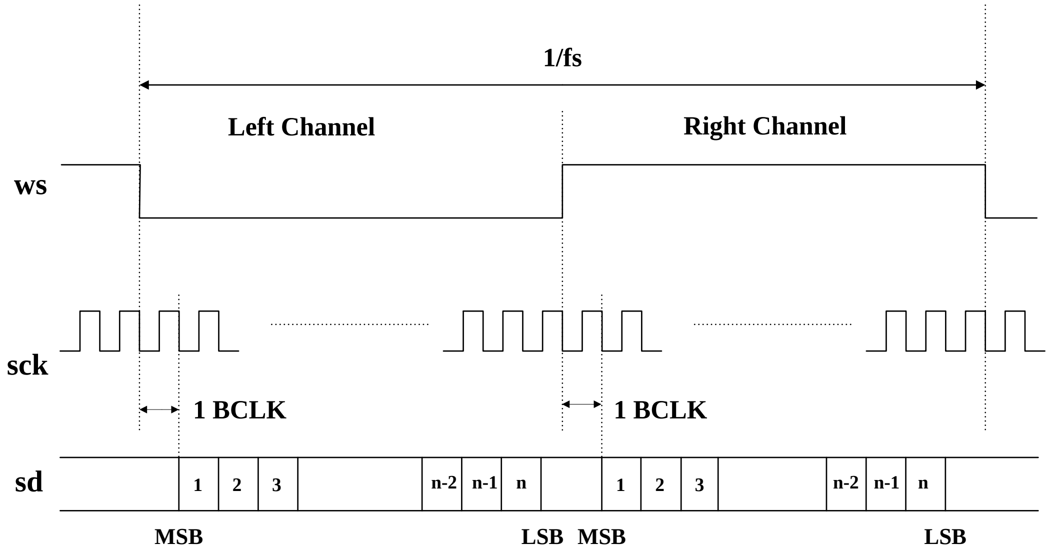


Fig. 8-4 I2S Mode (assuming n-bit word length)

For DSP/PCM mode, the left channel data is transmitted first, followed by right channel data. For DSP/PCM mode A/B, the MSB is available at the second and first rising edge of sck after the rising edge of ws respectively, as shown in Fig. 8-5 and Fig.8-6. Based on word length, sck frequency and sample rate, there may be some invalid data between the LSB of the right channel data and the next sample.

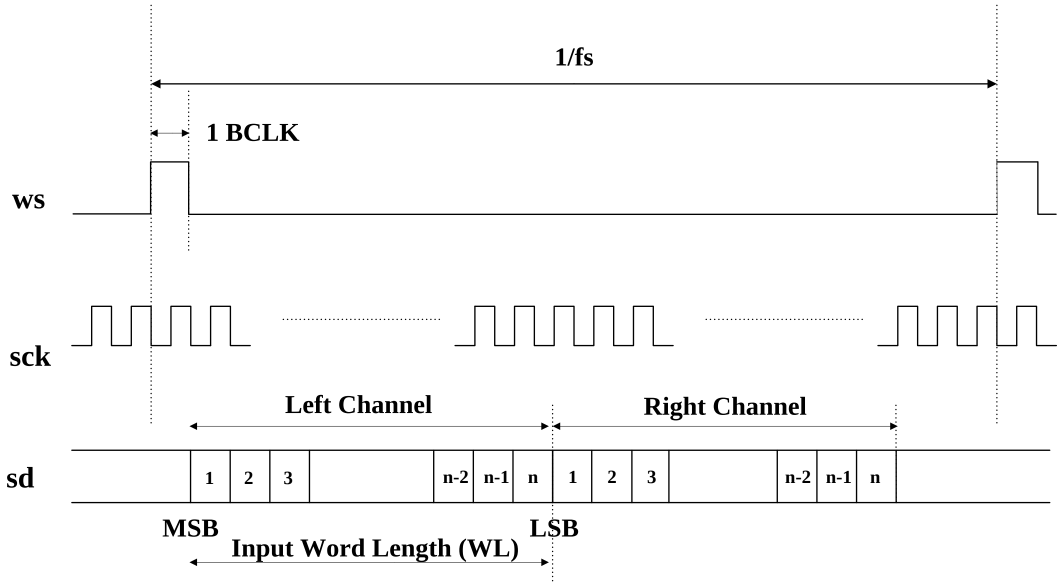


Fig. 8-5 DSP/PCM Mode A (assuming n-bit word length)

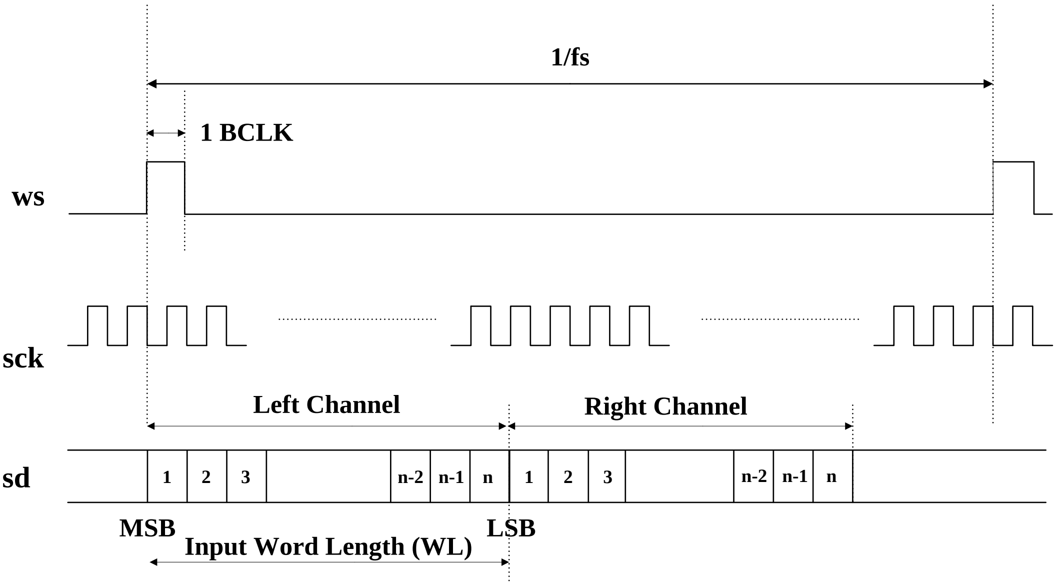


Fig. 8-6 DSP/PCM Mode B (assuming n-bit word length)

8.3.2 Analog Interface

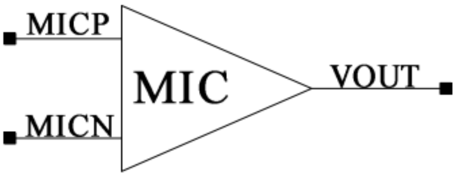


Fig. 8-7 Microphone Input

There are two microphone input pin, MICP and MICN. They can be configured as differential inputs by the microphone PGA (MIC).

The signal of microphone output should be input to Audio Codec through DC-blocking capacitor, as shown in following figure.

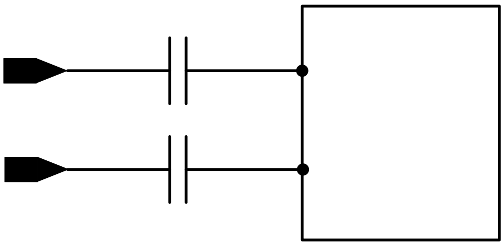


Fig. 8-8 Input DC-blocking capacitor

The capacitance and input resistance from a high pass filter. For example, when the gain of the MIC module is 20dB, the input resistance is 45kΩ and 0.1uF DC-blocking capacitor is used, the lower cut-off frequency is:

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 45 \times 10^{3} \times 0.1 \times 10^{-6}} = 35.4Hz$$

The capacitance of the DC-blocking capacitor should be determined by the minimum input impedance and application requirements. If the output of microphone is single-ended, the audio ADC input should be connected as following figure.

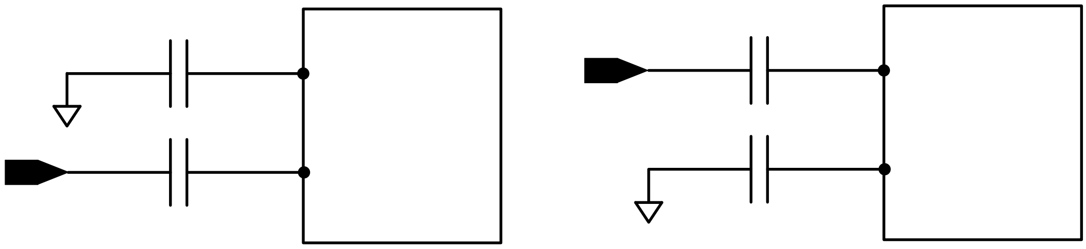


Fig. 8-9 Input single-ended DC-blocking capacitor

Microphone PGA has two gains to amplify the input signal, that is 0dB and +20dB.
Automatic Level Control (ALC) function is included to adjust the signal level, which is input into ADC. ALC will measure the signal magnitude and compare it to defined threshold. Then it will adjust the ALC controlled gain (ALC_L and ALC_R) according to the comparison result. When the AGC function is off, the gain (ALC_L and ALC_R) is directly controlled by register, and the programmable gain range is from -18dB to +28.5dB, and the tuning step is 1.5dB.
Audio Codec supports two line output and two headphone out configurations. The DAC output can drive load through DC-blocking capacitor.
In the configuration using DC-blocking capacitor, shown in the following figure, the headphone ground is connected to the real ground. The capacitance and the load resistance determine the lower cut-off frequency. For instance, if 600Ω load and 4.7uF DC-blocking capacitor are used, the lower cut-off frequency is:

$$f = \frac{1}{2\pi RC} = \frac{1}{2\pi \times 600 \times 4.7 \times 10^{-6}} = 56.5Hz$$

The DC-blocking capacitor can be increased to lower the cut-off frequency for better bass response.

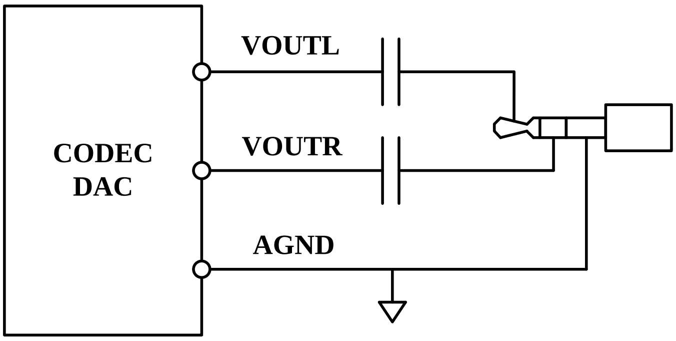


Fig. 8-10 Output DC-blocking capacitor

The headphone-out driver has a gain range from -39dB to +6dB with a tuning step of 1.5dB.
The line out driver has a gain range from -16.5dB to +6dB with a tuning step of 1.5dB.

8.3.3 Interface Relationship

The relationships between I2S interface and parallel audio data for different ADC and DAC channels are shown in the following figure.

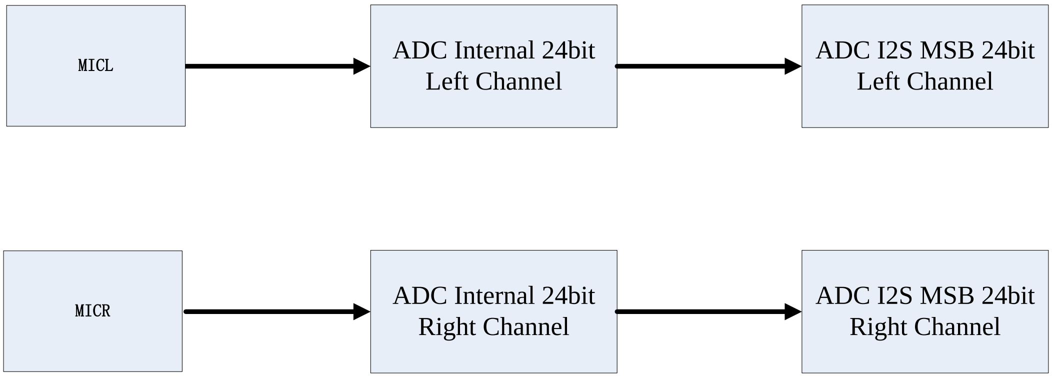

MICL
ADC Internal 24bit Left Channel
ADC I2S MSB 24bit Left Channel
MICR
ADC Internal 24bit Right Channel
ADC I2S MSB 24bit Right Channel

Fig. 8-11 ADC Channels Relationship

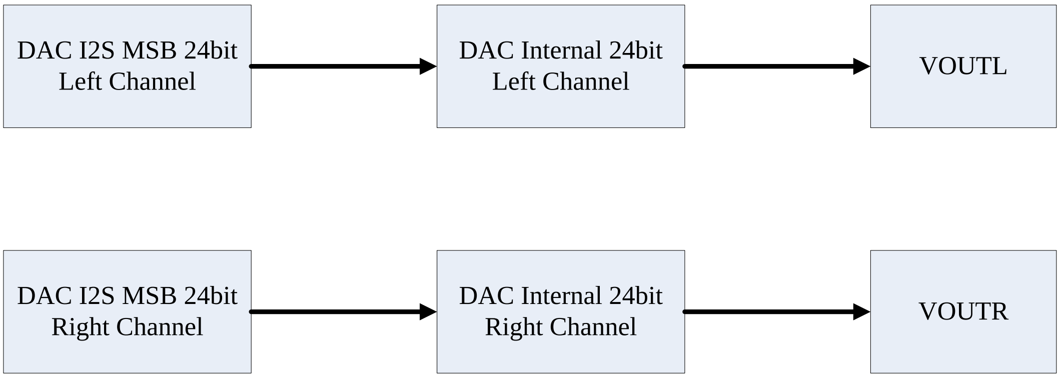

DAC I2S MSB 24bit Left Channel
DAC Internal 24bit Left Channel
VOUTL
DAC I2S MSB 24bit Right Channel
DAC Internal 24bit Right Channel
VOUTR

Fig. 8-12 DAC Channels Relationship

8.4 Register Description

8.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
ACODEC_GLB_CON	0x0000	W	0x00000007	Global Control Register
ACODEC_ADC_DIG_CON1	0x0004	W	0x00000050	ADC Digital Control 1 Register
ACODEC_ADC_DIG_CON2	0x0008	W	0x0000000e	ADC Digital Control 2 Register
ACODEC_ADC_DIG_CON3	0x000c	W	0x0000000f	ADC Digital Control 3 Register
ACODEC_ADC_DIG_CON4	0x0010	W	0x00000004	ADC Digital Control 4 Register
ACODEC_ADC_DIG_CON7	0x001c	W	0x00000000	ADC Digital Control 7 Register
ACODEC_ALC_DIG_CON0	0x0040	W	0x00000000	ALC Digital Control 0 Register
ACODEC_ALC_DIG_CON1	0x0044	W	0x00000046	ALC Digital Control 1 Register
ACODEC_ALC_DIG_CON2	0x0048	W	0x00000041	ALC Digital Control 2 Register
ACODEC_ALC_DIG_CON3	0x004c	W	0x0000002c	ALC Digital Control 3 Register
ACODEC_ALC_DIG_CON4	0x0050	W	0x00000000	ALC Digital Control 4 Register
ACODEC_ALC_DIG_CON5	0x0054	W	0x00000026	ALC Digital Control 5 Register
ACODEC_ALC_DIG_CON6	0x0058	W	0x00000040	ALC Digital Control 5 Register
ACODEC_ALC_DIG_CON7	0x005c	W	0x00000036	ALC Digital Control 7 Register
ACODEC_ALC_DIG_CON8	0x0060	W	0x00000020	ALC Digital Control 8 Register
ACODEC_ALC_DIG_CON9	0x0064	W	0x00000038	ALC Digital Control 9 Register
ACODEC_ALC_DIG_CON12	0x0070	W	0x0000000c	ALC Digital Control 12 Register
ACODEC_DAC_DIG_CON1	0x0304	W	0x00000050	DAC Digital Control 1 Register
ACODEC_DAC_DIG_CON2	0x0308	W	0x0000000e	DAC Digital Control 2 Register

Name	Offset	Size	Reset Value	Description
ACODEC_DAC_DIG_CON3	0x030c	W	0x00000000	DAC Digital Control 3 Register
ACODEC_DAC_DIG_CON4	0x0310	W	0x00000022	DAC Digital Control 4 Register
ACODEC_DAC_DIG_CON5	0x0314	W	0x00000000	DAC Digital Control 5 Register
ACODEC_DAC_DIG_CON10	0x0328	W	0x0000000f	DAC Digital Control 10 Register
ACODEC_DAC_DIG_CON11	0x032c	W	0x000000ff	DAC Digital Control 11 Register
ACODEC_DAC_DIG_CON12	0x0330	W	0x00000003	DAC Digital Control 12 Register
ACODEC_DAC_DIG_CON13	0x0334	W	0x000000c2	DAC Digital Control 13 Register
ACODEC_DAC_DIG_CON14	0x0338	W	0x00000000	DAC Digital Control 14 Register
ACODEC_ADC_ANA_CON0	0x0340	W	0x00000000	ADC Analog Control 0 Register
ACODEC_ADC_ANA_CON1	0x0344	W	0x00000000	ADC Analog Control 1 Register
ACODEC_ADC_ANA_CON2	0x0348	W	0x00000000	ADC Analog Control 2 Register
ACODEC_ADC_ANA_CON3	0x034c	W	0x0000000c	ALC Analog Control 3 Register
ACODEC_ADC_ANA_CON4	0x0350	W	0x0000000c	ALC Analog Control 4 Register
ACODEC_ADC_ANA_CON5	0x0354	W	0x00000000	ADC Analog Control 5 Register
ACODEC_ADC_ANA_CON6	0x0358	W	0x00000000	ADC Analog Control 6 Register
ACODEC_ADC_ANA_CON7	0x035c	W	0x00000050	ADC Analog Control 7 Register
ACODEC_ADC_ANA_CON8	0x0360	W	0x00000000	ADC Analog Control 8 Register
ACODEC_ADC_ANA_CON10	0x0368	W	0x00000001	ADC Analog Control 10 Register
ACODEC_ADC_ANA_CON11	0x036c	W	0x00000000	ADC Analog Control 11 Register
ACODEC_DAC_ANA_CON0	0x0440	W	0x00000000	DAC Analog Control 0 Register
ACODEC_DAC_ANA_CON1	0x0444	W	0x00000011	DAC Analog Control 1 Register
ACODEC_DAC_ANA_CON2	0x0448	W	0x00000000	DAC Analog Control 2 Register
ACODEC_DAC_ANA_CON3	0x044c	W	0x00000000	DAC Analog Control 3 Register
ACODEC_DAC_ANA_CON4	0x0450	W	0x00000000	DAC Analog Control 4 Register
ACODEC_DAC_ANA_CON5	0x0454	W	0x00000000	DAC Analog Control 5 Register
ACODEC_DAC_ANA_CON6	0x0458	W	0x00000000	DAC Analog Control 6 Register
ACODEC_DAC_ANA_CON7	0x045c	W	0x00000088	DAC Analog Control 7 Register
ACODEC_DAC_ANA_CON8	0x0460	W	0x00000088	DAC Analog Control8 Register
ACODEC_DAC_ANA_CON12	0x0470	W	0x00000022	DAC Analog Control 12 Register
ACODEC_DAC_ANA_CON13	0x0474	W	0x00000000	DAC Analog Control 13 Register

*Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access*
When configure ACODEC_ADC_DIG_CONx, user should add a offset address to correspond channel:
add offset 0x000 for left_0(ADC1) and right_0(ADC2) channel,
add offset 0x0c0 for left_1(ADC3) and right_1(ADC4) channel,
add offset 0x180 for left_2(ADC5) and right_2(ADC6) channel,
add offset 0x240 for left_3(ADC7) and right_3(ADC8) channel.

When configure ACODEC_ALC_DIG_CONx, user should add a offset address to correspond channel:
add offset 0x000 for left channel ALC1,
add offset 0x040 for right channel ALC2,
add offset 0x0c0 for left channel ALC3,
add offset 0x100 for right channel ALC4,
add offset 0x180 for left channel ALC5,
add offset 0x1c0 for right channel ALC6,
add offset 0x240 for left channel ALC7,
add offset 0x280 for right channel ALC8.
When configure ACODEC_ADC_ANA_CONx, user should add a offset address to correspond channel:
add offset 0x000 for left_0(ADC1) and right_0(ADC2) channel,
add offset 0x040 for left_1(ADC3) and right_1(ADC4) channel,
add offset 0x080 for left_2(ADC5) and right_2(ADC6) channel,
add offset 0x0c0 for left_3(ADC7) and right_3(ADC8) channel.

8.4.2 Detail Register Description

ACODEC_GLB_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	ADC_BIST_RESET ADC bist module reset 1'b0: Reset 1'b1: Work
6	RW	0x0	DAC_BIST_RESET DAC bist module reset 1'b0: Reset 1'b1: Work
5	RW	0x0	ADC_MCLK_GATING The enable signal of the ADC MCLK gating which cover the ADC data path(ADC1~ADC8). 1'b0: Enable 1'b1: Disable Note: The ADC data path don't contain the AGC and I2S TX module.
4	RW	0x0	DAC_MCLK_GATING The enable signal of the DAC MCLK gating which cover the DAC data path(DAC1~DAC2) 1'b0: Enable 1'b1: Disable Note: The DAC data path don't contain the I2S RX module.
3	RO	0x0	reserved
2	RW	0x1	CODEC_ADC_DIGITAL_RST Codec ADC digital core reset. This reset only reset the codec data path. 1'b0: Reset 1'b1: Work
1	RW	0x1	CODEC_DAC_DIGITAL_RST Codec DAC digital core reset. This reset only reset the codec data path. 1'b0: Reset 1'b1: Work

Bit	Attr	Reset Value	Description
0	RW	0x1	CODEC_SYSTEM_RST Codec system reset. This signal will reset the registers which control all the digital and analog part. 1'b0: Reset 1'b1: Work

ACODEC_ADC_DIG_CON1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	ADC_I2S_LRC_POL ADC I2S LRC Polarity 1'b0: Normal 1'b1: Reversal
6:5	RW	0x2	ADC_I2S_VALID_LEN Valid Word Length in one 1/2Frame 2'b00: 16 bits 2'b01: 20 bits 2'b10: 24 bits 2'b11: 32 bits
4:3	RW	0x2	ADC_I2S_MODE ADC I2S MODE 2'b00: Right Justified Mode 2'b01: Left Justified Mode 2'b10: I2S Mode 2'b11: PCM Mode Note: Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RO	0x0	reserved
1	RW	0x0	ADC_I2S_SWAP ADC Left-Right SWAP 1'b0: Normal 1'b1: Swap
0	RW	0x0	ADC_I2S_TYPE ADC I2S TYPE 1'b0: Stereo 1'b1: Mono

ACODEC_ADC_DIG_CON2

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	ADC_MODE_IO ADC I2S Mode select for IO pin 1'b0: Slave mode 1'b1: Master mode
4	RW	0x0	ADC_MODE ADC I2S Mode select for inner module 1'b0: Slave mode 1'b1: Master mode
3:2	RW	0x3	ADC_I2S_FRAME_LEN ADC I2S 1/2Frame Word Length 2'b00: 16 bits 2'b01: 20 bits 2'b10: 24 bits 2'b11: 32 bits
1	RW	0x1	ADC_I2S_RST ADC I2S Reset 1'b0: Reset 1'b1: Work
0	RW	0x0	ADC_I2S_BIT_CLK_POL ADC I2S Bit Clock Polarity 1'b0: Normal 1'b1: Reversal

ACODEC_ADC_DIG_CON3

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x3	ADC_LEFT_CHANNEL_BIST ADC Left Channel Bist Mode select 2'b00: Normal Left Channel data to the PCM(normal mode) 2'b01: Sine Wave to the PCM(bist mode enable and need to set the GLB_CON[7]) 2'b10: Cube Wave to the PCM(bist mode enable and need to set the GLB_CON[7]) 2'b11: Normal Right channel data to the PCM(normal mode)
1:0	RW	0x3	ADC_RIGHT_CHANNEL_BIST ADC Right Channel Bist Mode select 2'b00: Normal Right Channel data to the PCM(normal mode) 2'b01: Sine Wave to the PCM(bist mode enable and need to set the GLB_CON[7]) 2'b10: Cube Wave to the PCM(bist mode enable and need to set the GLB_CON[7]) 2'b11: Normal Left channel data to the PCM(normal mode)

ACODEC_ADC_DIG_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x1	HPF_DIS Disable the high pass filter in the ADC path 1'b0: HPF Enable 1'b1: HPF Disable
1:0	RW	0x0	HPF_CUT_FREQ Select the cut-off frequency of the high pass filter 2'b00: 20Hz 2'b01: 245Hz 2'b10: 612Hz 2'b11: reserved

ACODEC_ADC_DIG_CON7

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x0	Config_ADCL_DATA Configure the input of ADCL data path when the register ADC_DIG_CON7[1] is 1. Note: Please keep the default value
3:2	RW	0x0	Config_ADCR_DATA Configure the input of ADCR data path when the register ADC_DIG_CON7[0] is 1. Note: Please keep the default value
1	RW	0x0	ADCL_DATA_SEL Select the input of the ADCL data path 1'b0: Select the normal path as the input of ADC data path 1'b1: Select the register ADC_DIG_CON7[5:4] as the input of the ADC data path Note: Please keep the default value
0	RW	0x0	ADCR_DATA_SEL Select the input of the ADCR data path 1'b0: Select the normal path as the input of ADC data path 1'b1: Select the register ADC_DIG_CON7[3:2] as the input of the ADC data path Note: Please keep the default value

ACODEC_ALC_DIG_CON0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	GAIN_ATT_METHOD Choose the method to control the gain attack 1'b0: Normal way 1'b1: Jack way
5:4	RW	0x0	CTRL_GEN_METHOD There are four methods to generate the control signals 2'b00: Normal way 2'b01: Jack way 1 2'b10: Jack way 2 2'b11: Jack way 3 This register is used to choose the method to generate the control signals according to the actual situation
3:0	RW	0x0	AGC_HOLD_TIME AGC hold time before gain is increased in normal mode 4'b0000: 0ms 4'b0001: 2ms 4'b0010: 4ms 4'b0011: 8ms 4'b0100: 16ms 4'b0101: 32ms 4'b0110: 64ms 4'b0111: 128ms 4'b1000: 256ms 4'b1001: 512ms 4'b1010: 1s 4'b1011~1111: 0ms

ACODEC_ALC_DIG_CON1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:4	RW	0x4	DECAY_TIME Decay (gain ramp-up) time Normal MODE(reg_agc_mode =0) 4'b0000: 500us 4'b0001: 1ms 4'b0010: 2ms 4'b0011: 4ms 4'b0100: 8ms 4'b0101: 16ms 4'b0110: 32ms 4'b0111: 64ms 4'b1000: 128ms 4'b1001: 256ms 4'b1010: 512ms 4'b1011~1111: 512ms Limiter MODE(reg_agc_mode =1) 4'b0000: 125us 4'b0001: 250us 4'b0010: 500us 4'b0011: 1ms 4'b0100: 2ms 4'b0101: 4ms 4'b0110: 8ms 4'b0111: 16ms 4'b1000: 32ms 4'b1001: 64ms 4'b1010: 128ms 4'b1011~1111: 128ms

Bit	Attr	Reset Value	Description
3:0	RW	0x6	ATTACK_TIME Attack (gain ramp-down) Time Normal MODE(reg_agc_mode =0) 4'b0000: 125us 4'b0001: 250us 4'b0010: 500us 4'b0011: 1ms 4'b0100: 2ms 4'b0101: 4ms 4'b0110: 8ms 4'b0111: 16ms 4'b1000: 32ms 4'b1001: 64ms 4'b1010: 128ms 4'b1011~1111: 125us Limiter MODE(reg_agc_mode =1) 4'b0000: 32us 4'b0001: 64us 4'b0010: 125us 4'b0011: 250us 4'b0100: 500us 4'b0101: 1ms 4'b0110: 2ms 4'b0111: 4ms 4'b1000: 8ms 4'b1001: 16ms 4'b1010: 32ms 4'b1011~1111: 32us

ACODEC_ALC_DIG_CON2
Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	AGC_MODE Determines the AGC mode of operation 1'b0: AGC mode(normal mode) 1'b1: Limiter mode
6	RW	0x1	AGC_ZERO_CRO_EN AGC users zero cross enable 1'b0: Disabled 1'b1: Enabled, the AGC gain will update at zero cross enable
5	RW	0x0	LOW_AMPLITUDE_RECOVERY When in the limiter mode, the low amplitude signal will recovery in two modes: 1'b0: The gain will recovery to the value of the reg_pga_lvol 1'b1: The gain will recovery to the gain at the moment when the mode changes from AGC to Limiter
4	RW	0x0	FAST_DEC_CTRL When the amplitude of the signal is more than 87.5% of the Full scale, use this signal to control the fast decrement: 1'b0: Disabled 1'b1: Enabled
3	RW	0x0	AGC_NOISE_GATE_EN AGC noise gate function enable 1'b0: Disabled 1'b1: Enabled
2:0	RW	0x1	AGC_NOISE_GATE_THRESHOLD AGC noise gate threshold 3'b000: -39dB 3'b001: -45dB 3'b010: -51dB 3'b011: -57dB 3'b100: -63dB 3'b101: -69dB 3'b110: -75dB 3'b111: -81dB

ACODEC_ALC_DIG_CON3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x1	LEFT_CHANNEL_PGA_ZERO_CRO_EN Left channel input PGA zero cross enable 1'b0: Update gain when gain register changes. 1'b1: Update gain on 1st zero cross after gain register write

Bit	Attr	Reset Value	Description
4:0	RW	0x0c	LEFT_CHANNEL_PGA_GAIN Left channel input PGA gain 5'b0_0000: -18dB 5'b0_0001: -16.5dB 5'b0_0010: -15dB 5'b0_0011: -13.5dB 5'b0_0100: -12dB 5'b0_0101: -10.5dB 5'b0_0110: -9dB 5'b0_0111: -7.5dB 5'b0_1000: -6dB 5'b0_1001: -4.5dB 5'b0_1010: -3dB 5'b0_1011: -1.5dB 5'b0_1100: 0dB 5'b0_1101: +1.5dB 5'b0_1110: +3dB 5'b0_1111: +4.5dB 5'b1_0000: +6dB 5'b1_0001: +7.5dB 5'b1_0010: +9dB 5'b1_0011: +10.5dB 5'b1_0100: +12dB 5'b1_0101: +13.5dB 5'b1_0110: +15dB 5'b1_0111: +16.5dB 5'b1_1000: +18dB 5'b1_1001: +19.5dB 5'b1_1010: +21dB 5'b1_1011: +22.5dB 5'b1_1100: +24dB 5'b1_1101: +25.5dB 5'b1_1110: +27dB 5'b1_1111: +28.5dB

ACODEC_ALC_DIG_CON4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	SLOW_CLOCK_EN Slow clock enabled, used for the zero cross timeout

Bit	Attr	Reset Value	Description
2:0	RW	0x0	APPROXIMATE_RATE Approximate sample rate 3'b000: 96kHz 3'b001: 48kHz 3'b010: 44.1kHz 3'b011: 32kHz 3'b100: 24kHz 3'b101: 16kHz 3'b110: 12kHz 3'b111: 8kHz Note：According to the sample rate to choose the right configuration

ACODEC_ALC_DIG_CON5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x26	LOW_8_BITS_AGC_MAX The low 8 bits of the AGC maximum level

ACODEC_ALC_DIG_CON6

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x40	HIGH_8_BITS_AGC_MAX The high 8 bits of the AGC maximum level

ACODEC_ALC_DIG_CON7

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x36	LOW_8_BITS_AGC_MIN The low 8 bits of the AGC minimum level

ACODEC_ALC_DIG_CON8

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	HIGH_8_BITS_AGC_MIN The high 8 bits of the AGC minimum level

ACODEC_ALC_DIG_CON9

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	AGC_FUNCTION_SELECT AGC function select 1'b0: AGC function off 1'b1: AGC function enable
5:3	RW	0x7	MAX_GAIN_PGA Set maximum gain of PGA 3'b000: -13.5dB 3'b001: -7.5dB 3'b010: -1.5dB 3'b011: +4.5dB 3'b100: +10.5dB 3'b101: +16.5dB 3'b110: +22.5dB 3'b111: +28.5dB
2:0	RW	0x0	MIN_GAIN_PGA Set minimum gain of PGA 3'b000: -18dB 3'b001: -12dB 3'b010: -6dB 3'b011: 0dB 3'b100: +6dB 3'b101: +12dB 3'b110: +18dB 3'b111: +24dB

ACODEC_ALC_DIG_CON12
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x0c	AGC_GAIN_VALUE It shows the current value of the agc gain value. When the agc function is disable, this value is 0x0c

ACODEC_DAC_DIG_CON1
Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	DAC_I2S_LRC_POL DAC I2S LRC Polarity 1'b0: Normal 1'b1: Reversal

Bit	Attr	Reset Value	Description
6:5	RW	0x2	DAC_I2S_VALID_LEN Valid Word Length in one 1/2Frame 2'b00: 16 bits 2'b01: 20 bits 2'b10: 24 bits 2'b11: 32 bits
4:3	RW	0x2	DAC_I2S_MODE DAC I2S MODE 2'b00: Right Justified Mode 2'b01: Left Justified Mode 2'b10: I2S Mode 2'b11: PCM Mode Note: Same word length in 1/2frame and valid data is not supported in Right Justified Mode. For example, 32/24 or 24/20 is supported, but 32/32 or 24/24 is not supported. (1/2frame length/valid data length)
2	RW	0x0	DAC_I2S_SWAP DAC Left-Right SWAP 1'b0: Normal 1'b1: Swap
1:0	RO	0x0	reserved

ACODEC_DAC_DIG_CON2

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	DAC_MODE_IO DAC I2S Mode select for IO pin 1'b0: Slave mode 1'b1: Master mode
4	RW	0x0	DAC_MODE DAC I2S Mode select for inner module 1'b0: Slave mode 1'b1: Master mode
3:2	RW	0x3	DAC_I2S_FRAME_LEN DAC I2S 1/2Frame Word Length 2'b00: 16 bits 2'b01: 20 bits 2'b10: 24 bits 2'b11: 32 bits
1	RW	0x1	DAC_I2S_RST DAC I2S Reset 1'b0: Reset 1'b1: Work

Bit	Attr	Reset Value	Description
0	RW	0x0	DAC_I2S_BIT_CLK_POL DAC I2S Bit Clock Polarity 1'b0: Normal 1'b1: Reversal

ACODEC_DAC_DIG_CON3

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:2	RW	0x0	DAC_LEFT_CHANNEL_BIST DAC Left Channel Bist Mode select 2'b00: Normal Right Channel data to the PCM(normal mode) 2'b01: Sine Wave to the PCM(bist mode enable and need to set the GLB_CON[6]) 2'b10: Cube Wave to the PCM(bist mode enable and need to set the GLB_CON[6]) 2'b11: Normal Left channel data to the PCM(normal mode)
1:0	RW	0x0	DAC_RIGHT_CHANNEL_BIST DAC Right Channel Bist Mode select 2'b00: Normal Right Channel data to the PCM(normal mode) 2'b01: Sine Wave to the PCM(bist mode enable and need to set the GLB_CON[6]) 2'b10: Cube Wave to the PCM(bist mode enable and need to set the GLB_CON[6]) 2'b11: Normal Left channel data to the PCM(normal mode)

ACODEC_DAC_DIG_CON4

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:4	RW	0x2	MODULATOR_GAIN Set the gain of the modulator 3'b000:0db 3'b001:2db 3'b010:2.8db 3'b011:3.5db 3'b100:4.2db 3'b101:4.8db
3	RO	0x0	reserved
2:0	RW	0x2	CIC_IP_FILTER_GAIN Set the gain of output signal of the cic interpolation filter

ACODEC_DAC_DIG_CON5

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	DACL_DATA_SEL Choose the input of the DAC left channel. 1'b0: Normal Data 1'b1: Using the register to control the input data of the DAC left channel Note: When we choose the register to control the input data of the DAC left channel, we can use the register DAC_DIG_CON10~DAC_DIG_CON11 to set the input data value
1	RW	0x0	DACR_DATA_SEL Choose the input of the DAC right channel. 1'b0: Normal Data 1'b1: Using the register to control the input data of the DAC right channel Note: When we choose the register to control the input data of the DAC left channel, we can use the register DAC_DIG_CON10~DAC_DIG_CON11 to set the input data value
0	RO	0x0	reserved

ACODEC_DAC_DIG_CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0xf	DAC_DATA_HIGH Configure the bit [11:8] of DAC data when choose the register value as the input of the DAC channel. Please reference the description of register DAC_DIG_CON5[2:1]. Note: Please keep the default value

ACODEC_DAC_DIG_CON11

Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0xff	DAC_DATA_LOW Configure the bit [7:0] of DAC data when choose the register value as the input of the DAC channel. Please reference the description of register DAC_DIG_CON5[2:1]. Note: Please keep the default value

ACODEC_DAC_DIG_CON12

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x3	DELAY_TIME_DETECT_HIGH Please refer the DAC_DIG_CON13.

ACODEC_DAC_DIG_CON13

Address: Operational Base + offset (0x0334)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0xc2	DELAY_TIME_DETECT_LOW The value(DELAY_TIME_DETECT_HIGH[1:0]<<8 + DELAY_TIME_DETECT_LOW[7:0]) indicate the delay time of detecting the headphone inserting or unplugged.

ACODEC_DAC_DIG_CON14

Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	HEADPHONE_INSERT The flag indicates whether the headphone have inserted. 1'b0: Don't have 1'b1: Having headset to be inserted

ACODEC_ADC_ANA_CON0

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	ADC_CH2_MIC_MUTE The mute signal of MIC module in the ADC channel 2. 1'b0: Mute 1'b1: Work
6	RW	0x0	ADC_CH2_MIC_INIT The initial signal of MIC module in the ADC channel 2. 1'b0: Initialization 1'b1: Work
5	RW	0x0	ADC_CH2_MIC_EN The enable signal of MIC module in the ADC channel 2. 1'b0: Disable 1'b1: Enable
4	RW	0x0	ADC_CH2_BUF_REF_EN The enable signal of reference voltage buffer of ADC channel 2. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	ADC_CH1_MIC_MUTE The mute signal of MIC module in the ADC channel 1. 1'b0: Mute 1'b1: Work
2	RW	0x0	ADC_CH1_MIC_INIT The initial signal of MIC module in the ADC channel 1. 1'b0: Initialization 1'b1: Work
1	RW	0x0	ADC_CH1_MIC_EN The enable signal of MIC module in the ADC channel 1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	ADC_CH1_BUF_REF_EN The enable signal of reference voltage buffer of ADC channel 1. 1'b0: Disable 1'b1: Enable

ACODEC_ADC_ANA_CON1

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x0	ADC_CH2_MIC_GAIN The gain signal of MIC module in the ADC channel 2. 2'b00: 0dB 2'b11: 20dB Others: reserve Note: When ADC is used as loopback, it is recommended to configure 2'b00. When ADC is used as MIC input, it is recommended to configure 2'b11.
3:2	RO	0x0	reserved
1:0	RW	0x0	ADC_CH1_MIC_GAIN The gain signal of MIC module in the ADC channel 1. 2'b00: 0dB 2'b11: 20dB Others: reserve Note: When ADC is used as loopback, it is recommended to configure 2'b00. When ADC is used as MIC input, it is recommended to configure 2'b11.

ACODEC_ADC_ANA_CON2

Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	ADC_CH2_ZEROCROSS_DET_EN The enable signal of input zero-crossing detection module in the ADC channel 2. 1'b0: Disable 1'b1: Enable
5	RW	0x0	ADC_CH2_ALC_INIT The initial signal of ALC module in the ADC channel 2. 1'b0: Initialization 1'b1: Work
4	RW	0x0	ADC_CH2_ALC_EN The enable signal of ALC module in the ADC channel 2. 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	ADC_CH1_ZEROCROSS_DET_EN The enable signal of input zero-crossing detection module in the ADC channel 1. 1'b0: Disable 1'b1: Enable
1	RW	0x0	ADC_CH1_ALC_INIT The initial signal of ALC module in the ADC channel 1. 1'b0: Initialization 1'b1: Work
0	RW	0x0	ADC_CH1_ALC_EN The enable signal of ALC module in the ADC channel 1. 1'b0: Disable 1'b1: Enable

ACODEC_ADC_ANA_CON3

Address: Operational Base + offset (0x034c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x0c	ADC_CH1_ALC_GAIN The gain signal of ALC module in the ADC channel 1. 5'b0_0000: -18dB 5'b0_0001: -16.5dB 5'b0_0010: -15dB 5'b0_0011: -13.5dB 5'b0_0100: -12dB 5'b0_0101: -10.5dB 5'b0_0110: -9dB 5'b0_0111: -7.5dB 5'b0_1000: -6dB 5'b0_1001: -4.5dB 5'b0_1010: -3dB 5'b0_1011: -1.5dB 5'b0_1100: 0dB 5'b0_1101: +1.5dB 5'b0_1110: +3dB 5'b0_1111: +4.5dB 5'b1_0000: +6dB 5'b1_0001: +7.5dB 5'b1_0010: +9dB 5'b1_0011: +10.5dB 5'b1_0100: +12dB 5'b1_0101: +13.5dB 5'b1_0110: +15dB 5'b1_0111: +16.5dB 5'b1_1000: +18dB 5'b1_1001: +19.5dB 5'b1_1010: +21dB 5'b1_1011: +22.5dB 5'b1_1100: +24dB 5'b1_1101: +25.5dB 5'b1_1110: +27dB 5'b1_1111: +28.5dB

ACODEC_ADC_ANA_CON4

Address: Operational Base + offset (0x0350)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x0c	ADC_CH2_ALC_GAIN The gain signal of ALC module in the ADC channel 2. 5'b0_0000: -18dB 5'b0_0001: -16.5dB 5'b0_0010: -15dB 5'b0_0011: -13.5dB 5'b0_0100: -12dB 5'b0_0101: -10.5dB 5'b0_0110: -9dB 5'b0_0111: -7.5dB 5'b0_1000: -6dB 5'b0_1001: -4.5dB 5'b0_1010: -3dB 5'b0_1011: -1.5dB 5'b0_1100: 0dB 5'b0_1101: +1.5dB 5'b0_1110: +3dB 5'b0_1111: +4.5dB 5'b1_0000: +6dB 5'b1_0001: +7.5dB 5'b1_0010: +9dB 5'b1_0011: +10.5dB 5'b1_0100: +12dB 5'b1_0101: +13.5dB 5'b1_0110: +15dB 5'b1_0111: +16.5dB 5'b1_1000: +18dB 5'b1_1001: +19.5dB 5'b1_1010: +21dB 5'b1_1011: +22.5dB 5'b1_1100: +24dB 5'b1_1101: +25.5dB 5'b1_1110: +27dB 5'b1_1111: +28.5dB

ACODEC_ADC_ANA_CON5

Address: Operational Base + offset (0x0354)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	ADC_CH2_ADC_INIT The initial signal of ADC module in the ADC channel 2. 1'b0: Initialization 1'b1: Work

Bit	Attr	Reset Value	Description
5	RW	0x0	ADC_CH2_ADC_EN The enable signal of ADC module in the ADC channel 2. 1'b0: Disable 1'b1: Enable
4	RW	0x0	ADC_CH2_CLK_EN The enable signal of clock module in the ADC channel 2. 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	ADC_CH1_ADC_INIT The initial signal of ADC module in the ADC channel 1. 1'b0: Initialization 1'b1: Work
1	RW	0x0	ADC_CH1_ADC_EN The enable signal of ADC module in the ADC channel 1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	ADC_CH1_CLK_EN The enable signal of clock module in the ADC channel 1. 1'b0: Disable 1'b1: Enable

ACODEC_ADC_ANA_CON6

Address: Operational Base + offset (0x0358)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ADC_CURRENT_EN The enable signal of current source for ADC channel 1 and channel 2. 1'b0: Disable 1'b1: Enable

ACODEC_ADC_ANA_CON7

Address: Operational Base + offset (0x035c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RW	0x1	ADC_CH2_IN_SEL ADC_CH2_IN_SEL[1] is reserved, it should be 1'b0. ADC_CH2_IN_SEL[0], the signal to select microphone as input of the ADC channel 2 1'b0: Don't select 1'b1: Select Note: The register configuration is only valid for ADC2

Bit	Attr	Reset Value	Description
5:4	RW	0x1	ADC_CH1_IN_SEL ADC_CH1_IN_SEL[1] is reserved, it should be 1'b0. ADC_CH1_IN_SEL[0], the signal to select microphone as input of the ADC channel 1 1'b0: Don't select 1'b1: Select Note: The register configuration is only valid for ADC1
3	RW	0x0	ADC_MICROPHONE_BIAS_BUF_EN The enable signal of microphone bias voltage buffer. 1'b0: Disable 1'b1: Enable Note: Only the reg (ADC_ANA_CON7+0x40)[3] and reg (ADC_ANA_CON7+0x80)[3] represent the enable signal of MICBIAS1 and MICBIAS2 respectively
2:0	RW	0x0	ADC_LEVEL_RANGE_MICBIAS The level range control signal of MICBIAS voltage: 3'b000: 0.5*AVDD33 3'b001: 0.55*AVDD33 3'b010: 0.6*AVDD33 3'b011: 0.65*AVDD33 3'b100: 0.7*AVDD33 3'b101: 0.75*AVDD33 3'b110: 0.8*AVDD33 3'b111: 0.85*AVDD33 Note: Only the reg (ADC_ANA_CON7+0x0)[2:0] represent the level range control signal of MICBIAS voltage

ACODEC_ADC_ANA_CON8

Address: Operational Base + offset (0x0360)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	ADC_MICBIAS_CURRENT_EN The enable signal of current source for MICBIAS. 1'b0: Disable 1'b1: Enable Note: Only the reg (ADC_ANA_CON8+0x0)[4] represent the enable signal of current source for MICBIAS
3:0	RO	0x0	reserved

ACODEC_ADC_ANA_CON10

Address: Operational Base + offset (0x0368)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	ADC_RFF_EN The enable signal of reference voltage module. 1'b0: Disable 1'b1: Enable Note: Only the reg (ADC_ANA_CON10+0x0)[7] represent the enable signal of reference voltage module
6:0	RW	0x01	ADC_CURRENT_CHARGE_SEL The control signal to select current to pre-charge/dis-charge. CURRENT_CHARGE_SEL[6]: 1'b0: Choose the current I0 1'b1: Don't choose the current I0 CURRENT_CHARGE_SEL[5]: 1'b0: Choose the current 2*I0 1'b1: Don't choose the current 2*I0 CURRENT_CHARGE_SEL[4]: 1'b0: Choose the current 4*I0 1'b1: Don't choose the current 4*I0 CURRENT_CHARGE_SEL[3]: 1'b0: Choose the current 8*I0 1'b1: Don't choose the current 8*I0 CURRENT_CHARGE_SEL[2]: 1'b0: Choose the current 16*I0 1'b1: Don't choose the current 16*I0 CURRENT_CHARGE_SEL[1]: 1'b0: Choose the current 32*I0 1'b1: Don't choose the current 32*I0 CURRENT_CHARGE_SEL[0]: 1'b0: Choose the current 64*I0 1'b1: Don't choose the current 64*I0 Note: Only the reg (ADC_ANA_CON10+0x0)[6:0] represent the control signal to select current to pre-charge/dis_charge

ACODEC_ADC_ANA_CON11

Address: Operational Base + offset (0x036c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	ADC_ALCR_CON_GAIN_PGAR Enable the function that uses the ALCR module to control the gain of the PGAR 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
0	RW	0x0	ADC_ALCL_CON_GAIN_PGAL Enable the function that uses the ALCL module to control the gain of the PGAL 1'b0: Disable 1'b1: Enable

ACODEC_DAC_ANA_CON0
Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	DAC_HEADPHONE_DET_EN The enable signal of headphone detection module 1'b0: Disable 1'b1: Enable
0	RW	0x0	DAC_CURRENT_EN The enable signal of current source for DAC channel. 1'b0: Disable 1'b1: Enable

ACODEC_DAC_ANA_CON1
Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	DAC_BUF_REF_R_EN The enable signal of reference voltage buffer in the right DAC channel 1'b0: Disable 1'b1: Enable
5:4	RW	0x1	DAC_POP_SOUND_R The control signal of POP sound in the right DAC channel 2'b01: Initialization 2'b10: Work others: reserved
3	RO	0x0	reserved
2	RW	0x0	DAC_BUF_REF_L_EN The enable signal of reference voltage buffer in the left DAC channel 1'b0: Disable 1'b1: Enable
1:0	RW	0x1	DAC_POP_SOUND_L The control signal of POP sound in the left DAC channel 2'b01: Initialization 2'b10: Work others: reserved

ACODEC_DAC_ANA_CON2

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	DAC_R_DAC_INIT The initial signal of DAC module in the right DAC channel 1'b0: Initialization 1'b1: Work
6	RW	0x0	DAC_R_DAC_EN The enable signal of DAC module in the right DAC channel 1'b0: Disable 1'b1: Enable
5	RW	0x0	DAC_R_CLK_EN The enable signal of clock module in the right DAC channel. 1'b0: Disable 1'b1: Enable
4	RW	0x0	DAC_R_REF_EN The enable signal of reference voltage of DAC module in the right DAC channel 1'b0: Disable 1'b1: Enable
3	RW	0x0	DAC_L_DAC_INIT The initial signal of DAC module in the left DAC channel 1'b0: Initialization 1'b1: Work
2	RW	0x0	DAC_L_DAC_EN The enable signal of DAC module in the left DAC channel 1'b0: Disable 1'b1: Enable
1	RW	0x0	DAC_L_CLK_EN The enable signal of clock module in the left DAC channel. 1'b0: Disable 1'b1: Enable
0	RW	0x0	DAC_L_REF_EN The enable signal of reference voltage of DAC module in the left DAC channel 1'b0: Disable 1'b1: Enable

ACODEC_DAC_ANA_CON3

Address: Operational Base + offset (0x044c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	DAC_R_HPOUT_INIT The initial signal of HPOUT module in the right DAC channel 1'b0: Initialization 1'b1: Work
5	RW	0x0	DAC_R_HPOUT_EN The enable signal of HPOUT module in the right DAC channel 1'b0: Disable 1'b1: Enable
4	RW	0x0	DAC_R_HPOUT_MUTE The mute signal of HPOUT module in the right DAC channel 1'b0: Mute 1'b1: Work
3	RO	0x0	reserved
2	RW	0x0	DAC_L_HPOUT_INIT The initial signal of HPOUT module in the left DAC channel 1'b0: Initialization 1'b1: Work
1	RW	0x0	DAC_L_HPOUT_EN The enable signal of HPOUT module in the left DAC channel 1'b0: Disable 1'b1: Enable
0	RW	0x0	DAC_L_HPOUT_MUTE The mute signal of HPOUT module in the left DAC channel 1'b0: Mute 1'b1: Work

ACODEC_DAC_ANA_CON4
Address: Operational Base + offset (0x0450)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RW	0x0	DAC_R_GAIN The gain signal of line out module in the right DAC channel. 2'b00: -6dB 2'b01: -3dB 2'b10: -1.5dB 2'b11: 0dB
5	RW	0x0	DAC_R_LINEOUT_MUTE The mute signal of line out module in the right DAC channel 1'b0: Mute 1'b1: Work
4	RW	0x0	DAC_R_LINEOUT_EN The enable signal of line out module in the right DAC channel 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3:2	RW	0x0	DAC_L_GAIN The gain signal of line out module in the left DAC channel. 2'b00: -6dB 2'b01: -3dB 2'b10: -1.5dB 2'b11: 0dB
1	RW	0x0	DAC_L_LINEOUT_MUTE The mute signal of line out module in the left DAC channel 1'b0: Mute 1'b1: Work
0	RW	0x0	DAC_L_LINEOUT_EN The enable signal of line out module in the left DAC channel 1'b0: Disable 1'b1: Enable

ACODEC_DAC_ANA_CON5

Address: Operational Base + offset (0x0454)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	DAC_L_HPOUT_GAIN The gain signal of HPOUT module in the left DAC channel. 5'b00000: -39dB ... 5'b11010: 0dB ... 5'b11101: 4.5dB 5'b11110~5'b11111: 6dB Step: 1.5dB

ACODEC_DAC_ANA_CON6

Address: Operational Base + offset (0x0458)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	DAC_R_HPOUT_GAIN The gain signal of HPOUT module in the right DAC channel. 5'b00000: -39dB ... 5'b11010: 0dB ... 5'b11101: 4.5dB 5'b11110~5'b11111: 6dB Step: 1.5dB

ACODEC_DAC_ANA_CON7

Address: Operational Base + offset (0x045c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x8	HPOUTR_DS HPOUT driver strength for right channel The smaller the value , the greater the driver strength
3:0	RW	0x8	HPOUTL_DS HPOUT driver strength for left channel The smaller the value , the greater the driver strength

ACODEC_DAC_ANA_CON8

Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x8	HPOUTR_DS LINEOUT driver strength for right channel The smaller the value , the greater the driver strength
3:0	RW	0x8	LINEOUTL_DS LINEOUT driver strength for left channel The smaller the value , the greater the driver strength

ACODEC_DAC_ANA_CON12

Address: Operational Base + offset (0x0470)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RW	0x0	The select signal of HPMIX module in the right DAC channel. DAC_R_HPMIX_SEL [1] is reserved, it should be 1'b0. DAC_R_HPMIX_SEL[0],Select input from I2S 1'b0: Don't select 1'b1: Select
5:4	RW	0x2	DAC_R_HPMIX_GAIN The gain signal of HPMIX module in the right DAC channel. 2'b01: -6dB 2'b10: 0dB others: reserved
3:2	RW	0x0	DAC_L_HPMIX_SEL The select signal of HPMIX module in the left DAC channel. DAC_L_HPMIX_SEL [1] is reserved, it should be 1'b0. DAC_L_HPMIX_SEL[0],Select input from I2S 1'b0: Don't select 1'b1: Select

Bit	Attr	Reset Value	Description
1:0	RW	0x2	DAC_L_HPMIX_GAIN The gain signal of HPMIX module in the left DAC channel. 2'b01: -6dB 2'b10: 0dB others: reserved

ACODEC_DAC_ANA_CON13

Address: Operational Base + offset (0x0474)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	DAC_R_HPMIX_MUTE The mute signal of HPMIX module in the right DAC channel 1'b0: Mute 1'b1: Work
5	RW	0x0	DAC_R_HPMIX_INIT The initial signal of HPMIX module in the right DAC channel 1'b0: Initialization 1'b1: Work
4	RW	0x0	DAC_R_HPMIX_EN The enable signal of HPMIX module in the right DAC channel 1'b0: Disable 1'b1: Enable
3	RO	0x0	reserved
2	RW	0x0	DAC_L_HPMIX_MUTE The mute signal of HPMIX module in the left DAC channel 1'b0: Mute 1'b1: Work
1	RW	0x0	DAC_L_HPMIX_INIT The initial signal of HPMIX module in the left DAC channel 1'b0: Initialization 1'b1: Work
0	RW	0x0	DAC_L_HPMIX_EN The enable signal of HPMIX module in the left DAC channel 1'b0: Disable 1'b1: Enable

8.5 Interface Description

N/A

8.6 Application Note

8.6.1 Enable DAC I2S Interface Flow

0. Supply the power of the digital part and reset the Audio Codec.
1. Provide stable clock bus clock(CRU_CLKGAT_CON14[12]).
2. Provide stable clock pin_dac_mclk(Clock source from mclk_tx of I2S_8CH_2).
3. Reset ACODEC (CRU_SOFTRST_CON9[9]).
4. Reset ACODEC by soft reset register as follow:

Set ACODEC_GLB_CON=0x0
Set ACODEC_GLB_CON=0x1
5. Configure the ACODEC_DAC_DIG_CON1~3 related to the DAC I2S function.
6. Enable the DAC as follow:
Set ACODEC_GLB_CON=0x3
7. Begin to input the I2S data stream to the I2S interface of the ACODEC DAC.

8.6.2 Enable DC I2S Interface Flow

1. Provide stable clock bus clock(CRU_CLKGAT_CON14[12]).
2. Provide stable clock pin_dac_mclk(Clock source from mclk_rx of I2S_8CH_2).
3. Reset ACODEC (CRU_SOFTRST_CON9[9]).
4. Reset ACODEC by soft reset register as follow:
Set ACODEC_GLB_CON=0x0
Set ACODEC_GLB_CON=0x1
5. Configure the ACODEC_ADC_DIG_CON1~3 for the all ADC channel, a offset address should be added as follow:
add offset 0x000 for left_0(ADC1) and right_0(ADC2) channel;
add offset 0x0c0 for left_1(ADC3) and right_1(ADC4) channel;
add offset 0x180 for left_2(ADC5) and right_2(ADC6) channel;
add offset 0x240 for left_3(ADC7) and right_3(ADC8) channel.
It's recommended to configure ACODEC_ADC_DIG_CON3 to 0x0.
6. Enable the ADC as follow:
Set ACODEC_GLB_CON=0x5
7. Begin to use the I2S interface of ADC to output the parallel data.

8.6.3 Power Up

1. Supply the power of digital part and reset the Audio Codec.
2. Configure ACODEC_DAC_ANA_CON1[1:0] and ACODEC_DAC_ANA_CON1[5:4] to 0x1, to setup dc voltage of the DAC channel output.
3. Configure the register ACODEC_ADC_ANA_CON10[6:0] to 0x1.
4. Supply the power of the analog part(AVDD,AVDDRV).
5. Configure the register ACODEC_ADC_ANA_CON10[7] to 0x1 to setup reference voltage.
6. Change the register ACODEC_ADC_ANA_CON10[6:0] from the 0x1 to 0x7f step by step or configure the ACODEC_ADC_ANA_CON10[6:0] to 0x7f directly. The suggestion slot time of the step is 20ms.
7. Wait until the voltage of VCM keeps stable at the AVDD/2.
8. Configure the register ACODEC_ADC_ANA_CON10[6:0] to the appropriate value(expect 0x0) for reducing power.

8.6.4 Power Off

0. Keep the power on and disable the DAC and ADC.
1. Configure the register ACODEC_ADC_ANA_CON10[6:0] to 0x1.
2. Configure the register ACODEC_ADC_ANA_CON10[7] to 0x0.
3.Change the register ACODEC_ADC_ANA_CON10[6:0] from the 0x1 to 0x7f step by step or configure the ACODEC_ADC_ANA_CON10[6:0] to 0x7f directly. The suggestion slot time of the step is 20ms.
4. Wait until the voltage of VCM keeps stable at the AGND.
5. Power off the analog power supply.
6. Power off the digital power supply.

8.6.5 Enable MICBIAS

0. Power up the ACODEC and keep the AVDDH stable.
1. Configure ACODEC_ADC_ANA_CON7[2:0] to the certain value.
2. Wait until the VCMH keep stable.
3. Configure ACODEC_ADC_ANA_CON8[4] to 0x1.
4. Configure the (ADC_ANA_CON7+0x40)[3] or (ADC_ANA_CON7+0x80)[3] to 0x1.
(ADC_ANA_CON7+0x40)[3] used to control the MICBIAS1, and
(ADC_ANA_CON7+0x80)[3] used to control the MICBIAS2.

8.6.6 Disable MICBIAS

0. Enable the MICBIAS and keep the Audio Codec stable.

1. Configure the (ADC_ANA_CON7+0x40)[3] or (ADC_ANA_CON7+0x80)[3] to 0x0.
2. Configure ACODEC_ADC_ANA_CON8[4] to 0x0.

8.6.7 Enable DAC

0. Power up the Audio Codec and input the mute signal.
1. Set the ACODEC_DAC_ANA_CON0[0] to 0x1, to enable the current source of DAC.
2. Set the ACODEC_DAC_ANA_CON1[6] and ACODEC_DAC_ANA_CON1[2] to 0x1, to enable the reference voltage buffer.
3. Set the ACODEC_DAC_ANA_CON1[5:4] and ACODEC_DAC_ANA_CON1[1:0] to 0x2, to select the dc voltage of the HPDRV module from internal.
4. Set the ACODEC_DAC_ANA_CON13[4] and ACODEC_DAC_ANA_CON13[0] to 0x1, to enable the HPMIX of DAC.
5.Set the ACODEC_DAC_ANA_CON13[5] and ACODEC_DAC_ANA_CON13[1] to 0x1, to end the initialization of HPMIX.
6. Set the ACODEC_ DAC_ANA_CON4[4] and ACODEC_ DAC_ANA_CON4[0] to 0x1, to enable the line out module in DAC.
7. Set the ACODEC_ DAC_ANA_CON3[5] and ACODEC_ DAC_ANA_CON3[1] to 0x1, to enable the HPDRV module in DAC.
8. Set the ACODEC_ DAC_ANA_CON3[6] and ACODEC_ DAC_ANA_CON3[2] to 0x1, to end the initialization of HPDRV.
9. Set ACODEC_DAC_ANA_CON2[4] and ACODEC_DAC_ANA_CON2[0] to 0x1, to enable the high and low reference voltage of DAC.
10. Set ACODEC_DAC_ANA_CON2[5] and ACODEC_DAC_ANA_CON2[1] to 0x1 , to enable the clock module of DAC.
11. Set ACODEC_DAC_ANA_CON2[6] and ACODEC_DAC_ANA_CON2[2] to 0x1, to enable the DAC.
12. Set ACODEC_DAC_ANA_CON2[7] and ACODEC_DAC_ANA_CON2[3] to 0x1,to end the initialization of DAC.
13. Set ACODEC_DAC_ANA_CON12[7:6] and ACODEC_DAC_ANA_CON12[3:2] to select the input signal of HPMIX module.
14. Set ACODEC_DAC_ANA_CON13[6] and ACODEC_DAC_ANA_CON13[2] to 0x1 to end the mute station of the HPMIX module in the DAC.
15. Set ACODEC_DAC_ANA_CON12[5:4] and ACODEC_DAC_ANA_CON12[1:0] to select the gain of HPMIX module.
16. Set ACODEC_DAC_ANA_CON3[4] and ACODEC_DAC_ANA_CON3[0] to 0x1 to end the mute station of the DRV module in the DAC.
17. Set ACODEC_DAC_ANA_CON4[5] and ACODEC_DAC_ANA_CON4[1] to 0x1 to end the mute station of the line out module in the DAC.
18. Set ACODEC_ DAC_ANA_CON5[4:0] and ACODEC_ DAC_ANA_CON6[4:0] to select the gain of HPDDRV module in the DAC.
19. Set ACODEC_ DAC_ANA_CON4[7:6] and ACODEC_ DAC_ANA_CON4[3:2] to select the gain of line out module in the DAC.
20. Play the music.

Note1. If the ACODEC_ DAC_ANA_CON12[6] or ACODEC_ DAC_ANA_CON12[2] is set to 0x0, ignoring the step9~12.
Note2. If the ACODEC_ DAC_ANA_CON12[7] or ACODEC_ DAC_ANA_CON12[3] is set to 0x1, the ADC0 or ADC1 should be enabled firstly, and please refer to Enable ADC Configuration Standard Usage Flow(expect step7~step9,step14).
Note3. If no opening the line out, ignoring the step6, step17 and step19.
Note4. If no opening the headphone out, ignoring the step3,step7~8,step16 and step18.
Note5. In the step18, adjust the register step by step to the appropriate value and taking 10ms as one time step.

8.6.8 Disable DAC

0. Keep the DAC channel work and input the mute signal.
1. Set ACODEC_ DAC_ANA_CON4[7:6] and ACODEC_ DAC_ANA_CON4[3:2] to select the gain of line out module in the DAC.
2. Set ACODEC_ DAC_ANA_CON5[4:0] and ACODEC_ DAC_ANA_CON6[4:0] to 0x0, to select the gain of HPDDRV module in the DAC.

3. Set ACODEC_DAC_ANA_CON13[6] and ACODEC_DAC_ANA_CON13[2] to 0x0, to mute the HPMIX module in the DAC.
4.Set ACODEC_DAC_ANA_CON12[7:6] and ACODEC_DAC_ANA_CON12[3:2] to 0x0, to select the input signal of HPMIX module.
5. Set ACODEC_DAC_ANA_CON3[4] and ACODEC_DAC_ANA_CON3[0] to 0x0 to mute the DRV module in the DAC.
6. Set ACODEC_DAC_ANA_CON2[7] and ACODEC_DAC_ANA_CON2[3] to 0x0, to initialize the DAC.
7. Set the ACODEC_ DAC_ANA_CON3[5] and ACODEC_ DAC_ANA_CON3[1] to 0x0, to disable the HPDRV module in DAC.
8. Set ACODEC_DAC_ANA_CON4[5] and ACODEC_DAC_ANA_CON4[1] to 0x0 to mute the line out module.
9. Set the ACODEC_ DAC_ANA_CON4[4] and ACODEC_ DAC_ANA_CON4[0] to 0x0, to disable the line out module in DAC.
10. Set the ACODEC_DAC_ANA_CON13[4] and ACODEC_DAC_ANA_CON13[0] to 0x0, to disable the HPMIX of DAC.
11. Set ACODEC_DAC_ANA_CON2[6] and ACODEC_DAC_ANA_CON2[2] to 0x0, to disable the DAC.
12. Set ACODEC_DAC_ANA_CON2[5] and ACODEC_DAC_ANA_CON2[1] to 0x0 , to disable the clock module of DAC.
13. Set ACODEC_DAC_ANA_CON2[4] and ACODEC_DAC_ANA_CON2[0] to 0x0, to disable the high and low reference voltage of DAC.
14. Set the ACODEC_DAC_ANA_CON1[5:4] and ACODEC_DAC_ANA_CON1[1:0] to 0x1, to select the dc voltage of the HPDRV module from VCM PAD.
15. Set the ACODEC_DAC_ANA_CON1[6] and ACODEC_DAC_ANA_CON1[2] to 0x0, to disable the reference voltage buffer.
16. Set the ACODEC_DAC_ANA_CON0[0] to 0x0, to disable the current source of DAC.
17. Set the ACODEC_ DAC_ANA_CON3[6] and ACODEC_ DAC_ANA_CON3[2] to 0x0, to begin the initialization of HPDRV.
18. Set the ACODEC_DAC_ANA_CON13[5] and ACODEC_DAC_ANA_CON13[1] to 0x0, to begin the initialization of HPMIX.
19. Set ACODEC_DAC_ANA_CON12[5:4] and ACODEC_DAC_ANA_CON12[1:0] to select the gain of HPMIX module.
Note1. In the step2, adjusting the register step by step to the appropriate value and taking 20ms as time step.
Note2. If the ACODEC_DAC_ANA_CON12[7] or ACODEC_DAC_ANA_CON12[3] is set to 0x1, add the steps from the section Disable ADC Configuration Standard Usage Flow after complete the step 19.

8.6.9 Enable ADC

0. Power up the Audio Codec.
1. Configure the register ACODEC_ADC_ANA_CON7[7:6], to set microphone as input of the ADC right channel.
Configure the register ACODEC_ADC_ANA_CON7[5:4], to set microphone as input of the ADC left channel1.
2. Configure the register ACODEC_ADC_ANA_CON0[7] to 1, to end the mute station of the ADC right channel.
Configure the register ACODEC_ADC_ANA_CON0[3] to 1, to end the mute station of the ADC left channel.
3. Configure the register ACODEC_ADC_ANA_CON6[0] to 1, to enable the current source of audio.
4. Configure the register ACODEC_ADC_ANA_CON0[4] to 1, to enable the reference voltage buffer in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON0[0] to 1, to enable the reference voltage buffer in ADC left channel.
5. Configure the register ACODEC_ADC_ANA_CON0[5] to 1, to enable the MIC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON0[1] to 1, to enable the MIC module in ADC

left channel.
6. Configure the register ACODEC_ADC_ANA_CON2[4] to 1, to enable the ALC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON2[0] to 1, to enable the ALC module in ADC right channel.
7. Configure the register ACODEC_ADC_ANA_CON5[4] to 1, to enable the clock module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON5[0] to 1, to enable the clock module in ADC left channel.
8. Configure the register ACODEC_ADC_ANA_CON5[5] to 1, to enable the ADC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON5[1] to 1, to enable the ADC module in ADC left channel.
9. Configure the register ACODEC_ADC_ANA_CON5[6] to 1, to end the initialization of the ADC module.
Configure the register ACODEC_ADC_ANA_CON5[2] to 1, to end the initialization of the ADC module.
10. Configure the register ACODEC_ADC_ANA_CON2[5] to 1, to end the initialization of the ALC right module.
Configure the register ACODEC_ADC_ANA_CON2[1] to 1, to end the initialization of the ALC left module.
11. Configure the register ACODEC_ADC_ANA_CON0[6] to 1, to end the initialization of the MIC right module.
Configure the register ACODEC_ADC_ANA_CON0[2] to 1, to end the initialization of the MIC right module.
12. Configure the register ACODEC_ADC_ANA_CON1[5:4], to select the gain of the MIC right module.
Configure the register ACODEC_ADC_ANA_CON1[1:0], to select the gain of the MIC left module.
13. Configure the register ACODEC_ADC_ANA_CON3[4:0], to select the gain of the ALC left module.
Configure the register ACODEC_ADC_ANA_CON4[4:0], to select the gain of the ALC right module.
14. Configure the register ACODEC_ADC_ANA_CON2[6] to 1, to enable the zero-crossing detection function in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON2[2] to 1, to enable the zero-crossing detection function in ADC left channel
Note1. Please ignore the step1 for enabling ADC3, ADC4, ADC5, ADC6,ADC7, and ADC8
Note2. Please ignore the step14 if the AGC function is closed
Note3. When configure ACODEC_ADC_ANA_CONx, user should add a offset address to correspond channel:
add offset 0x000 for left_0(ADC1) and right_0(ADC2) channel;
add offset 0x040 for left_1(ADC3) and right_1(ADC4) channel;
add offset 0x080 for left_2(ADC5) and right_2(ADC6) channel;
add offset 0x0c0 for left_3(ADC7) and right_3(ADC8) channel.
Note4. When enable ADC for exiting lower power mode (ADC is configured to disable and Audio Codec is still power on), step0,1,2,12,13 can be ignored. Step4 and step5 can be merged as one step, step7,8 and 9 can be merged as one step.

8.6.10 Disable ADC

0. Keep ADC channel work and stop recording.
1. Configure the register ACODEC_ADC_ANA_CON2[6] to 0, to disable the zero-crossing detection function in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON2[2] to 0, to disable the zero-crossing detection function in ADC left channel.
2. Configure the register ACODEC_ADC_ANA_CON5[5] to 0, to disable the ADC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON5[1] to 0, to disable the ADC module in ADC

left channel.
3. Configure the register ACODEC_ADC_ANA_CON5[4] to 0, to disable the clock module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON5[0] to 0, to disable the clock module in ADC left channel.
4. Configure the register ACODEC_ADC_ANA_CON2[4] to 0, to disable the ALC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON2[0] to 0, to disable the ALC module in ADC left channel.
5. Configure the register ACODEC_ADC_ANA_CON0[5] to 0, to disable the MIC module in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON0[1] to 0, to disable the MIC module in ADC left channel
6. Configure the register ACODEC_ADC_ANA_CON0[4] to 0, to disable the reference voltage buffer in ADC right channel.
Configure the register ACODEC_ADC_ANA_CON0[0] to 0, to disable the reference voltage buffer in ADC left channel.
7. Configure the register ACODEC_ADC_ANA_CON6[0] to 0, to disable the current source of the ADC.
8. Configure the register ACODEC_ADC_ANA_CON5[6] to 0, to begin the initialization of the ADC right module.
Configure the register ACODEC_ADC_ANA_CON5[2] to 0, to begin the initialization of the ADC left module.
9. Configure the register ACODEC_ADC_ANA_CON2[5] to 0, to begin the initialization of the ALC right module.
Configure the register ACODEC_ADC_ANA_CON2[1] to 0, to begin the initialization of the ALC left module.
10. Configure the register ACODEC_ADC_ANA_CON0[6] to 0, to begin the initialization of the MIC right module.
Configure the register ACODEC_ADC_ANA_CON0[2] to 0, to begin the initialization of the MIC left module

8.6.11 Enable ALC

1. Set he max level and min level of the ALC need to control. The AGC register ACODEC_ALC_DIG_CON5 and ACODEC_ALC_DIG_CON6 control the max level of the ALC. max_level={ACODEC_ALC_DIG_CON6, ACODEC_ALC_DIG_CON5}; The relationship between the max_level and the signal amplitude(dB) is equal to: amplitude(dB)=20log(max_level/0x7fff), 0x7fff means the full scale amplitude. The ACODEC_ALC_DIG_CON7 and ACODEC_ALC_DIG_CON8 control the min level of the ALC. min_level={ACODEC_ALC_DIG_CON8, ACODEC_ALC_DIG_CON7}; The relationship between the min_level and the signal amplitude(dB) is equal to: amplitude(dB)=20log(min_level/0x7fff), 0x7fff means the full scale amplitude
2. Set ACODEC_ALC_DIG_CON4[2:0] according to the sample rate
3. Set ACODEC_ALC_DIG_CON9[6] to 0x1, to enable the ALC module
4.Set ACODEC_ADC_ANA_CON11[1:0], (ACODEC_ADC_ANA_CON11+0x40)[1:0], (ACODEC_ADC_ANA_CON11+0x80)[1:0] and (ACODEC_ADC_ANA_CON11+0xc0)[1:0] to 0x3, to enable the ALC module to control the gain of PGA.
5.Observe the current ALC output gain by reading ACODEC_ALC_DIG_CON12[4:0].
Note1. expect the step4, other step should add a offset address to correspond channel as follow:
add offset 0x000 for left channel ALC1;
add offset 0x040 for right channel ALC2;
add offset 0x0c0 for left channel ALC3;
add offset 0x100 for right channel ALC4;
add offset 0x180 for left channel ALC5;
add offset 0x1c0 for right channel ALC6;
add offset 0x240 for left channel ALC7;
add offset 0x280 for right channel ALC8.

8.6.12 Disable ALC

1. Set ACODEC_ALC_DIG_CON9[6] to 0x0, to disable the ALC module, then the ALC output gain will keep to the last vale.
2. Set ACODEC_ADC_ANA_CON11[1:0], (ACODEC_ADC_ANA_CON11+0x40)[1:0], (ACODEC_ADC_ANA_CON11+0x80)[1:0] and (ACODEC_ADC_ANA_CON11+0xc0)[1:0] to 0x0, to disable the ALC module to control the gain of PGA.

Note1. expect the step2, other step should add a offset address as follow:
to correspond channel:
add offset 0x000 for left channel ALC1;
add offset 0x040 for right channel ALC2;
add offset 0x0c0 for left channel ALC3;
add offset 0x100 for right channel ALC4;
add offset 0x180 for left channel ALC5;
add offset 0x1c0 for right channel ALC6;
add offset 0x240 for left channel ALC7;
add offset 0x280 for right channel ALC8.

8.6.13 Enable Headset Insert

1. Set ACODEC_DAC_ANA_CON0[1] to 0x1, to enable the headset insert detection

Note1. When the voltage of PAD HPDET> 8*AVDD/9, the output value of the pin_hpdet will be set to 0x1 and assert a interrupt.

8.6.14 Disable Headset Insert

1. Set ACODEC_DAC_ANA_CON0[1] to 0x0, to disable the headset insert detection.

8.6.15 LOW POWER

This section describes the application of the low power in the Audio Codec. The following will introduce the two application mode of the low power.

light power down

The DAC take the gated clock technique which can reduce power. You can enable or disable the gated clock by configure the register ACODEC_GLB_CON[5:4]. When the register ACODEC_GLB_CON[5] and ACODEC_GLB_CON[4] set 1, the Audio Codec clock which is in the ADC path and DAC path will be cut off, which make the digital part in the low power state. If you want the Audio Codec work in the normal mode immediately, please configure the ACODEC_GLB_CON[5:4] to 2’b00.

deep power down

If you want the Audio Codec to go to the deep power down mode, please refer to following steps.

1. Close ALC according to the section disable ALC
2. Close DAC and ADC according to the section disable DAC and disable DAC.
3. Disable the MICBIAS according to the section disable MICBIAS.
4. Power off the codec according to the section power off.

However, once the Audio Codec enters this mode, if you want to return to the normal mode, please refer to the following steps.

1. Power up the codec according to the section power on.
2. Enable the MICBIAS according to the section enable MICBIAS.
3. Enable the DAC and ADC according to the section enable DAC and enable ADC.
4. Enable ALC according to the section enable ALC.

Chapter 9 DMA Controller (DMAC)

9.1 Overview

This device supports 2 Direct Memory Access (DMA) Controllers, DMAC0 and DMAC1. Both of these two DMAC support transfers between memory and memory, peripheral and memory. DMAC is under Non-secure state after reset, and the secure state can be changed by configuring SGRF module.

DMAC0 supports the following features:

- Supports Trustzone technology
- Supports 12 peripheral request
- Up to 64bits data size
- 6 channel at the same time
- Up to burst 16
- 12 interrupts output and 1 abort output
- Supports 32 MFIFO depth

Following table shows the DMAC0 request mapping scheme.

Table 9-1 DMAC0 Request Mapping Table

Req number	Source	Polarity
0	SPI0 tx	High level
1	SPI0 rx	High level
2	SPI1 tx	High level
3	SPI1 rx	High level
4	UART0 tx	High level
5	UART0 rx	High level
6	UART1 tx	High level
7	UART1 rx	High level
8	UART2 tx	High level
9	UART2 rx	High level
10	UART3 tx	High level
11	UART3 rx	High level

DMAC1 supports the following features:

- Supports Trustzone technology
- 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 128 MFIFO depth

Following table shows the DMAC1 request mapping scheme.

Table 9-2 DMAC1 Request Mapping Table

Req number	Source	Polarity
0	I2S/TDM/PCM_8CH_0 tx	High level
1	I2S/TDM/PCM_8CH_0 rx	High level
2	I2S/TDM/PCM_8CH_1 tx	High level
3	I2S/TDM/PCM_8CH_1 rx	High level
4	I2S/TDM/PCM_8CH_2 tx	High level
5	I2S/TDM/PCM_8CH_2 rx	High level
6	I2S/TDM/PCM_8CH_3 tx	High level
7	I2S/TDM/PCM_8CH_3 rx	High level
8	I2S/PCM_2CH_0 tx	High level
9	I2S/PCM_2CH_0 rx	High level
10	I2S/PCM_2CH_1 tx	High level
11	I2S/PCM_2CH_1 rx	High level

Req number	Source	Polarity
12	PDM_8CH rx	High level
13	SPDIF_8CH_TX tx	High level
14	SPDIF_8CH_RX rx	High level
15	PWM_4CH rx	High level
16	SPI2 tx	High level
17	SPI2 rx	High level
18	UART4 tx	High level
19	UART4 rx	High level

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

9.2 Block Diagram

Following figure shows the block diagram of DMAC.

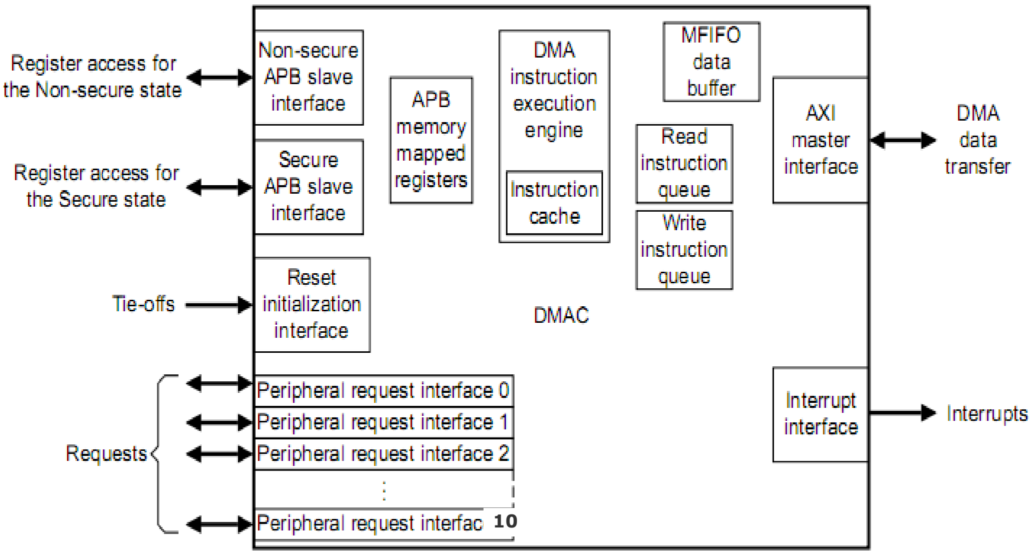

Register access for the Non-secure state
Non-secure APB slave interface
APB memory mapped registers
DMA instruction execution engine
Instruction cache
MFIFO data buffer
Read instruction queue
Write instruction queue
AXI master interface
DMA data transfer
Register access for the Secure state
Secure APB slave interface
Tie-offs
Reset initialization interface
DMAC
Requests
Peripheral request interface 0
Peripheral request interface 1
Peripheral request interface 2
Peripheral request interfac 10
Interrupt interface
Interrupts

Fig. 9-1 Block diagram of DMAC

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

9.3 Function Description

9.3.1 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.

DMAC0 supports 6 channels and DMAC1 supports 8 channels, each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an

instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.
When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete.
When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

9.3.2 Operating states

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

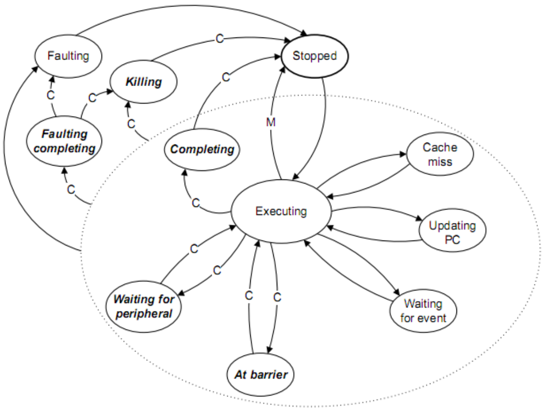


Fig. 9-2 DMAC operation states

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

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

9.4 Register Description

9.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DMA_DSR	0x0000	W	0x00000000	DMA Manager Status Register
DMA_DPC	0x0004	W	0x00000000	DMA Program Counter Register
DMA_INTEN	0x0020	W	0x00000000	Interrupt Enable Register
DMA_EVENT_RIS	0x0024	W	0x00000000	Event-Interrupt Raw Status Register
DMA_INTMIS	0x0028	W	0x00000000	Interrupt Status Register
DMA_INTCLR	0x002c	W	0x00000000	Interrupt Clear Register

Name	Offset	Size	Reset Value	Description
DMA_FSRD	0x0030	W	0x00000000	Fault Status DMA Manager Register
DMA_FSRC	0x0034	W	0x00000000	Fault Status DMA Channel Register
DMA_FTRD	0x0038	W	0x00000000	Fault Type DMA Manager Register
DMA_FTR0	0x0040	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR1	0x0044	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR2	0x0048	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR3	0x004c	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR4	0x0050	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR5	0x0054	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR6	0x0058	W	0x00000000	Fault Type DMA Channel Register
DMA_FTR7	0x005c	W	0x00000000	Fault Type DMA Channel Register
DMA_CSR0	0x0100	W	0x00000000	Channel Status Registers
DMA_CPC0	0x0104	W	0x00000000	Channel Program Counter Registers
DMA_CSR1	0x0108	W	0x00000000	Channel Status Registers
DMA_CPC1	0x010c	W	0x00000000	Channel Program Counter Registers
DMA_CSR2	0x0110	W	0x00000000	Channel Status Registers
DMA_CPC2	0x0114	W	0x00000000	Channel Program Counter Registers
DMA_CSR3	0x0118	W	0x00000000	Channel Status Registers
DMA_CPC3	0x011c	W	0x00000000	Channel Program Counter Registers
DMA_CSR4	0x0120	W	0x00000000	Channel Status Registers
DMA_CPC4	0x0124	W	0x00000000	Channel Program Counter Registers
DMA_CSR5	0x0128	W	0x00000000	Channel Status Registers
DMA_CPC5	0x012c	W	0x00000000	Channel Program Counter Registers
DMA_CSR6	0x0130	W	0x00000000	Channel Status Registers
DMA_CPC6	0x0134	W	0x00000000	Channel Program Counter Registers
DMA_CSR7	0x0138	W	0x00000000	Channel Status Registers
DMA_CPC7	0x013c	W	0x00000000	Channel Program Counter Registers
DMA_SAR0	0x0400	W	0x00000000	Source Address Registers
DMA_DAR0	0x0404	W	0x00000000	Destination Address Registers
DMA_CCR0	0x0408	W	0x00000000	Channel Control Registers
DMA_LC0_0	0x040c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_0	0x0410	W	0x00000000	Loop Counter 1 Registers
DMA_SAR1	0x0420	W	0x00000000	Source Address Registers

Name	Offset	Size	Reset Value	Description
DMA_DAR1	0x0424	W	0x00000000	Destination Address Registers
DMA_CCR1	0x0428	W	0x00000000	Channel Control Registers
DMA_LC0_1	0x042c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_1	0x0430	W	0x00000000	Loop Counter 1 Registers
DMA_SAR2	0x0440	W	0x00000000	Source Address Registers
DMA_DAR2	0x0444	W	0x00000000	Destination Address Registers
DMA_CCR2	0x0448	W	0x00000000	Channel Control Registers
DMA_LC0_2	0x044c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_2	0x0450	W	0x00000000	Loop Counter 1 Registers
DMA_SAR3	0x0460	W	0x00000000	Source Address Registers
DMA_DAR3	0x0464	W	0x00000000	Destination Address Registers
DMA_CCR3	0x0468	W	0x00000000	Channel Control Registers
DMA_LC0_3	0x046c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_3	0x0470	W	0x00000000	Loop Counter 1 Registers
DMA_SAR4	0x0480	W	0x00000000	Source Address Registers
DMA_DAR4	0x0484	W	0x00000000	Destination Address Registers
DMA_CCR4	0x0488	W	0x00000000	Channel Control Registers
DMA_LC0_4	0x048c	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_4	0x0490	W	0x00000000	Loop Counter 1 Registers
DMA_SAR5	0x04a0	W	0x00000000	Source Address Registers
DMA_DAR5	0x04a4	W	0x00000000	Destination Address Registers
DMA_CCR5	0x04a8	W	0x00000000	Channel Control Registers
DMA_LC0_5	0x04ac	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_5	0x04b0	W	0x00000000	Loop Counter 1 Registers
DMA_SAR6	0x04c0	W	0x00000000	Source Address Registers
DMA_DAR6	0x04c4	W	0x00000000	Destination Address Registers
DMA_CCR6	0x04c8	W	0x00000000	Channel Control Registers
DMA_LC0_6	0x04cc	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_6	0x04d0	W	0x00000000	Loop Counter 1 Registers
DMA_SAR7	0x04e0	W	0x00000000	Source Address Registers
DMA_DAR7	0x04e4	W	0x00000000	Destination Address Registers
DMA_CCR7	0x04e8	W	0x00000000	Channel Control Registers
DMA_LC0_7	0x04ec	W	0x00000000	Loop Counter 0 Registers
DMA_LC1_7	0x04f0	W	0x00000000	Loop Counter 1 Registers
DMA_DBGSTATUS	0x0d00	W	0x00000000	Debug Status Register
DMA_DBGCMD	0x0d04	W	0x00000000	Debug Command Register
DMA_DBGINST0	0x0d08	W	0x00000000	Debug Instruction-0 Register
DMA_DBGINST1	0x0d0c	W	0x00000000	Debug Instruction-1 Register
DMA_CR0	0x0e00	W	0x00047051	Configuration Register 0
DMA_CR1	0x0e04	W	0x00000057	Configuration Register 1
DMA_CR2	0x0e08	W	0x00000000	Configuration Register 2
DMA_CR3	0x0e0c	W	0x00000000	Configuration Register 3

Name	Offset	Size	Reset Value	Description
DMA_CR4	0x0e10	W	0x00000006	Configuration Register 4
DMA_CRDn	0x0e14	W	0x02094733	Configuration Register
DMA_WD	0x0e80	W	0x00000000	DMA Watchdog Register

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

9.4.2 Detail Register Description

DMA_DSR
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	dns 1'b0: DMA manager operates in the Secure state 1'b1: DMA manager operates in the Non-secure state
8:4	RO	0x00	wakeup_event 5'b00000: event[0] 5'b00001: event[1] 5'b00010: event[2] … 5'b11111: event[31]
3:0	RO	0x0	dma_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101-4'b1110: reserved 4'b1111: Faulting

DMA_DPC
Address: Operational Base + offset (0x0004)

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

DMA_INTEN
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	event_irq_select Bit [N] 1'b0: If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request 1'b1: If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interrupt request

DMA_EVENT_RIS

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dmasev_active Bit [N] 1'b0: Event N is inactive or irq[N] is LOW 1'b1: Event N is active or irq[N] is HIGH

DMA_INTMIS

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	irq_status Bit [N] 1'b0: Interrupt N is inactive and therefore irq[N] is LOW 1'b1: Interrupt N is active and therefore irq[N] is HIGH

DMA_INTCLR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	irq_clr Bit [N] 1'b0: The status of irq[N] does not change 1'b1: The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change

DMA_FSRD

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	fs_mgr 1'b0: the DMA manager thread is not in the Faulting state 1'b1: the DMA manager thread is in the Faulting state

DMA_FSRC

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	fault_status Bit [N] 1'b0: No fault is present on DMA channel N. 1'b1: DMA channel N is in the Faulting or Faulting completing state

DMA_FTRD

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface
29:17	RO	0x0	reserved
16	RO	0x0	instr_fetch_err performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response
15:6	RO	0x0	reserved
5	RO	0x0	mgr_evnt_err 1'b0: the DMA manager has appropriate security to execute DMAWFE or DMASEV 1'b1: a DMA manager thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt
4	RO	0x0	dmago_err 1'b0: the DMA manager has appropriate security to execute DMAGO 1'b1: a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state
3:2	RO	0x0	reserved
1	RO	0x0	operand_invalid the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand

Bit	Attr	Reset Value	Description
0	RW	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction

DMA_FTR0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR4

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR5

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR6

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_FTR7

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31	RO	0x0	lockup_err 1'b0: DMA channel has adequate resources 1'b1: DMA channel has locked-up because of insufficient resources This fault is an imprecise abort.
30	RO	0x0	dbg_instr memory or from the debug interface: 1'b0: instruction that generated an abort was read from system memory 1'b1: instruction that generated an abort was read from the debug interface This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	RO	0x0	reserved
18	RO	0x0	data_read_err thread performs a data read: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
17	RO	0x0	data_write_err thread performs a data write: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is an imprecise abort.
16	RO	0x0	instr_fetch_err thread performs an instruction fetch: 1'b0: OKAY response 1'b1: EXOKAY, SLVERR, or DECERR response This fault is a precise abort.
15:14	RO	0x0	reserved
13	RO	0x0	st_data_unavailable 1'b0: MFIFO contains all the data to enable the DMAST to complete 1'b1: previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete This fault is a precise abort.

Bit	Attr	Reset Value	Description
12	RO	0x0	mfifo_err DMALD 1'b0: MFIFO contains sufficient space 1'b1: MFIFO is too small to hold the data that DMALD requires DMAST 1'b0: MFIFO contains sufficient data 1'b1: MFIFO is too small to store the data to enable DMAST to complete This fault is an imprecise abort.
11:8	RO	0x0	reserved
7	RO	0x0	ch_rdwr_err to perform a secure read or secure write: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write This fault is a precise abort.
6	RO	0x0	ch_periph_err DMASTP, or DMAFLUSHP with inappropriate security permissions: 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFP to wait for a secure peripheral o DMALDP or DMASTP to notify a secure peripheral o DMAFLUSHP to flush a secure peripheral This fault is a precise abort.
5	RO	0x0	ch_event_err 1'b0: a DMA channel thread in the Non-secure state is not violating the security permissions 1'b1: a DMA channel thread in the Non-secure state attempted to execute either: o DMAWFE to wait for a secure event o DMASEV to create a secure event or secure interrupt This fault is a precise abort.
4:2	RO	0x0	reserved
1	RO	0x0	operand_invalid valid for the configuration of the DMAC: 1'b0: valid operand 1'b1: invalid operand This fault is a precise abort.

Bit	Attr	Reset Value	Description
0	RO	0x0	undef_instr 1'b0: defined instruction 1'b1: undefined instruction This fault is a precise abort.

DMA_CSR0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC0

Address: Operational Base + offset (0x0104)

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

DMA_CSR1

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC1

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 1 thread

DMA_CSR2
Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC2
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 2 thread

DMA_CSR3

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC3

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 3 thread

DMA_CSR4

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC4

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 4 thread

DMA_CSR5

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC5

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 5 thread

DMA_CSR6

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC6

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 6 thread

DMA_CSR7

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	cns 1'b0: DMA channel operates in the Secure state 1'b1: DMA channel operates in the Non-secure state
20:16	RO	0x0	reserved
15	RO	0x0	dmawfp_periph 1'b0: DMAWFP executed with the periph operand not set 1'b1: DMAWFP executed with the periph operand set
14	RO	0x0	dmawfp_b_ns 1'b0: DMAWFP executed with the single operand set 1'b1: DMAWFP executed with the burst operand set
13:9	RO	0x0	reserved
8:4	RO	0x00	wakeup_number indicate the event or peripheral number that the channel is waiting for: 5'b00000: DMA channel is waiting for event, or peripheral, 0 5'b00001: DMA channel is waiting for event, or peripheral, 1 5'b00010: DMA channel is waiting for event, or peripheral, 2 … 5'b11111: DMA channel is waiting for event, or peripheral, 31
3:0	RO	0x0	channel_status 4'b0000: Stopped 4'b0001: Executing 4'b0010: Cache miss 4'b0011: Updating PC 4'b0100: Waiting for event 4'b0101: At barrier 4'b0110: reserved 4'b0111: Waiting for peripheral 4'b1000: Killing 4'b1001: Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMA_CPC7

Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pc_chnl Program counter for the DMA channel 7 thread

DMA_SAR0

Address: Operational Base + offset (0x0400)

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

DMA_DAR0

Address: Operational Base + offset (0x0404)

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

DMA_CCR0

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_0

Address: Operational Base + offset (0x040c)

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

DMA_LC1_0

Address: Operational Base + offset (0x0410)

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

DMA_SAR1

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 1

DMA_DAR1

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 1

DMA_CCR1

Address: Operational Base + offset (0x0428)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.

Bit	Attr	Reset Value	Description
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.

Bit	Attr	Reset Value	Description
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_1
Address: Operational Base + offset (0x042c)

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

DMA_LC1_1
Address: Operational Base + offset (0x0430)

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

DMA_SAR2
Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 2

DMA_DAR2
Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 2

DMA_CCR2
Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_2

Address: Operational Base + offset (0x044c)

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

DMA_LC1_2

Address: Operational Base + offset (0x0450)

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

DMA_SAR3

Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 3

DMA_DAR3

Address: Operational Base + offset (0x0464)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 3

DMA_CCR3

Address: Operational Base + offset (0x0468)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_3

Address: Operational Base + offset (0x046c)

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

DMA_LC1_3

Address: Operational Base + offset (0x0470)

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

DMA_SAR4

Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 4

DMA_DAR4

Address: Operational Base + offset (0x0484)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 4

DMA_CCR4

Address: Operational Base + offset (0x0488)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH

Bit	Attr	Reset Value	Description
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH

Bit	Attr	Reset Value	Description
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_4

Address: Operational Base + offset (0x048c)

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

DMA_LC1_4

Address: Operational Base + offset (0x0490)

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

DMA_SAR5

Address: Operational Base + offset (0x04a0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 5

DMA_DAR5

Address: Operational Base + offset (0x04a4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 5

DMA_CCR5

Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.

Bit	Attr	Reset Value	Description
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.

Bit	Attr	Reset Value	Description
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_5
Address: Operational Base + offset (0x04ac)

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

DMA_LC1_5
Address: Operational Base + offset (0x04b0)

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

DMA_SAR6
Address: Operational Base + offset (0x04c0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 6

DMA_DAR6
Address: Operational Base + offset (0x04c4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 6

DMA_CCR6
Address: Operational Base + offset (0x04c8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.

Bit	Attr	Reset Value	Description
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_6

Address: Operational Base + offset (0x04cc)

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

DMA_LC1_6

Address: Operational Base + offset (0x04d0)

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

DMA_SAR7

Address: Operational Base + offset (0x04e0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	src_addr Address of the source data for DMA channel 7

DMA_DAR7

Address: Operational Base + offset (0x04e4)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dst_addr Address of the Destinationdata for DMA channel 7

DMA_CCR7

Address: Operational Base + offset (0x04e8)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:25	RO	0x0	dst_cache_ctrl Bit [27] 1'b0: AWCACHE[3] is LOW 1'b1: AWCACHE[3] is HIGH Bit [26] 1'b0: AWCACHE[1] is LOW 1'b1: AWCACHE[1] is HIGH Bit [25] 1'b0: AWCACHE[0] is LOW 1'b1: AWCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
24:22	RO	0x0	dst_prot_ctrl Bit [24] 1'b0: AWPROT[2] is LOW 1'b1: AWPROT[2] is HIGH Bit [23] 1'b0: AWPROT[1] is LOW 1'b1: AWPROT[1] is HIGH Bit [22] 1'b0: AWPROT[0] is LOW 1'b1: AWPROT[0] is HIGH
21:18	RO	0x0	dst_burst_len the destination data: 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
17:15	RO	0x0	dst_burst_size 3'b000: writes 1 byte per beat 3'b001: writes 2 bytes per beat 3'b010: writes 4 bytes per beat 3'b011: writes 8 bytes per beat 3'b100: writes 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	RO	0x0	dst_inc 1'b0: Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	RO	0x0	src_cache_ctrl Bit [13] 1'b0: ARCACHE[2] is LOW 1'b1: ARCACHE[2] is HIGH Bit [12] 1'b0: ARCACHE[1] is LOW 1'b1: ARCACHE[1] is HIGH Bit [11] 1'b0: ARCACHE[0] is LOW 1'b1: ARCACHE[0] is HIGH

Bit	Attr	Reset Value	Description
10:8	RO	0x0	src_prot_ctrl Bit [10] 1'b0: ARPROT[2] is LOW 1'b1: ARPROT[2] is HIGH Bit [9] 1'b0: ARPROT[1] is LOW 1'b1: ARPROT[1] is HIGH Bit [8] 1'b0: ARPROT[0] is LOW 1'b1: ARPROT[0] is HIGH
7:4	RO	0x0	src_burst_len 4'b0000: 1 data transfer 4'b0001: 2 data transfers 4'b0010: 3 data transfers … 4'b1111: 16 data transfers The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
3:1	RO	0x0	src_burst_size 3'b000: reads 1 byte per beat 3'b001: reads 2 bytes per beat 3'b010: reads 4 bytes per beat 3'b011: reads 8 bytes per beat 3'b100: reads 16 bytes per beat 3'b101-3'b111: reserved The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size.
0	RO	0x0	src_inc 1'b0: Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1'b1: Incrementing-address burst. The DMAC signals ARBURST[0] HIGH.

DMA_LC0_7

Address: Operational Base + offset (0x04ec)

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

DMA_LC1_7

Address: Operational Base + offset (0x04f0)

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

DMA_DBGSTATUS

Address: Operational Base + offset (0x0d00)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	dbgstatus 1'b0: idle 1'b1: busy

DMA_DBGCMD

Address: Operational Base + offset (0x0d04)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	WO	0x0	dbgcmd 2'b00: execute the instruction that the DBGINST [1:0] Registers contain 2'b01: reserved 2'b10: reserved 2'b11: reserved

DMA_DBGINST0

Address: Operational Base + offset (0x0d08)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	instruction_byte1 Instruction byte 1
23:16	WO	0x00	instruction_byte0 Instruction byte 0
15:11	RO	0x0	reserved
10:8	WO	0x0	channel_number 3'b000: DMA channel 0 3'b001: DMA channel 1 3'b010: DMA channel 2 … 3'b111: DMA channel 7
7:1	RO	0x0	reserved
0	WO	0x0	debug_thread 1'b0: DMA manager thread 1'b1: DMA channel

DMA_DBGINST1

Address: Operational Base + offset (0x0d0c)

Bit	Attr	Reset Value	Description
31:24	WO	0x00	instruction_byte5 Instruction byte 5
23:16	WO	0x00	instruction_byte4 Instruction byte 4
15:8	WO	0x00	instruction_byte3 Instruction byte 3
7:0	WO	0x00	instruction_byte2 Instruction byte 2

DMA_CR0

Address: Operational Base + offset (0x0e00)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:17	RO	0x02	num_events 5'b00000: 1 interrupt output, irq[0] 5'b00001: 2 interrupt outputs, irq[1:0] 5'b00010: 3 interrupt outputs, irq[2:0] … 5'b11111: 32 interrupt outputs, irq[31:0]
16:12	RO	0x07	num_periph_req 5'b00000: 1 peripheral request interface 5'b00001: 2 peripheral request interfaces 5'b00010: 3 peripheral request interfaces … 5'b11111: 32 peripheral request interfaces
11:7	RO	0x0	reserved
6:4	RO	0x5	num_chnls 3'b000: 1 DMA channel 3'b001: 2 DMA channels 3'b010: 3 DMA channels … 3'b111: 8 DMA channels
3	RO	0x0	reserved
2	RO	0x0	mgr_ns_at_rst 1'b0: boot_manager_ns was LOW 1'b1: boot_manager_ns was HIGH
1	RO	0x0	boot_en 1'b0: boot_from_pc was LOW 1'b1: boot_from_pc was HIGH
0	RO	0x1	periph_req 1'b0: the DMAC does not provide a peripheral request interface 1'b1: the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies

DMA_CR1

Address: Operational Base + offset (0x0e04)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RO	0x5	num_i_cache_lines 4'b0000: 1 i-cache line 4'b0001: 2 i-cache lines 4'b0010: 3 i-cache lines … 4'b1111: 16 i-cache lines
3	RO	0x0	reserved
2:0	RO	0x7	i_cache_len 3'b000-3'b001: reserved 3'b010: 4 bytes 3'b011: 8 bytes 3'b100: 16 bytes 3'b101: 32 bytes 3'b110-3'b111: reserved

DMA_CR2

Address: Operational Base + offset (0x0e08)

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

DMA_CR3

Address: Operational Base + offset (0x0e0c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ins Bit [N] 1'b0: Assigns event<N> or irq[N] to the Secure state 1'b1: Assigns event<N> or irq[N] to the Non-secure state

DMA_CR4

Address: Operational Base + offset (0x0e10)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000006	pns Bit [N] 1'b0: Assigns peripheral request interface N to the Secure state 1'b1: Assigns peripheral request interface N to the Non-secure state

DMA_CRDn

Address: Operational Base + offset (0x0e14)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:20	RO	0x020	data_buffer_dep 10'b000000000: 1 line 10'b000000001: 2 lines … 10'b111111111: 1024 lines
19:16	RO	0x9	rd_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
15	RO	0x0	reserved
14:12	RO	0x4	rd_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
11:8	RO	0x7	wr_q_dep 4'b0000: 1 line 4'b0001: 2 lines … 4'b1111: 16 lines
7	RO	0x0	reserved
6:4	RO	0x3	wr_cap 3'b000: 1 3'b001: 2 … 3'b111: 8
3	RO	0x0	reserved
2:0	RO	0x3	data_width 3'b000: reserved 3'b001: reserved 3'b010: 32-bit 3'b011: 64-bit 3'b100: 128-bit 3'b101-3'b111: reserved

DMA_WD

Address: Operational Base + offset (0x0e80)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	wd_irq_only 1'b0: the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1'b1: the DMAC sets irq_abort HIGH

9.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

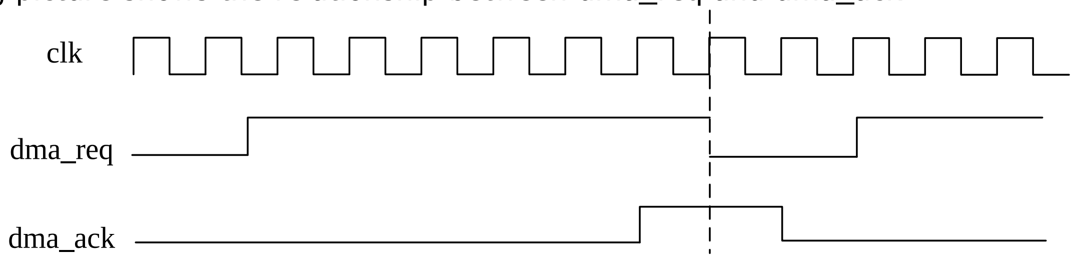


Fig. 9-3 DMAC request and acknowledge timing

9.6 Interface Description

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

Table 9-3 DMAC0 boot interface

Interface	Reset value	Control source
boot_addr	0x0	SGRF_DMAC_CON6[19:0]
boot_from_pc	0x0	SGRF_DMAC_CON7[0]
boot_manager_ns	0x1	SGRF_DMAC_CON7[1]
boot_irq_ns	0xfff	SGRF_DMAC_CON4[11:0]
boot_periph_ns	0xfff	SGRF_DMAC_CON5[11:0]

Table 9-4 DMAC1 boot interface

Interface	Reset value	Control source
boot_addr	0x0	SGRF_DMAC_CON10[19:0]
boot_from_pc	0x0	SGRF_DMAC_CON7[2]
boot_manager_ns	0x1	SGRF_DMAC_CON7[3]
boot_irq_ns	0xffff	SGRF_DMAC_CON8[15:0]
boot_periph_ns	0xfffff	SGRF_DMAC_CON9[19:0]

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.

9.7 Application Notes

9.7.1 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.

9.7.2 Security usage

DMA manager thread is in the secure state
If the DNS bit is 0, the DMA manager thread operates in the secure state and it only performs secure instruction fetches. When a DMA manager thread in the secure state processes:
DMAGO
It uses the status of the ns bit, to set the security state of the DMA channel thread by writing to the CNS bit for that channel.
DMAWFE
It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.
DMASEV
It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.
DMA manager thread is in the Non-secure state
If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:
DMAGO
The DMAC uses the status of the ns bit, to control if it starts a DMA channel
thread. If:
ns = 0
The DMAC does not start a DMA channel thread and instead it:
1. Executes a NOP.

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

ns = 1

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

DMAWFE

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

INS = 0

The event is in the Secure state. The DMAC:

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

INS = 1

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

DMASEV

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

INS = 0

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

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

INS = 1

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

DMA channel thread is in the secure state

When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.

When a DMA channel thread in the secure state processes the following instructions:

DMAWFE

The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMASEV

The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMAWFP

The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMALDP, DMASTP

The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMAFLUSHP

The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses

DMA channel thread is in the Non-secure state

When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure

state and it only performs non-secure instruction fetches.
When a DMA channel thread in the Non-secure state processes the following instructions:

DMAWFE

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

INS = 0

The event is in the Secure state. The DMAC:

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

INS = 1

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

DMASEV

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

INS = 0

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

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

INS = 1

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

DMAWFP

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

PNS = 0

The peripheral is in the Secure state. The DMAC:

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

PNS = 1

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

DMALDP, DMASTP

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

PNS = 0

The peripheral is in the secure state. The DMAC:

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

PNS = 1

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

DMAFLUSHP

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

PNS = 0
The peripheral is in the secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.
PNS = 1
The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.
When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:
1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

9.7.3 Programming restrictions

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

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

Unaligned write

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

Endian swap size restrictions
If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.
Updating DMA channel control registers during a DMA cycle restrictions
Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.
Resource sharing between DMA channels
DMA channel programs share the MFIFO data storage resource. You must not start a set of concurrently running DMA channel programs with a resource requirement that exceeds the configured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

9.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.
Data corruption might occur if all of the following are true:
1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is splited across two lines in the data buffer (see Splitting data, below).

3. There is one idle cycle between the two read data beats.
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data

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

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

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

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

Table 9-5 Source size in CCRn

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

9.7.5 Interrupt shares between channel

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

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

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

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

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

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

9.7.6 Instruction sets

Table 9-6 DMAC Instruction sets

Mnemonic	Instruction	Thread usage
DMAADDH	Add Halfword	C
DMAEND	End	M/C
DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C

Mnemonic	Instruction	Thread usage
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C
DMAMOV	Move	C
DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wait For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

Notes: Thread usage: C=DMA channel, *M=DMA manager*

9.7.7 Assembler directives

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

DMAADNH

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

Table 9-7 DMAC instruction encoding

Imm[15:8]	Imm[7:0]	0	1	0	1	1	1	ra	0

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>
where:
<address_register>
    Selects the address register to use. It must be either:
    SAR
        SARn Register and sets ra to 0.
    DAR
        DARn Register and sets ra to 1.
<16-bit immediate>
    The immediate value to be added to the <address_register>.

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

Chapter 10 Generic Interrupt Controller (GIC)

10.1 Overview

There is a generic interrupt controller(GIC400) in RK3308 which generates physical interrupts to Cortex-A35. It has two interfaces, the distributor interface connects to the interrupt source, and the CPU interface connects to Cortex-A35. The details of CPU interface connectivity are shown in the following table.

Table 10-1 CPU interface connectivity

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

It supports the following features:

- Supports 128 hardware interrupt inputs
- Masking of any interrupts
- Prioritization of interrupts
- Distribution of the interrupts to the target Cortex-A35 processor(s)
- Generation of interrupts by software
- Supports Security Extensions

10.2 Block Diagram

The generic interrupt controller comprises with:

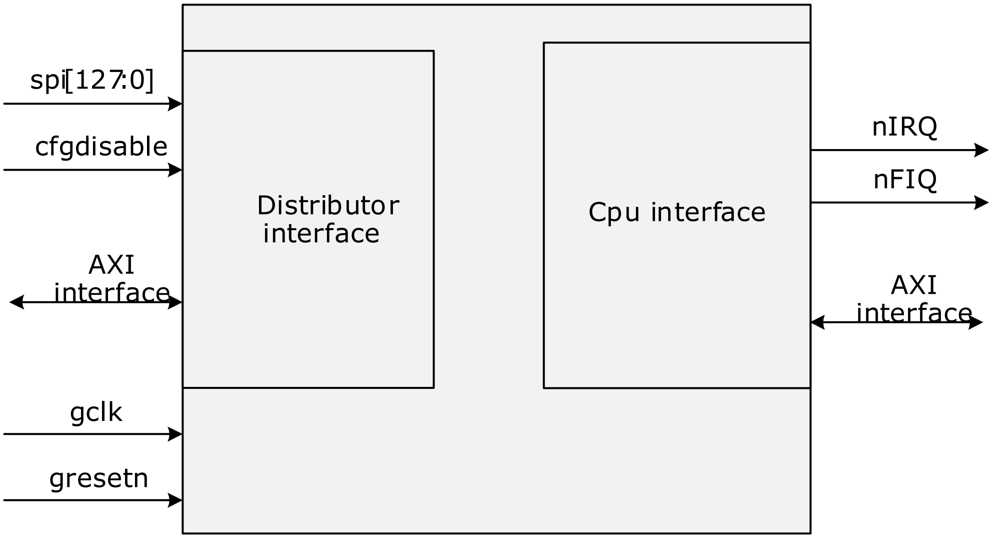


Fig. 10-1 Block Diagram

10.3 Function Description

Please refer to the document "IHI0048B_gic_architecture_specification.pdf" for the detailed function description.

Chapter 11 Power Management Unit (PMU)

11.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RK3308. The RK3308 PMU is dedicated for managing the power of the whole chip.

11.1.1 Features

- Support 2 voltage domains: VD_CORE, VD_LOGIC
- Support power off VD_CORE only
- 4 Power domains in VD_CORE: PD_CPU_0/1/2/3
- PD_CPU_0/1/2/3 support CPU auto power down, support SCU auto power down
- There is no independent power domain in the VD_LOGIC
- Support DDR self-refresh, auto-gating and retention
- Support wakeup sources
  - SDMMC detect
  - SDMMC detect interrupt
  - VAD interrupt
  - GPIO interrupt
  - USB detect interrupt
  - Interrupt of all timers
  - ARM interrupt
  - Interrupt output from GIC
  - Timeout
- Support Flush L2 by software and hardware
- Support NIU idle interface (idle request, ack and status)
- Support PMU debug through IO or UART interface

11.2 Block Diagram

11.2.1 Voltage partition

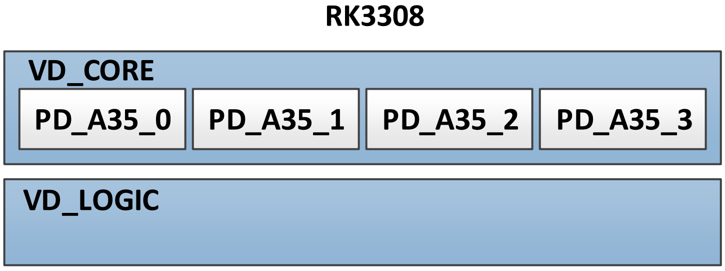


Fig. 11-1 RK3308 Power Domain Partition

The above diagram describes voltage domain partition.

Table 11-1 RK3308 Power Domain and Voltage Domain Summary

Voltage Domain	Blocks	Description
VD_CORE	PD_CPU0	CPU Core 0 with NEON and FPU
	PD_CPU1	CPU Core 1 with NEON and FPU
	PD_CPU2	CPU Core 2 with NEON and FPU
	PD_CPU3	CPU Core 3 with NEON and FPU
VD_LOGIC		DDR_CTRL, DDR_STDBY and DDR_MONITOR, GMAC, SFC, EMMC, NAND and SDIO, USB_OTG and USB_HOST, CRYPTO, DCF，DMAC, GIC, I2S, PDM, INTMEM, ROM, OTP, KEYREADER, CRU, CPU_BOOST,

Voltage Domain	Blocks	Description
		GRF, I2C, WDT, TIMER, TSADC, SARADC, SPI, PWM, GPIO, UART, DCF, PLL and ANALOG PHYs, PMU, VOP

11.2.2 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

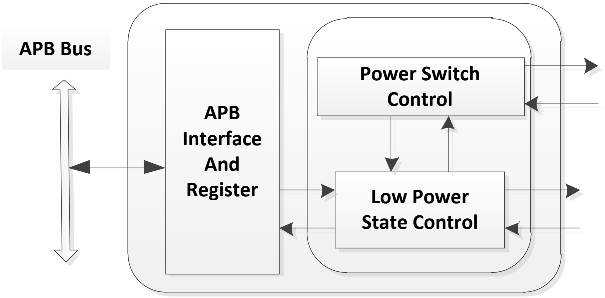


Fig. 11-2 PMU Block Diagram

11.3 Function Description

First of all, we define two operation modes of PMU, normal mode and low power mode. When operating at normal mode, that means software can manage power sources directly by accessing PMU register.

For example, Cortex-A35 CPU can write PMU_PWRDN_CON register to determine that power off/on which power domain independently.

When operating at low power mode, software manages power sources indirectly through FSM (Finite States Machine) in PMU and those settings always not take effect immediately. That means software also can configure PMU registers to power down/up some power resources, but these setting will not be executed immediately after configuration. They will delay to execute after FSM running in particular phase.

To entering low power mode, after setting some power configurations, the PMU_PWRMODE_COMMON_CON_LO[0] bit must be set 1 to enable PMU FSM. Then Cortex-A35 CPU needs to execute a WFI command to perform ready signal. After PMU detects all Cortex-A35 CPUs in WFI status, then the FSM will be fetched. And the specific power sources will be controlled during specific status in FSM. So the low power mode is a "delay affect" way to handle power sources inside the RK3308 chip.

11.4 Register Description

11.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG	0x0010	W	0x00000000	Wakeup source config register
PMU_PWRDN_CON	0x0018	W	0x00000000	Power down control register
PMU_PWRDN_ST	0x0020	W	0x00000000	Power status register
PMU_PWRMODE_CORE_CON	0x0024	W	0x00000000	Core power mode register
PMU_PWRMODE_COMMON_CON_LO	0x002c	W	0x00000000	Power mode register low 16 bits
PMU_PWRMODE_COMMON_CON_HI	0x0030	W	0x00000000	Power mode register high 16 bits

Name	Offset	Size	Reset Value	Description
PMU_SFT_CON_LO	0x0034	W	0x00000000	Software control register low 16 bits
PMU_SFT_CON_HI	0x0038	W	0x00000000	Software control register high 16 bits
PMU_INT_CON_LO	0x003c	W	0x00000000	Interrupt control register low 16 bits
PMU_INT_CON_HI	0x0040	W	0x00000000	Interrupt control register high 16 bits
PMU_INT_ST	0x0044	W	0x00000000	Interrupt status register
PMU_CORE_PWR_ST	0x0060	W	0x00000000	PD_CORE status register
PMU_BUS_IDLE_REQ	0x0064	W	0x00000000	Idle request register
PMU_BUS_IDLE_ST	0x006c	W	0x00000000	Idle status register
PMU_POWER_ST	0x0070	W	0x00000000	Power state register
PMU_OSC_CNT_LO	0x0074	W	0x00005dc0	Oscillator counter low 16 bits
PMU_OSC_CNT_HI	0x0078	W	0x00000000	Oscillator counter high 16 bits
PMU_PLLLOCK_CNT_LO	0x007c	W	0x00005dc0	PLL lock counter low 16 bits
PMU_PLLLOCK_CNT_HI	0x0080	W	0x00000000	PLL lock counter high 16 bits
PMU_DDRIO_PWRON_CNT_LO	0x0094	W	0x00005dc0	DDR IO power counter low 16 bits
PMU_DDRIO_PWRON_CNT_HI	0x0098	W	0x00000000	DDR IO power counter high 16 bits
PMU_DDR_SREF_ST	0x00a4	W	0x00000003	DDR self-refresh status register
PMU_SYS_REG0_LO	0x00a8	W	0x00000000	System register0 low 16 bits
PMU_SYS_REG0_HI	0x00ac	W	0x00000000	System register0 high 16 bits
PMU_SYS_REG1_LO	0x00b0	W	0x00000000	System register1 low 16 bits
PMU_SYS_REG1_HI	0x00b4	W	0x00000000	System register1 high 16 bits
PMU_SYS_REG2_LO	0x00b8	W	0x00000000	System register2 low 16 bits
PMU_SYS_REG2_HI	0x00bc	W	0x00000000	System register2 high 16 bits
PMU_SYS_REG3_LO	0x00c0	W	0x00000000	System register3 low 16 bits
PMU_SYS_REG3_HI	0x00c4	W	0x00000000	System register3 high 16 bits
PMU_CORE_PWRDN_CNT_LO	0x00c8	W	0x00005dc0	Vd_core power down count low 16 bits
PMU_CORE_PWRDN_CNT_HI	0x00cc	W	0x00000000	Vd_core power down count high 16 bits
PMU_CORE_PWRUP_CNT_LO	0x00d0	W	0x00005dc0	Vd_core power up count low 16 bits
PMU_CORE_PWRUP_CNT_HI	0x00d4	W	0x00000000	Vd_core power up count low 16 bits
PMU_TIMEOUT_CNT_LO	0x00d8	W	0x00005dc0	Time out count low 16 bits
PMU_TIMEOUT_CNT_HI	0x00dc	W	0x00000000	Time out count high 16 bits
PMU_CPU0APM_CON	0x00e0	W	0x00000000	CPU0 apm control register
PMU_CPU1APM_CON	0x00e4	W	0x00000000	CPU1 apm control register
PMU_CPU2APM_CON	0x00e8	W	0x00000000	CPU2 apm control register

Name	Offset	Size	Reset Value	Description
PMU_CPU3APM_CON	0x00ec	W	0x00000000	CPU3 apm control register
PMU_INFO_TX_CON	0x00f0	W	0x00000000	PMU info output control 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

PMU_WAKEUP_CFG

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:9	RO	0x0	reserved
8	RW	0x0	wakeup_timeout_en pmu timeout wakeup enable 1'b0: disable 1'b1: enable
7	RW	0x0	wakeup_usbdev_en usb wakeup enable 1'b0: disable 1'b1: enable
6	RW	0x0	wakeup_timer_en timer wakeup enable 1'b0: disable 1'b1: enable
5	RW	0x0	wakeup_sdmmc_grf_irq_en sdmmc grf irq enable, should set GRF. DETECT_GRF_SDMMC_DETECT_CON. sdmmc_detectn_pos_irq_enable or sdmmc_detectn_neg_irq_enable to “1” 1'b0: disable 1'b1: enable
4	RW	0x0	wakeup_sdmmc_en sdmmc io wakeup enable 1'b0: disable 1'b1: enable
3	RW	0x0	wakeup_vad_en vad interrupt wakeup enable 1'b0: disable 1'b1: enable
2	RW	0x0	wakeup_gpio0_int_en gpio0 interrupt wakeup enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
1	RW	0x0	wakeup_arm_int_en arm interrupt before entering gic as wakeup source 1'b0: disable 1'b1: enable
0	RW	0x0	wakeup_int_cluster_en cluster interrupt generated by gic wakeup enable 1'b0: disable 1'b1: enable

PMU_PWRDN_CON

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:5	RO	0x0	reserved
4	RW	0x0	pd_scu_pwrdwn_en pd_scu power down enable 1'b0: disable 1'b1: enable
3	RW	0x0	pd_a35_3_pwrdwn_en pd_a35_3 power down enable 1'b0: disable 1'b1: enable
2	RW	0x0	pd_a35_2_pwrdwn_en pd_a35_2 power down enable 1'b0: disable 1'b1: enable
1	RW	0x0	pd_a35_1_pwrdwn_en pd_a35_1 power down enable 1'b0: disable 1'b1: enable
0	RW	0x0	pd_a35_0_pwrdwn_en pd_a35_0 power down enable 1'b0: disable 1'b1: enable

PMU_PWRDN_ST

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	pd_scu_pwr_status 1'b0: pd_scu is power on 1'b1: pd_scu is power off

Bit	Attr	Reset Value	Description
3	RW	0x0	pd_a35_3_pwr_status 1'b0: pd_a35_3 is power on 1'b1: pd_a35_3 is power off
2	RW	0x0	pd_a35_2_pwr_status 1'b0: pd_a35_2 is power on 1'b1: pd_a35_2 is power off
1	RW	0x0	pd_a35_1_pwr_status 1'b0: pd_a35_1 is power on 1'b1: pd_a35_1 is power off
0	RO	0x0	pd_a35_0_pwr_status 1'b0: pd_a35_0 is power on 1'b1: pd_a35_0 is power off

PMU_PWRMODE_CORE_CON

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:12	RO	0x0	reserved
11	RW	0x0	vpll1_pd_en VPLL1 power down enable in low power mode 1'b0: disable 1'b1: enable
10	RW	0x0	vpll0_pd_en VPLL0 power down enable in low power mode 1'b0: disable 1'b1: enable
9	RW	0x0	dpll_pd_en DPLL power down enable in low power mode 1'b0: disable 1'b1: enable
8	RW	0x0	apll_pd_en APLL power down enable in low power mode 1'b0: disable 1'b1: enable
7	RW	0x0	l2_flush_en 1'b0: not flush L2 in low power mode 1'b1: flush L2 in low power mode
6	RW	0x0	l2_idle_en 1'b0: not wait for L2 idle when in low power mode 1'b1: wait for L2 idle in low power mode
5	RW	0x0	scu_pd_en 1'b0: not power down scu(vd_core) in low power mode 1'b1: power down scu(vd_core) in low power mode

Bit	Attr	Reset Value	Description
4	RW	0x0	clr_core 1'b0: not clear core niu in low power mode 1'b1: clear core niu in low power mode
3	RW	0x0	cpu0_pd_en 1'b0: not power down cpu0 in low power mode 1'b1: power down cpu0 in low power mode
2	RW	0x0	pmu_sleep_pol pmu_sleep polarity 1'b0: high active 1'b1: low active
1	RW	0x0	clk_core_src_gate_en 1'b0: core clock not gating when power mode 1'b1: core clock gating when power mode
0	RW	0x0	global_int_disable_cfg 1'b0: global interrupt enable 1'b1: global interrupt disable

PMU_PWRMODE_COMMON_CON_LO
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15	RW	0x0	clr_msch 1'b0: not clear msch niu when in low power mode 1'b1: clear msch niu when in low power mode
14	RW	0x0	clr_voice 1'b0: not clear voice niu when in low power mode 1'b1: clear voice niu when in low power mode
13	RW	0x0	clr_peri2bus 1'b0: not clear peri2bus niu in low power mode 1'b1: clear peri2bus niu in low power mode This bit should be same as the "clr_peri" and "clr_peri2msch"
12	RW	0x0	clr_peri 1'b0: not clear peri niu in low power mode 1'b1: clear peri niu in low power mode This bit should be same as the "clr_peri2bus" and "clr_peri2msch"
11	RW	0x0	clr_bus 1'b0: not clear bus niu in low power mode 1'b1:clear bus niu in low power mode
10	RW	0x0	ddr_ret_de_req set 1 to de-request for ddr io retention bit when exiting the low power mode
9	RW	0x0	ddrio_ret_en 1'b0: ddr io retention disable in low power mode 1'b1: ddr io retention enable in low power mode
8	RW	0x0	ddrphy1x_clk_src_gate_en 1'b0: ddr phy 1x clock source gating disable in low power mode 1'b1: ddr phy 1x clock source gating enable in low power mode
7	RW	0x0	ddrphy4x_clk_src_gate_en 1'b0: ddr phy 4x clock source gating disable in low power mode 1'b1: ddr phy 4x clock source gating enable in low power mode
6	RW	0x0	sref_enter_en 1'b0: ddr not enter self-refresh in low power mode 1'b1: ddr enter self-refresh in low power mode
5	RW	0x0	osc_24m_dis 1'b0: not disable 24MHz OSC in low power mode 1'b1: disable 24MHz OSC in low power mode
4	RW	0x0	lf_32k clock switch to 32KHz clock in low power mode 1'b0: disable 1'b1: enable

3	RW	0x0	lf_24m clock switch to 24MHz clock in low power mode 1'b0: disable 1'b1: enable
2	RW	0x0	pll_pd_en 1'b0: not power down pll in low power mode 1'b1: power down pll in low power mode
1	RO	0x0	reserved
0	RW	0x0	power_mode_en 1'b0: low power mode disable 1'b1: low power mode enable

PMU_PWRMODE_COMMON_CON_HI

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:6	RO	0x0	reserved
5	RW	0x0	scu_apm_en 1'b0: vd_core auto power down disable 1'b1: vd_core auto power down enable
4	RW	0x0	scu_auto_gating_en 1'b0: vd_core auto gating disable 1'b1: vd_core auto gating enable
3	RW	0x0	pd_bus_clk_src_gate_en clock gating bus niu when in low power mode 1'b0: not gating clock 1'b1: gating clock
2	RW	0x0	pd_peri_clk_src_gate_en clock gating peri niu when in low power mode 1'b0: not gating clock 1'b1: gating clock
1	RW	0x0	wait_wakeup_begin set "1" to start the wait wakeup state
0	RW	0x0	clr_peri2msch 1'b0: peri2msch niu not clear in low power mode 1'b1: peri2msch niu clear in low power mode This bit should be same as the "clr_peri2bus" and "clr_peri"

PMU_SFT_CON_LO

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0

Bit	Attr	Reset Value	Description
15	RW	0x0	dbgpwrup_cfg debug power up status for cpu0 1'b0: power off 1'b1: power on
14	RW	0x0	osc_disable_cfg osc disable software config 1'b0: osc work normally 1'b1: osc disable
13	RW	0x0	pmu_lf_24m_ena_cfg for software to switch pmu clock to 24MHz clock 1'b0: normal working clock 1'b1: switch to 24MHz clock
12	RW	0x0	pmu_lf_32k_ena_cfg for software to switch pmu clock to 32KHz clock 1'b0: normal working clock 1'b1: switch to 32KHz clock
11	RW	0x0	lf_24m_ena_cfg for software to switch system clock (except pmu) to 24MHz clock 1'b0: normal working clock 1'b1: switch to 24MHz clock
10	RW	0x0	lf_32k_ena_cfg for software to switch system clock (except pmu) to 32KHz clock 1'b0: normal working clock 1'b1: switch to 32KHz clock
9	RW	0x0	power_off_ddrio_cfg for software to ddr io power off 1'b0: power on 1'b1: power off
8	RW	0x0	ddr_io_ret_cfg for software to config ddr io retention 1'b0: not io retention 1'b1: io retention
7	RW	0x0	upctl_c_sysreq_cfg for software to config upctl to self-refresh 1'b0: not self-refresh request 1'b1: self-refresh request
6	RW	0x0	cluster_clk_src_gating_cfg for software to config cluster clock source gating 1'b0: not gate clock 1'b1: gate clock
5	RO	0x0	reserved
4	RW	0x0	l2flushreq_cluster_cfg for software to config L2 flush 1'b0: not flush 1'b1: flush

Bit	Attr	Reset Value	Description
3	RW	0x0	vpll1_pd_cfg for software to config vpll1 power off 1'b0: power on 1'b1: power off
2	RW	0x0	vpll0_pd_cfg for software to config vpll0 power off 1'b0: power on 1'b1: power off
1	RW	0x0	dpll_pd_cfg for software to config dpll power off 1'b0: power on 1'b1: power off
0	RW	0x0	apll_pd_cfg for software to config apll power off 1'b0: power on 1'b1: power off

PMU_SFT_CON_HI

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15	RW	0x0	dbgpwrupreq_en cpu debug power up request enable 1'b0: request disable 1'b1: request enable
14:11	RW	0x0	dbgnopwrdwn_enable cpu debug power down enable, bit[3:0] for cpu3-cpu0 1'b0: power down disable 1'b1: power down enable
10:0	RO	0x0	reserved

PMU_INT_CON_LO

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:11	RO	0x0	reserved
10	RW	0x0	wakeup_timeout_int_en wakeup by timeout interrupt enable 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
9	RW	0x0	wakeup_sdmmc_irq_int_en wakeup by sdmmc detectn interrupt enable 1'b0: disable 1'b1: enable
8	RW	0x0	wakeup_sdmmc_detectn_en wakeup by sdmmc detectn enable 1'b0: disable 1'b1: enable
7	RW	0x0	wakeup_vad_int_en wakeup by vad interrupt enable 1'b0: disable 1'b1: enable
6	RW	0x0	wakeup_gpio_int_int_en wakeup by gpio0 interrupt enable 1'b0: disable 1'b1: enable
5	RW	0x0	wakeup_usbdev_int_en wakeup by usb interrupt enable 1'b0: disable 1'b1: enable
4	RW	0x0	wakeup_timer_int_en wakeup by timer interrupt enable 1'b0: disable 1'b1: enable
3	RW	0x0	wakeup_arm_int_int_en wakeup by arm interrupt enable 1'b0: disable 1'b1: enable
2	RW	0x0	wakeup_int_cluster_int_en wakeup by cluster interrupt enable 1'b0: disable 1'b1: enable
1	RW	0x0	pwrmode_wakeup_int_en wakeup from low power mode interrupt enable 1'b0: disable 1'b1: enable
0	RO	0x0	pmu_int_en pmu global interrupt enable 1'b0: disable 1'b1: enable

PMU_INT_CON_HI
Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:5	RO	0x0	reserved
4	RW	0x0	pd_scu_pwr_switch_int_en pd_scu power switch interrupt enable 1'b0: disable 1'b1: enable
3	RW	0x0	pd_a35_3_pwr_switch_int_en pd_a35_3 power switch interrupt enable 1'b0: disable 1'b1: enable
2	RW	0x0	pd_a35_2_pwr_switch_int_en pd_a35_2 power switch interrupt enable 1'b0: disable 1'b1: enable
1	RW	0x0	pd_a35_1_pwr_switch_int_en pd_a35_1 power switch interrupt enable 1'b0: disable 1'b1: enable
0	RW	0x0	pd_a35_0_pwr_switch_int_en pd_a35_0 power switch interrupt enable 1'b0: disable 1'b1: enable

PMU_INT_ST

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RW	0x0	pd_scu_pwr_switch_status pd_scu power switch interrupt status 1’b0: no interrupt 1’b1: interrupt issue
19	RW	0x0	pd_a35_3_pwr_switch_status pd_a35_3 power switch interrupt status 1’b0: no interrupt 1’b1: interrupt issue
18	RW	0x0	pd_a35_2_pwr_switch_status pd_a35_2 power switch interrupt status 1’b0: no interrupt 1’b1: interrupt issue
17	RW	0x0	pd_a35_1_pwr_switch_status pd_a35_1 power switch interrupt status 1’b0: no interrupt 1’b1: interrupt issue

Bit	Attr	Reset Value	Description
16	RW	0x0	pd_a35_0_pwr_switch_status pd_a35_0 power switch interrupt status 1’b0: no interrupt 1’b1: interrupt issue
15:11	RO	0x0	reserved
10	RW	0x0	wakeup_timeout_status wakeup by timeout interrupt status 1’b0: no interrupt 1’b1: interrupt issue
9	RW	0x0	wakeup_sdmmc_irq_status wakeup by sdmmc detectn interrupt status 1’b0: no interrupt 1’b1: interrupt issue
8	RW	0x0	wakeup_sdmmc_status wakeup by sdmmc detectn status 1’b0: no interrupt 1’b1: interrupt issue
7	RW	0x0	wakeup_vad_status wakeup by vad interrupt status 1’b0: no interrupt 1’b1: interrupt issue
6	RW	0x0	wakeup_gpio_int_status wakeup by gpio0 interrupt status 1’b0: no interrupt 1’b1: interrupt issue
5	RW	0x0	wakeup_usbdev_status wakeup by usb interrupt status 1’b0: no interrupt 1’b1: interrupt issue
4	RW	0x0	wakeup_timer_status wakeup by timer interrupt status 1’b0: no interrupt 1’b1: interrupt issue
3	RW	0x0	wakeup_arm_int_status wakeup by arm interrupt status 1’b0: no interrupt 1’b1: interrupt issue
2	RW	0x0	wakeup_int_cluster_status wakeup by cluster interrupt status 1’b0: no interrupt 1’b1: interrupt issue
1	RW	0x0	pwrmode_wakeup_status wakeup from low power mode interrupt status 1’b0: no interrupt 1’b1: interrupt issue

Bit	Attr	Reset Value	Description
0	RO	0x0	reserved

PMU_CORE_PWR_ST
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu_standby_wfi 1'b0: cpu is not in wfi standby 1'b1: cpu is in wfi standby
2	RW	0x0	cpu_standby_wfe 1'b0: cpu is not in wfe standby 1'b1: cpu is in wfe standby
1	RW	0x0	l2_standby 1'b0: l2 is not in standby 1'b1: l2 is in standby
0	RO	0x0	l2flushdone 1'b0: L2 flush is not done 1'b1: L2 flush is done

PMU_BUS_IDLE_REQ
Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:7	RO	0x0	reserved
6	RW	0x0	idle_req_peri2msch_cfg peri2msch niu idle request 1'b0: not request 1'b1: request
5	RW	0x0	idle_req_msch_cfg msch niu idle request 1'b0: not request 1'b1: request
4	RW	0x0	idle_req_peri2bus_cfg software config peri2bus niu idle request 1'b0: not request 1'b1: request
3	RW	0x0	idle_req_peri_cfg software config peri niu idle request 1'b0: not request 1'b1: request
2	RW	0x0	idle_req_voice_cfg software config voice niu idle request 1'b0: not request 1'b1: request
1	RW	0x0	idle_req_core_cfg software config core niu idle request 1'b0: not request 1'b1: request
0	RW	0x0	idle_req_bus_cfg software config bus niu idle request 1'b0: not request 1'b1: request

PMU_BUS_IDLE_ST

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	idle_peri2msch peri2msch niu idle status 1'b0: niu not idle 1'b1: niu idle
21	RW	0x0	idle_msch msch niu idle status 1'b0: niu not idle 1'b1: niu idle

Bit	Attr	Reset Value	Description
20	RW	0x0	idle_peri2bus peri2bus niu idle status 1'b0: niu not idle 1'b1: niu idle
19	RW	0x0	idle_peri peri idle status 1'b0: niu not idle 1'b1: niu idle
18	RW	0x0	idle_voice voice niu idle status 1'b0: niu not idle 1'b1: niu idle
17	RW	0x0	idle_core core niu idle status 1'b0: niu not idle 1'b1: niu idle
16	RW	0x0	idle_bus bus niu idle status 1'b0: niu not idle 1'b1: niu idle
15:7	RO	0x0	reserved
6	RW	0x0	idle_ack_peri2msch peri2msch niu idle ack status 1'b0: niu not ack 1'b1: niu ack
5	RW	0x0	idle_ack_msch msch niu idle ack 1'b0: niu not ack 1'b1: niu ack
4	RW	0x0	idle_ack_peri2bus peri2bus niu idle ack status 1'b0: niu not ack 1'b1: niu ack
3	RO	0x0	idle_ack_peri peri niu idle ack status 1'b0: niu not ack 1'b1: niu ack
2	RO	0x0	idle_ack_voice voice niu idle ack status 1'b0: niu not ack 1'b1: niu ack
1	RO	0x0	idle_ack_core core niu idle ack status 1'b0: niu not ack 1'b1: niu ack

Bit	Attr	Reset Value	Description
0	RO	0x0	idle_ack_bus bus niu idle ack status 1'b0: niu not ack 1'b1: niu ack

PMU_POWER_ST
Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	power_state issue the low power state of pmu

PMU_OSC_CNT_LO
Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	osc_cnt_lo osc counter[15:0]

PMU_OSC_CNT_HI
Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	osc_cnt_hi osc counter[19:16]

PMU_PLLLOCK_CNT_LO
Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	plllock_cnt_lo plllock counter[15:0]

PMU_PLLLOCK_CNT_HI
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	plllock_cnt_hi plllock counter[19:16]

PMU_DDRIO_PWRON_CNT_LO
Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	ddrio_powerup_cnt_lo ddrio power up waiting time counter[15:0]

PMU_DDRIO_PWRON_CNT_HI
Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	ddrio_powerup_cnt_hi ddrio power up waiting time counter[19:16]

PMU_DDR_SREF_ST
Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x1	upctl_c_sysack upctl c_sysack status, same with GRF.GRF_UPCTL_STATUS0.upctrl_c_sysack 1'b0: acknowledge 1'b1: no acknowledge
0	RO	0x1	upctl_c_active upctl c_active status, same with GRF.GRF_UPCTL_STATUS0.upctrl_c_active 1'b0: enter the self-refresh 1'b1: not enter the self-refresh

PMU_SYS_REG0_LO
Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg0_lo pmu system register0[15:0]

PMU_SYS_REG0_HI
Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg0_hi pmu system register0[31:16]

PMU_SYS_REG1_LO
Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg1_lo pmu system register1[15:0]

PMU_SYS_REG1_HI

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg1_hi pmu system register1[31:16]

PMU_SYS_REG2_LO

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg2_lo pmu system register2[15:0]

PMU_SYS_REG2_HI

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg2_hi pmu system register2[31:16]

PMU_SYS_REG3_LO

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg3_lo pmu system register3[15:0]

PMU_SYS_REG3_HI

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb
15:0	RW	0x0000	pmu_sys_reg3_hi pmu system register3[31:16]

PMU_CORE_PWRDN_CNT_LO

Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_core_pwrdn_cnt_lo vd_core power down counter[15:0]

PMU_CORE_PWRDN_CNT_HI
Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_core_pwrdn_cnt_hi vd_core power down counter[19:16]

PMU_CORE_PWRUP_CNT_LO
Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_core_pwrdn_cnt_lo vd_core power up counter[15:0]

PMU_CORE_PWRUP_CNT_HI
Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_core_pwrdn_cnt_hi vd_core power up counter[19:16]

PMU_TIMEOUT_CNT_LO
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:0	RW	0x5dc0	pmu_timeout_cnt_lo timeout_cnt[15:0]

PMU_TIMEOUT_CNT_HI
Address: Operational Base + offset (0x00dc)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3:0	RW	0x0	pmu_timeout_cnt_hi timeout_cnt[19:16]

PMU_CPU0APM_CON

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu0_sft_wakeup software wakeup for cpu0: 1'b0: not issue the software wakeup 1'b1: issue the software wakeup
2	RW	0x0	global_int_disable0_cfg global interrupt disable for cpu0: 1'b0: not disable the interrupt 1'b1: disable the interrupt
1	RW	0x0	pd_cpu0_int_wakeup_en 1'b0: pd_cpu0 interrupt wakeup disable 1'b1: pd_cpu0 interrupt wakeup enable
0	RW	0x0	pd_cpu0_wfi_pwrdn_en 1'b0: pd_cpu0 wfi power down disable 1'b1: pd_cpu0 wfi power down enable

PMU_CPU1APM_CON

Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu1_sft_wakeup software wakeup for cpu1: 1'b0: not issue the software wakeup 1'b1: issue the software wakeup
2	RW	0x0	global_int_disable1_cfg global interrupt disable for cpu1: 1'b0: not disable the interrupt 1'b1: disable the interrupt
1	RW	0x0	pd_cpu1_int_wakeup_en 1'b0: pd_cpu1 interrupt wakeup disable 1'b1: pd_cpu1 interrupt wakeup enable
0	RW	0x0	pd_cpu1_wfi_pwrdn_en 1'b0: pd_cpu1 wfi power down disable 1'b1: pd_cpu1 wfi power down enable

PMU_CPU2APM_CON

Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu2_sft_wakeup software wakeup for cpu2: 1'b0: not issue the software wakeup 1'b1: issue the software wakeup
2	RW	0x0	global_int_disable2_cfg global interrupt disable for cpu2: 1'b0: not disable the interrupt 1'b1: disable the interrupt
1	RW	0x0	pd_cpu2_int_wakeup_en 1'b0: pd_cpu2 interrupt wakeup disable 1'b1: pd_cpu2 interrupt wakeup enable
0	RW	0x0	pd_cpu2_wfi_pwrdn_en 1'b0: pd_cpu2 wfi power down disable 1'b1: pd_cpu2 wfi power down enable

PMU_CPU3APM_CON
Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:4	RO	0x0	reserved
3	RW	0x0	cpu3_sft_wakeup software wakeup for cpu3: 1'b0: not issue the software wakeup 1'b1: issue the software wakeup
2	RW	0x0	global_int_disable3_cfg global interrupt disable for cpu3: 1'b0: not disable the interrupt 1'b1: disable the interrupt
1	RW	0x0	pd_cpu3_int_wakeup_en 1'b0: pd_cpu3 interrupt wakeup disable 1'b1: pd_cpu3 interrupt wakeup enable
0	RW	0x0	pd_cpu3_wfi_pwrdn_en 1'b0: pd_cpu3 wfi power down disable 1'b1: pd_cpu3 wfi power down enable

PMU_INFO_TX_CON
Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask 16 bits write mask for lsb 15-0
15:8	RW	0x00	pmu_info_tx_intv_time pmu informations transmit counter

Bit	Attr	Reset Value	Description
7	RO	0x0	reserved
6:4	RW	0x0	pmu_info_tx_byte_con pmu informations selection 3'b000: info_tx_byte = power_state other: reserved
3:1	RO	0x0	reserved
0	RW	0x0	pmu_info_tx_en pmu informations transmit output enable 1'b0: disable 1'b1: enable

11.5 Timing Diagram

11.5.1 Each domain power switch timing

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

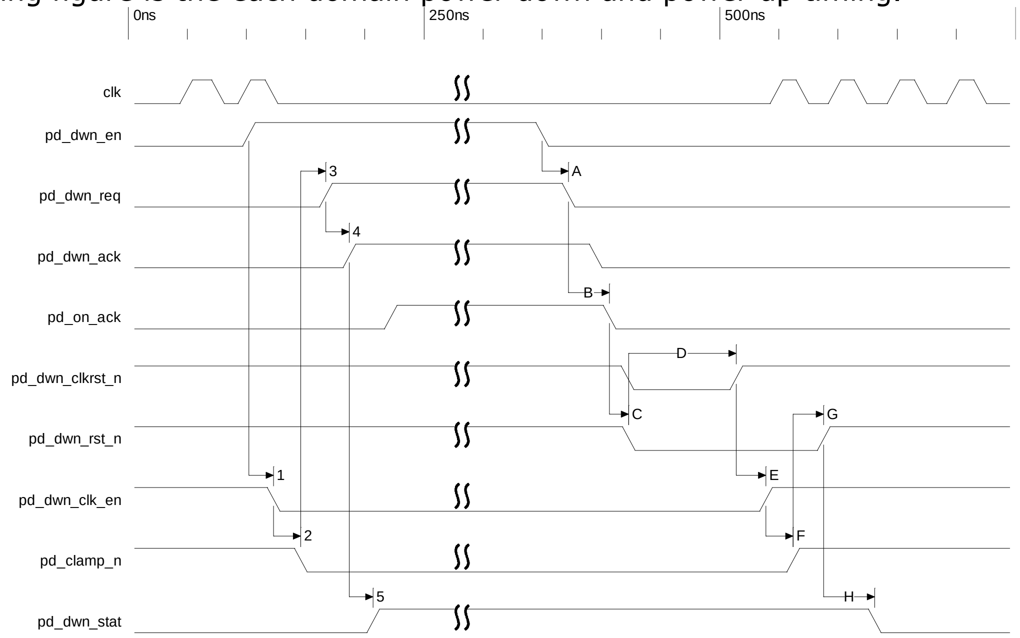


Fig. 11-3 Each Domain Power Switch Timing

11.5.2 External wakeup PAD timing

The PMU supports a lot of external wakeup sources, such as SD/MMDC, USBDEV 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.

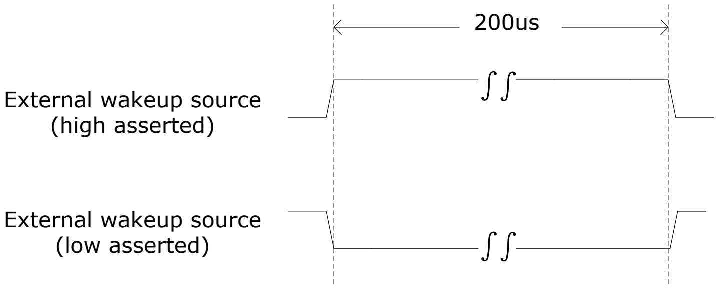


Fig. 11-4 External Wakeup Source PAD Timing

11.6 Application Note

11.6.1 Low power mode

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

Table 11-2 Low Power State

Num	STATES	Description
0	ST_NORMAL	Still in normal state
1	ST_CPU0_PWRDN	Power down cpu core 0
2	ST_L2_FLUSH	Flush L2
3	ST_L2_IDLE	L2 Idle
4	ST_TRANS_NO_FIN_CORE	wait for buses in vd_core idle
5	ST_SCU_PWRDN	power down vd_core(not really remove power, assert reset only)
6	ST_CORE_CLK_DIS	clock gating core clock
7	ST_TRANS_NO_FIN_BUS	wait for buses in vd_logic idle
8	ST_SREF_ENTER	DDR enter self-refresh
9	ST_DDR_IO_RET	DDR IO retention
10	ST_DDR_IO_PWROFF	NOT USED
11	ST_CLK_LF	Vd_logic clock switch to low frequency
12	ST_PLL_PWRDN	power down PLL
13	ST_24M_OSC_DIS	disable 24M OSC
14	ST_WAIT_WAKEUP	wait for wakeup
15	ST_24M_OSC_EN	enable 24M OSC
16	ST_PMU_HF	pmu clock switch to high frequency
17	ST_PLL_PWRUP	power up PLL
18	ST_CLK_HF	Vd_logic clock switch to high frequency
19	ST_DDR_IO_PWRUP	NOT USED
20	ST_SREF_EXIT	DDR exit self-refresh
21	ST_TRANS_RESTORE_BUS	release idle request to vd_logic bus
22	ST_CORE_CLK_EN	Enable core clock
23	ST_SCU_PWRUP	power up vd_core(de-assert reset)
24	ST_TRANS_RESTORE_CORE	release idle request to vd_core
25	ST_CPU0_PWRUP	Power up CPU core 0

11.6.2 Debug IO

RK3308 provide PMU Debug IO for FSM observation. Each IO maps to the correspond bit of the PMU_POWER_ST register.

Table 11-3 Debug IO MUX

Module Pin	Direction	Pin Name	IOMUX Setting
power_state [0]	O	GPIO4_D0/SDMMC_D0	GRF_SOC_CON2[15]=1’b1 & GRF_GPIO4D_IOMUX[1:0] = 2’b00
power_state [1]	O	GPIO4_D1/SDMMC_D1	GRF_SOC_CON2[15]=1’b1 & GRF_GPIO4D_IOMUX[3:2] = 2’b00
power_state [2]	O	GPIO4_D2/SDMMC_D2/UART 2_RX_M1	GRF_SOC_CON2[15]=1’b1 & GRF_GPIO4D_IOMUX[5:4] = 2’b00

Module Pin	Direction	Pin Name	IOMUX Setting
power_state [3]	O	GPIO4_D3/SDMMC_D3/UART 2_TX_M1	GRF_SOC_CON2[15]=1'b1 & GRF_GPIO4D_IOMUX[7:6] = 2'b00
power_state [4]	O	GPIO4_D4/SDMMC_CMD	GRF_SOC_CON2[15]=1'b1 & GRF_GPIO4D_IOMUX[9:8] = 2'b00
debug_Sout	O	GPIO4_D5/SDMMC_CLK	GRF_SOC_CON1[15]=1'b1 & GRF_GPIO4D_IOMUX[11:10] = 2'b00

Another way, RK3308 provided the serial output by the debug_sout for FSM observation by setting the bit[0] of PMU_INFO_TX_CON register. Once the bit was set to "1", pmu will sent the serial signal as the following timing diagram.

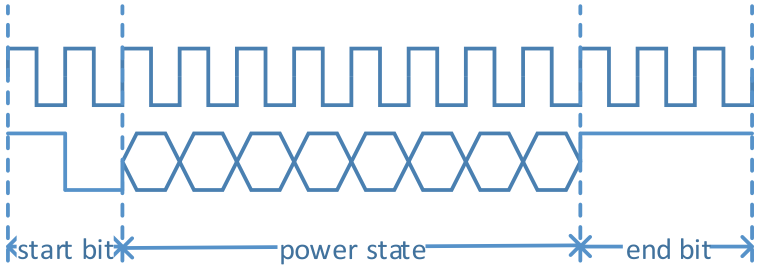


Fig. 11-5 PMU Info TX Timing

Start_bit: always is "10";
Power_state: shown the current power state of pmu FSM
End_bit: pmu_info_tx_intv_time + 2

11.6.3 Clock Switch

When the system work in the low power mode, some modules (including GPIO, DETECT_GRF, GIC, TIMER, PMU) should still work to generate the wakeup sources. The bus clocks and working clocks of these modules can be switch to low speed frequency (24MHz or 32KHz) by the PMU FSM.
There is two bits for switching to 24MHz or 32KHz in the PMU_PWRMODE_COMMON_CON_LO register.
lf_24m: bit[3], switch to 24MHz;
lf_32K: bit[4], switch to 32KHz;
There is only one bit can been set in the low power mode.
For example, if the lf_24m was set to "1", the following clocks will be switched to 24MHz: pclk_gpio, pclk_detect_grf, aclk_gic, pclk_timer, pclk_pmu.

Note: the high speed frequency of clk_pmu is 24MHz, so it only can be switched to 32KHz.

11.6.4 Power Mode Control Registers and Sft Control Registers

There are two types registers: PMU_PWRMODE_COMMON_CON_x and PMU_SFT_CON_x to control the same function, for example pll on/off, ddr self-refresh. The PMU_PWRMODE_COMMON_CON_X register in valid only in the pmu low power mode, and the PMU_SFT_CON_x register will take effect as soon as the register it set.
The following figure shows the structure of the two types registers:

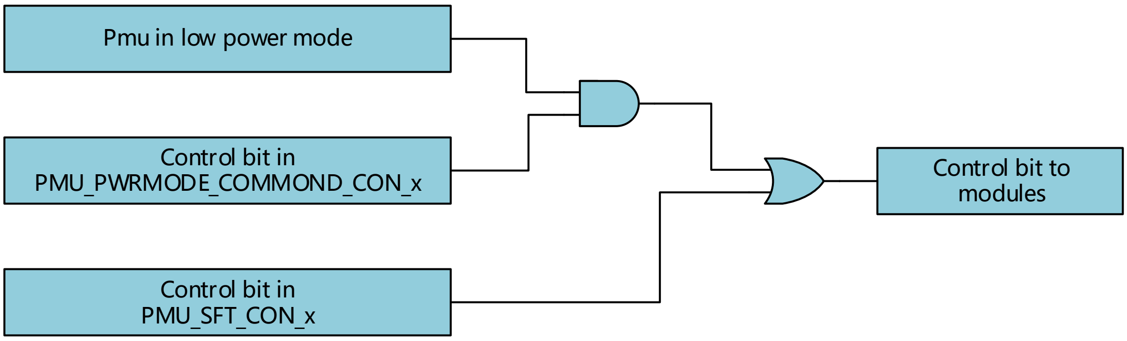

Pmu in low power mode
Control bit in PMU_PWRMODE_COMMOND_CON_x
Control bit in PMU_SFT_CON_x
Control bit to modules

Fig. 11-6 PMU Control Register Relationship

Chapter 12 NAND Flash Controller (NANDC)

12.1 Overview

NAND Flash Controller (NANDC) is used to control data transmission from host to NAND Flash device or from NAND Flash device to host. NANDC is connected to AHB BUS through an AHB Master and an AHB Slave. The data transmission between host and external memory can be done through AHB Master interface or AHB Slave interface.

NANDC supports the following features:

- Software Interface Type
  - Support directly mode
- Flash Interface Type
  - Support Asynchronous Flash Interface with 8bits data width ("Asyn8x" for short)
  - Support 1 NAND Flash devices(1 chip select)
- Flash Type
  - Support SLC Flash
- Flash Interface Timing
  - Asyn8x: configurable timing
- BCH/ECC Ability
  - 16bit/1KB BCH/ECC: support 16bit BCH/ECC, which can detect and correct up to 16 error bits in every 1K bytes data and 4 bytes meta-data
- Transmission Ability
  - Support 32K bytes data transmission at a time at most
  - Support two transfer working modes: Bypass or DMA
- Internal Memory
  - 2 built-in SRAMs, and the size is 1k bytes respectively
  - Can be accessed by other masters
  - Can be operated in Ping-Pong mode by other masters
- FIFO Mode
  - One built-in FIFO with 32 bits wide and 8 depth
  - Store command, address and data temporarily that are intend to write to the external NAND Flash

12.2 Block Diagram

NANDC comprises with:

- MIF: AHB Master Interface
- SIF: AHB Slave Interface
- SRIF: SRAM Interface
- TRANSC: Transfer Controller
- BCHENC: BCH Encoder
- BCHDEC: BCH Decoder
- FIF_GEN: Flash Interface Generation

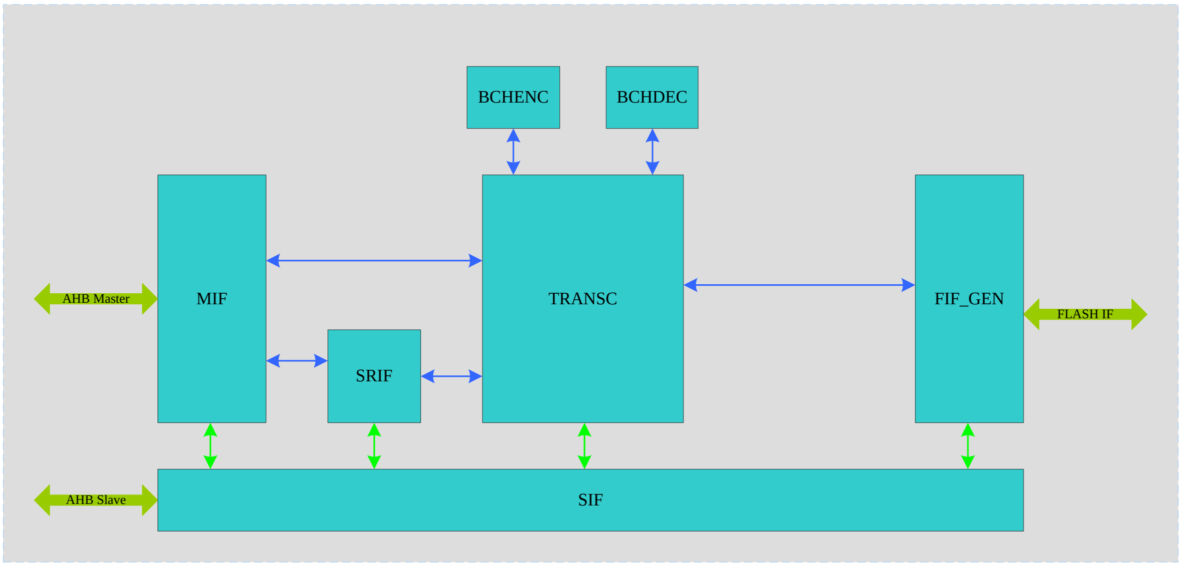


Fig. 12-1 NANDC Block Diagram

12.3 Function Description

12.3.1 AHB Interface

There is an AHB master interface in NANDC, which is selectable and configurable. It is responsible for transferring data from external memory to internal memory when NAND Flash program, or inverse when NAND Flash read.

There is an AHB slave interface in NANDC. It is responsible for accessing registers and internal memories. The addresses of these registers and memories are listed in Register Description section.

12.3.2 Flash Type/Flash Interface

The NAND Flash controller supports asynchronous 8bits NAND Flash interface. You can use it by software (configure FMCTL) to suit for devices. Also you can configure their timing parameters by software (configure FMWAIT_ASYN) to have your desired rate.

12.3.3 BCH Encoder/BCH Decoder

The BCH Encoder is responsible for encoding data to be written into NAND Flash device. The encoded length is 1056 bytes, in which the data length is 1024 bytes, system information (Meta-data) is 4 bytes, BCH code is 28 bytes.

The BCH Decoder is responsible for decoding data read from NAND Flash device. The decoded length is 1056 bytes, in which the data length is 1024 bytes, spare length is 32 bytes.

12.4 Register Description

12.4.1 Internal Address Mapping

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

Table 12-1 NANDC Address Mapping

Base Address[12:8]	Device	Address Length	Offset Address Range
5'b00_00x(x=0, 1)	FLR	512 BYTE	0x0000 ~ 0x01ff
5'b00_01x(x=0, 1)	SPR	512 BYTE	0x0200 ~ 0x03ff
5'b00_10x(x=0, 1)	FLR1	512 BYTE	0x0400 ~ 0x05ff
5'b01_000	Flash0	256 BYTE	0x0800 ~ 0x08ff
5'b10_0xx(x=0, 1)	Sram0	1K BYTE	0x1000 ~ 0x13ff
5'b10_1xx(x=0, 1)	Sram1	1K BYTE	0x1400 ~ 0x17ff
5'b11_000	FIFO	1K BYTE	0x1800 ~ 0x1bff

12.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
NANDC_FMCTL	0x0000	W	0x00000600	Flash Interface Control Register
NANDC_FMWAIT_ASYN	0x0004	W	0x003ff7ff	Flash Timing Control Register For Asynchronous Timing
NANDC_FLCTL	0x0008	W	0x00100000	Internal Transfer Control Register
NANDC_BCHCTL	0x000c	W	0x00000008	BCH Control Register
NANDC_MTRANS_CFG	0x0010	W	0x000001d0	Bus Transfer Configuration Register
NANDC_MTRANS_SADDR0	0x0014	W	0x00000000	Start Address Register For Page Data Transmission
NANDC_MTRANS_SADDR1	0x0018	W	0x00000000	Start Address Register For Spare Data Transmission
NANDC_MTRANS_STAT	0x001c	W	0x00000000	Bus Transfer Status Register
NANDC_BCHST0	0x0020	W	0x04000000	BCH Status Register For Codeword 0~1
NANDC_BCHST1	0x0024	W	0x00000000	BCH Status Register For Codeword 2~3
NANDC_BCHST2	0x0028	W	0x00000000	BCH Status Register For Codeword 4~5
NANDC_BCHST3	0x002c	W	0x00000000	BCH Status Register For Codeword 6~7
NANDC_BCHST4	0x0030	W	0x00000000	BCH Status Register For Codeword 8~9
NANDC_BCHST5	0x0034	W	0x00000000	BCH Status Register For Codeword 10~11
NANDC_BCHST6	0x0038	W	0x00000000	BCH Status Register For Codeword 12~13
NANDC_BCHST7	0x003c	W	0x00000000	BCH Status Register For Codeword 14~15
NANDC_MTRANS_STAT2	0x015c	W	0x00000000	Bus Transfer Status Register2
NANDC_NANDC_VER	0x0160	W	0x00000801	NANDc Version Register
NANDC_INTEN	0x016c	W	0x00000000	NANDC Interrupt Enable Register
NANDC_INTCLR	0x0170	W	0x00000000	NANDC Interrupt Clear Register
NANDC_INTST	0x0174	W	0x00000000	NANDC Interrupt Status Register
NANDC_SPARE0_0	0x0200	W	0xffffffff	System Information For Codeword 0
NANDC_SPARE1_0	0x0230	W	0xffffffff	System Information For Codeword 1
NANDC_BCHST8	0x0520	W	0x00000000	BCH Status Register For Codeword 16~17
NANDC_BCHST9	0x0524	W	0x00000000	BCH Status Register For Codeword 18~19

NANDC_BCHST10	0x0528	W	0x00000000	BCH Status Register For Codeword 20~21
NANDC_BCHST11	0x052c	W	0x00000000	BCH Status Register For Codeword 22~23
NANDC_BCHST12	0x0530	W	0x00000000	BCH Status Register For Codeword 24~25
NANDC_BCHST13	0x0534	W	0x00000000	BCH Status Register For Codeword 26~27
NANDC_BCHST14	0x0538	W	0x00000000	BCH Status Register For Codeword 28~29
NANDC_BCHST15	0x053c	W	0x00000000	BCH Status Register For Codeword 30~31
NANDC_FLASH0_DATA	0x0800	W	0x00000000	Flash0 Data
NANDC_FLASH0_ADDR	0x0804	W	0x00000000	Flash0 Address
NANDC_FLASH0_CMD	0x0808	W	0x00000000	Command Send To Flash0

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

12.4.3 Detail Register Description

NANDC_FMCTL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:24	RW	0x0	read_delay The number of delay cycle to capture the NAND Flash data after posedge of rdn.
23:18	RO	0x0	reserved
17	RO	0x0	flash_abort_stat reserved
16	RW	0x0	flash_abort_en reserved
15	RW	0x0	syn_mode reserved
14	RW	0x0	syn_clken reserved
13	RW	0x0	tm Timing mode indication. 1'b0: Asynchronous Mode. 1'b1: reserved
12	RW	0x0	dwidth Flash data bus width indication. 1'b0: 8bits, active in both Asynchronous Mode flash and Synchronous Mode flash. 1'b1: 16bits, active only in Asynchronous Mode flash.
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x1	fifo_empty fifo empty signal. 1'b0: fifo is not empty; 1'b1: fifo is emtpy;
9	RO	0x1	frdy Flash ready/busy indicate signal. 1'b0: flash is busy. 1'b1: flash is ready. This bit is the sample of the pin of R/Bn.
8	RW	0x0	wp Flash write protect. 1'b0: flash program/erase disabled. 1'b1: flash program/erase enabled. This bit is output to the pin of WPn.
7:1	RO	0x0	reserved
0	RW	0x0	fcs0 Flash memory chip 0 select control. 1'b1: hold flash memory chip select activity. 1'b0: flash memory chip select activity free.

NANDC_FMWAIT_ASYN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:18	RW	0x0f	wait_frdy_dly The number of delay cycle to accept the flash ready signal.
17:12	RW	0x3f	csrw When in Asynchronous mode, this field specifies the number of processor clock cycles from the falling edge of CSn to the falling edge of RDn or WRn. The min value of csrw is 0.
11	RW	0x0	hard_rdy Hardware handshaking controller bit. When asserted, an external device asserts signal "RDY" to extend a wait-state access and the rest bits in this register will be ignored.
10:5	RW	0x3f	rwpw When in Asynchronous mode, this field specifies the width of RDn or WRn in processor clock cycles, 0x0<=rwpw<=0x3f.
4:0	RW	0x1f	rwcs When in Asynchronous mode, this field specifies the number of processor clock cycles from the rising edge of RDn or WRn to the rising edge of CSn, 0x0<=rwcs<=0x1f.

NANDC_FLCTL

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	bypass_fifo_mode The enable sigal for bypass with fifo mode. 1'b0: disable fifo mode 1'b1: enable fifo mode
29	RO	0x0	reserved
28	RW	0x0	low_power NANDc low power control 1'b0: normal mode 1'b1: low power mode
27:22	RW	0x00	page_num Transmission codeword number in internal DMA mode when bus-mode is master-mode 1~32: 1~32 codeword. default: not support. Notes: a. Only active in internal DMA mode b. Only active when bus-mode is master-mode
21	RW	0x0	page_size Transmission codeword size in internal DMA mode 1'b0: 1024bytes/codeword 1'b1: reserved
20	RO	0x1	tr_rdy Internal DMA transmission ready indication. 1'b0: internal DMA transmission is busy 1'b1: internal DMA transmission is ready When reading flash, tr_rdy should not be set to 1 until all data transmission and correct finished. When programming flash, tr_rdy should not be set to 1 until all data transmission finished. Notes: Only active in internal DMA mode.
19:12	RO	0x0	reserved
11	RW	0x0	lba_en LBA mode indication, 1 active. 1'b0: NO-LBA mode, NANDC should transfer both page data and spare data in every codeword, and the page size is 1024 bytes or 512 bytes determined by BCHCTL[16](bchpage), spare size is 32 if bchmode=2'b00. 1'b1: reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	cor_able Auto correct enable indication, 1 active. 1'b0: auto correct disable 1'b1: auto correct enable Notes: a. Only active in internal DMA mode. b. lba_en is prior to cor_able. When lba_en=1, cor_able is ignored.
9:8	RO	0x0	reserved
7	RW	0x0	flash_st_mod Mode for NANDC to start internal data transmission in internal DMA mode. 1'b0: busy mode: hardware should not start internal data transmission until flash is ready even flash_st is asserted. 1'b1: ready mode: hardware should start internal data transmission directly when flash_st is asserted. Notes: Only active in internal DMA mode.
6:5	RW	0x0	tr_count Transmission codeword number in internal DMA mode when bus-mode is slave-mode 2'b00: 0 codeword need transferred 2'b01: 1 codeword need transferred 2'b10: 2 codeword need transferred 2'b11: not supported Notes: a. Only active in internal DMA mode. b. Only active when bus-mode is slave-mode.
4	RW	0x0	st_addr Start buffer address. 1'b0: start transfer from sram0 1'b1: start transfer from sram1 Notes: Only active in internal DMA mode.
3	RW	0x0	bypass NANDC internal DMA bypass indication. 1'b0: bypass the internal DMA, data are transferred to/from flash by direct path. 1'b1: internal DMA active, data are transferred to/from flash by internal DMA.

Bit	Attr	Reset Value	Description
2	R/W SC	0x0	flash_st Start signal for NANDC to transfer data between flash and internal buffer in internal DMA mode. When asserted, it will auto cleared. 1'b0: not start transmission 1'b1: start transmission Notes: Only active in internal DMA mode
1	RW	0x0	flash_rdn Indicate data flow direction. 1'b0: NANDC read data from flash. 1'b1: NANDC write data to flash
0	R/W SC	0x0	flash_rst NANDC software reset indication. When asserted, it will auto cleared. 1'b0: not software reset 1'b1: software reset Notes: flash_rst is prior to flash_st

NANDC_BCHCTL

Address: Operational Base + offset (0x000c)

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

28	RW	0x0	bch_toddr
			enable signal for storing bch decode status into ddr.
			1'b0: disable;
			1'b1: enable;

27	RO	0x0	reserved
26:19	RW	0x00	bchthres BCH error number threshold
18	RW	0x0	bchmode1 High bit of BCH mode selection. BchMode=bchmode1, bchmode0: 2'b00: 16bitBCH 2'b01: reserved 2'b10: reserved 2'b11: reserved
17	RO	0x0	reserved

Bit	Attr	Reset Value	Description
16	RW	0x0	bchpage The data size indication when BCH is active. 1'b0: 1024 bytes, all the 1024 bytes data in codeword are valid data to be transferred. 1'b1: 512 bytes, higher 512bytes are valid, and lower 512bytes are invalid and stuffed with 0xff. Notes: a. Only active when data transferred in internal DMA mode. b. Only active for asynchronous flash.
15:8	RW	0x00	addr BCH active range selection. BCH should be active when access in range address.
7:5	RW	0x0	region BCH active region selection indication. 3'b000: Flash memory 0 region (flash 0) 3'b001: Flash memory 1 region (flash 1) 3'b010~3'b011：reserved
4	RW	0x0	bchmode0 BCH mode selection indication. BCH mode is determined by both bchmode0 and bchmode1, detailed information is showed in BCHCTL[18].
3	RW	0x1	bchepd BCH encoder/decoder power down indication. 1'b0: BCH encoder/decoder working. 1'b1: BCH encoder/decoder not working.
2	RW	0x0	mode_addrcare BCH address care mode selection indication. 1'b0: address care. 1'b1: address not care. Notes: This bit is just active for data transmission in bypass mode, but not for command and address transmission.
1	RO	0x0	reserved
0	R/W SC	0x0	bchrst BCH software reset indication, When asserted, it will auto cleared. 1'b0: not software reset 1'b1: software reset Notes: a. BCH Decoder should be software reset before decode begin. b. bch software reset should be used with NANDC software reset at the same time.

NANDC_MTRANS_CFG
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	R/W SC	0x0	ahb_rst ahb master interface software reset, auto cleared
14	RW	0x0	fl_pwd Flash power down indication, 1 active. 1'b0: Flash power on, data transferred through master interface is data that to be written into or read from flash. 1'b1: Flash power down, data transferred through master interface is not data that to be written into or read from flash. NANDC is just used as DMA for external memory and internal memory.
13:9	RW	0x00	incr_num AHB Master incr num indication. incr_num=1~16. When burst=001, software should configure incr_num. Notes: Only active for master-mode.
8:6	RW	0x7	burst AHB Master burst type indication: 3'b000: Single transfer 3'b011: 4-beat burst 3'b101: 8-beat Burst 3'b111: 16-beat burst default: not supported Notes: Only active for master-mode.
5:3	RW	0x2	hsize AHB Master data size indication: 3'b000: 8 bits 3'b001: 16 bits 3'b010: 32 bits default: not supported Notes: Only active for master-mode.
2	RW	0x0	bus_mode Bus interface selection. 1'b0: Slave interface, flash data is transferred through slave interface 1'b1: Master interface, flash data is transferred through master interface

1	RW	0x0	ahb_wr Data transfer direction through master interface. 1'b0: read direction(external memory ->internal memory) 1'b1: write direction (internal memory->external memory) Notes: a. Only active for master-mode. b. When read flash(flash_rdn=0), ahb_wr=1; when program flash(flash_rdn=1), ahb_wr=0.
0	R/W SC	0x0	ahb_wr_st Start indication for loading data from external memory to internal memory or storing data from internal memory to external memory through master. When asserted, it will be automatically cleared only active when fl_pwd is 1 Notes: a. Only active for master-mode and fl_pwd=1. b. When fl_pwd=0, flash is active, NANDC start to transfer data through master interface if flash_st=1 c. When fl_pwd=1, flash is not active, NANDC start to transfer data through master interface if ahb_wr_st=1

NANDC_MTRANS_SADDR0

Address: Operational Base + offset (0x0014)

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

NANDC_MTRANS_SADDR1

Address: Operational Base + offset (0x0018)

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

NANDC_MTRANS_STAT

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:16	RO	0x00	mtrans_cnt finished counter for codeword transmission through Master interface Notes: Only active for master-mode.
15:0	RO	0x0000	bus_err Bus error indication for codeword0~15. [0]: bus error for codeword 0 …… [15]: bus error for codeword 15 Notes: Only active for master-mode.

NANDC_BCHST0
Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum1_h1 Highest bit of err_hnum1
29	RO	0x0	err_tnum1_h1 Highest bit of err_tnum1
28	RO	0x0	err_hnum0_h1 Highest bit of err_hnum0
27	RO	0x0	err_tnum0_h1 Highest bit of err_tnum0
26	RO	0x1	bchrdy Ready indication for bch encoder/decoder, 1 active. 1'b0: bch encoder/decoder is busy 1'b1: bch encoder/decoder is ready
25:21	RO	0x00	err_hnum1_l5 Lower 5 bits of number of error bits found in first 512bytes of 1st backup codeword
20:16	RO	0x00	err_tnum1_l5 Lower 5 bits of number of error bits found in 1st backup codeword
15	RO	0x0	fail1 Indication for the 1st backup codeword decoded failed or not. 1'b0: decode successfully 1'b1: decode fail
14	RO	0x0	done1 Indication for finishing decoding the 1st backup codeword 1'b0: not finished 1'b1: finished

Bit	Attr	Reset Value	Description
13	RO	0x0	errf1 Indication for error found in 1st backup codeword. 1'b0: no error 1'b1: error found
12:8	RO	0x00	err_hnum0_l5 Lower 5 bits of number of error bits found in first 512bytes of current backup codeword
7:3	RO	0x00	err_tnum0_l5 Lower 5 bits of number of error bits found in current backup codeword
2	RO	0x0	fail0 Indication for current backup codeword decode failed or not 1'b0: decode successfully 1'b1: decode fail
1	RO	0x0	done0 Indication for finishing decoding the current backup codeword. 1'b0: not finished 1'b1: finished
0	RO	0x0	errf0 Indication for error found in current backup codeword. 1'b0: no error 1'b1: error found

NANDC_BCHST1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	err_hnum3_h1 Highest bit of err_hnum3
29	RO	0x0	err_tnum3_h1 Highest bit of err_tnum3
28	RO	0x0	err_hnum2_h1 Highest bit of err_hnum2
27	RO	0x0	err_tnum2_h1 Highest bit of err_tnum2
26	RO	0x0	reserved
25:21	RO	0x00	err_hnum3_l5 Lower 5 bits of number of error bits found in first 512bytes of 3th backup codeword
20:16	RO	0x00	err_tnum3_l5 Lower 5 bits of number of error bits found in 3th backup codeword

Bit	Attr	Reset Value	Description
15	RO	0x0	fail3 Indication for the 3th backup codeword decoded failed or not. 1'b0: decode successfully 1'b1: decode fail
14	RO	0x0	done3 Indication for finishing decoding the 3th backup codeword 1'b0: not finished 1'b1: finished
13	RO	0x0	errf3 Indication for error found in 3th backup codeword. 1'b0: no error 1'b1: error found
12:8	RO	0x00	err_hnum2_l5 Lower 5 bits of number of error bits found in first 512bytes of 2th backup codeword
7:3	RO	0x00	err_tnum2_l5 Lower 5 bits of number of error bits found in 2th backup codeword
2	RO	0x0	fail2 Indication for 2th backup codeword decode failed or not 1'b0: decode successfully 1'b1: decode fail
1	RO	0x0	done2 Indication for finishing decoding the 2th backup codeword. 1'b0: not finished 1'b1: finished
0	RO	0x0	errf2 Indication for error found in 2th backup codeword. 1'b0: no error 1'b1: error found

NANDC_BCHST2
Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd4_cwd5 BCHST information for 4th and 5th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST3
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd6_cwd7 BCHST information for 6th and 7th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST4

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd8_cwd9 BCHST information for 8th and 9th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST5

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd10_cwd11 BCHST information for 10th and 11th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST6

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd12_cwd13 BCHST information for 12th and 13th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST7

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd14_cwd15 BCHST information for 14th and 15th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_MTRANS_STAT2

Address: Operational Base + offset (0x015c)

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

NANDC_NANDC_VER

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000801	version Version indication for NANDC

NANDC_INTEN

Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	master_idle_int_en Enable for master idle interrupt 1'b0: interrupt disable 1'b1: interrupt enable When master_idle_int_en is active, an interrupt is generated if posedge of master idle happen
5:4	RO	0x0	reserved
3	RW	0x0	bchfail_int_en Enable for bch fail interrupt. 1'b0: interrupt disable 1'b1: interrupt enable When bchfail_int_en is active, an interrupt is generated if bch decode failed
2	RW	0x0	bcherr_int_en Enable for bch error interrupt. 1'b0: interrupt disable 1'b1: interrupt enable When bcherr_int_en is active, an interrupt is generated if bch decode error bit is larger than bchthres(BCHCTL[26:19])
1	RW	0x0	frdy_int_en Enable for flash_rdy interrupt 1'b0: interrupt disable 1'b1: interrupt enable When frdy_int_en is active, an interrupt is generated if flash R/B# changes from 0 to 1

Bit	Attr	Reset Value	Description
0	RW	0x0	dma_int_en Enable for internal DMA transfer finished interrupt 1'b0: interrupt disable 1'b1: interrupt enable When dma_int_en is active, an interrupt is generated if page_num(FLCTL[27:22]) of flash data transfer in DMA mode is finished

NANDC_INTCLR
Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	R/W SC	0x0	master_idle_int_clr Clear for master idle interrupt. When asserted, this bit will be auto cleared. 1'b0: interrupt not cleared 1'b1: interrupt cleared
5:4	RO	0x0	reserved
3	R/W SC	0x0	bchfail_int_clr Clear for bch decode fail interrupt. When asserted, this bit will be auto cleared. 1'b0: interrupt cleared 1'b1: interrupt not cleared
2	R/W SC	0x0	bcherr_int_clr Clear for bch error interrupt. When asserted, this bit will be auto cleared. 1'b0: interrupt cleared 1'b1: interrupt not cleared
1	R/W SC	0x0	frdy_int_clr Clear for flash_rdy interrupt. When asserted, this bit will be auto cleared. 1'b0: interrupt cleared 1'b1: interrupt not cleared
0	R/W SC	0x0	dma_int_clr Clear for internal DMA transfer finished interrupt. When asserted, this bit will be auto cleared. 1'b0: interrupt cleared 1'b1: interrupt not cleared

NANDC_INTST
Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RO	0x0	master_idle_int_stat Status for master idle interrupt, high active
5:4	RO	0x0	reserved
3	RO	0x0	bchfail_int_stat Status for bch decode fail interrupt, high active
2	RO	0x0	bcherr_int_stat Status for bch error interrupt, high active
1	RO	0x0	frdy_int_stat Status for flash_rdy interrupt, high active
0	RO	0x0	dma_int_stat Status for internal DMA transfer finished interrupt, high active

NANDC_SPARE0_0
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 0
23:16	RW	0xff	system_2 the 3rd system byte of codeword 0
15:8	RW	0xff	system_1 the 2nd system byte of codeword 0
7:0	RW	0xff	system_0 the 1st system byte of codeword 0

NANDC_SPARE1_0
Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:24	RW	0xff	system_3 the 4th system byte of codeword 1
23:16	RW	0xff	system_2 the 3rd system byte of codeword 1
15:8	RW	0xff	system_1 the 2nd system byte of codeword 1
7:0	RW	0xff	system_0 the 1st system byte of codeword 1

NANDC_BCHST8
Address: Operational Base + offset (0x0520)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd16_cwd17 BCHST information for 16th and 17th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST9

Address: Operational Base + offset (0x0524)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd18_cwd19 BCHST information for 18th and 19th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST10

Address: Operational Base + offset (0x0528)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd20_cwd21 BCHST information for 20th and 21th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST11

Address: Operational Base + offset (0x052c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd22_cwd23 BCHST information for 22th and 23th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST12

Address: Operational Base + offset (0x0530)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd24_cwd25 BCHST information for 24th and 25th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST13

Address: Operational Base + offset (0x0534)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd26_cwd27 BCHST information for 26th and 27th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST14

Address: Operational Base + offset (0x0538)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd28_cwd29 BCHST information for 28th and 29th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_BCHST15

Address: Operational Base + offset (0x053c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	bchst_cwd30_cwd31 BCHST information for 30th and 31th backup codeword. Bit assignment is similar to BCHST1 register. For more description, please refer to BCHST1 register.

NANDC_FLASH0_DATA

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	flash0_data valid for flash bypass internal dma mode(FLCTL[3]=0) FLASH0_DATA[15:0] is used for Asynchronous 16 bits mode or Toggle mode and FLASH0_DATA[7:0] is used for 8 bits Asynchronous mode. If Synchronous mode is selected, please refer to FLASH0_DATA_SYN.

NANDC_FLASH0_ADDR

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	WO	0x00	flash0_addr page or block address send to flash0 before flash erase, read or write operation

NANDC_FLASH0_CMD

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	flash0_cmd contain the command write to flash0

12.5 Interface Description

Table 12-2 NANDC Interface Description

Module Pin	Direction	PIN Name	IOMUX Setting
flash_ale	O	GPIO3_B3/FLASH_ALE/EMMC_PWREN/SPI1_CLK	GRF_GPIO3B_IOMUX [7:6]=2’b01

Module Pin	Direction	PIN Name	IOMUX Setting
flash_cle	O	GPIO3_B1/FLASH_CLE/EMMC_CLK	GRF_GPIO3B_IOMUX [3:2]=2'b01
flash_wrn	O	GPIO3_B0/FLASH_WRN/EMMC_CMD	GRF_GPIO3B_IOMUX [1:0]=2'b01
flash_rdn	O	GPIO3_B2/FLASH_RDN/SPI1_MISO	GRF_GPIO3B_IOMUX [5:4]=2'b01
flash_data[0]	I/O	GPIO3_A0/FLASH_D0/EMMC_D0/SFC_SIO0	GRF_GPIO3A_IOMUX [1:0]=2'b01
flash_data[1]	I/O	GPIO3_A1/FLASH_D1/EMMC_D1/SFC_SIO1	GRF_GPIO3A_IOMUX [3:2]=2'b01
flash_data[2]	I/O	GPIO3_A2/FLASH_D2/EMMC_D2/SFC_WP_SIO2	GRF_GPIO3A_IOMUX [5:4]=2'b01
flash_data[3]	I/O	GPIO3_A3/FLASH_D3/EMMC_D3/SFC_HOLD_SIO3	GRF_GPIO3A_IOMUX [7:6]=2'b01
flash_data[4]	I/O	GPIO3_A4/FLASH_D4/EMMC_D4/SFC_CLK	GRF_GPIO3A_IOMUX [9:8]=2'b01
flash_data[5]	I/O	GPIO3_A5/FLASH_D5/EMMC_D5/SFC_CSN0	GRF_GPIO3A_IOMUX [11:10]=2'b01
flash_data[6]	I/O	GPIO3_A6/FLASH_D6/EMMC_D6	GRF_GPIO3A_IOMUX [13:12]=2'b01
flash_data[7]	I/O	GPIO3_A7/FLASH_D7/EMMC_D7	GRF_GPIO3A_IOMUX [15:14]=2'b01
flash_rdy	I	GPIO3_B4/FLASH_RDY/I2C3_SDA_M1/SPI1_MOSI/UART3_RX	GRF_GPIO3B_IOMUX [10:8]=3'b001
flash_csn0	O	GPIO3_B5/FLASH_CSN0/I2C3_SCL_M1/SPI1_CSN0/UART3_TX	GRF_GPIO3B_IOMUX [14:12]=3'b001

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

Furthermore, different IOs are selected and connected to different NAND Flash interface, which is shown as follows.

Table 12-3 NANDC Interface Connection

Module Pin	Direction	Flash Interface			
		Asyn8x	Asyn16x	ONFI	Toggle
flash_csni(i=0)	O	√	-	-	-
flash_ale	O	√	-	-	-
flash_cle	O	√	-	-	-
flash_wrn	O	√	-	-	-
flash_rdn	O	√	-	-	-
flash_data[7:0]	I/O	√	-	-	-
flash_rdy	I	√	-	-	-

12.6 Application Notes

12.6.1 BCHST/BCHLOC/BCHDE/SPARE Application

1. BCHST

There are 16 BCHST-registers in NANDC to store 32 codeword's BCH decode status(bchst) information. Every register stores 2 codeword's bchst information except BCHST0, which not only includes bchst information, but also includes one bit for *bchrdy*.

Let bchst_cwd0~bchst_cwd31 be the bchst information for 32 codewords. In BCHST-registers, the latest codeword's bchst is stored into bchst_cwd0, and the former is shifted into bchst_cwd1. That is, bchst_cwd0→ bchst_cwd1 →……→bchst_cwd31. Therefore, for example, if 32 codewords are decoded, then bchst_cwd0 is the bch decode status for codeword31, and bchst_cwd31 is the bch decode status for codeword0.

bchst_cwd0 = {BCHST0[28], BCHST0[12:8], BCHST0[27], BCHST0[7:3], BCHST0[2:0]}
bchst_cwd1 = {BCHST0[30], BCHST0[25:21], BCHST0[29], BCHST0[20:16], BCHST0[15:13]}
bchst_cwd2 = {BCHST1[28], BCHST1[12:8], BCHST1[27], BCHST1[7:3], BCHST1[2:0]}
bchst_cwd3 = {BCHST1[30], BCHST1[25:21], BCHST1[29], BCHST1[20:16], BCHST1[15:13]}
bchst_cwd4 = {BCHST2[28], BCHST2[12:8] , BCHST2[27], BCHST2[7:3], BCHST2[2:0]}
bchst_cwd5 = {BCHST2[30], BCHST2[25:21], BCHST2[29], BCHST2[20:16], BCHST2[15:13]}
bchst_cwd6 = {BCHST3[28], BCHST3[12:8] , BCHST3[27], BCHST3[7:3], BCHST3[2:0]}
bchst_cwd7 = {BCHST3[30], BCHST3[25:21], BCHST3[29], BCHST3[20:16], BCHST3[15:13]}
bchst_cwd8 = {BCHST4[28], BCHST4[12:8] , BCHST4[27], BCHST4[7:3], BCHST4[2:0]}
bchst_cwd9 = {BCHST4[30], BCHST4[25:21], BCHST4[29], BCHST4[20:16], BCHST4[15:13]}
bchst_cwd10 = {BCHST5[28], BCHST5[12:8] , BCHST5[27], BCHST5[7:3], BCHST5[2:0]}

bchst_cwd11 = {BCHST5[30], BCHST5[25:21], BCHST5[29], BCHST5[20:16], BCHST5[15:13]}
bchst_cwd12 = {BCHST6[28], BCHST6[12:8] , BCHST6[27], BCHST6[7:3], BCHST6[2:0]}
bchst_cwd13 = {BCHST6[30], BCHST6[25:21], BCHST6[29], BCHST6[20:16], BCHST6[15:13]}
bchst_cwd14 = {BCHST7[28], BCHST7[12:8] , BCHST7[27], BCHST7[7:3], BCHST7[2:0]}
bchst_cwd15 = {BCHST7[30], BCHST7[25:21], BCHST7[29], BCHST7[20:16], BCHST7[15:13]}
bchst_cwd16 = {BCHST8[28], BCHST8[12:8] , BCHST8[27], BCHST8[7:3], BCHST8[2:0]}
bchst_cwd17 = {BCHST8[30], BCHST8[25:21], BCHST8[29], BCHST8[20:16], BCHST8[15:13]}
bchst_cwd18 = {BCHST9[28], BCHST9[12:8] , BCHST9[27], BCHST9[7:3], BCHST9[2:0]}
bchst_cwd19 = {BCHST9[30], BCHST9[25:21], BCHST9[29], BCHST9[20:16], BCHST9[15:13]}
bchst_cwd20 = {BCHST10[28], BCHST10[12:8] , BCHST10[27], BCHST10[7:3], BCHST10[2:0]}
bchst_cwd21 = {BCHST10[30], BCHST10[25:21], BCHST10[29], BCHST10[20:16], BCHST10[15:13]}
bchst_cwd22 = {BCHST11[28], BCHST11[12:8] , BCHST11[27], BCHST11[7:3], BCHST11[2:0]}
bchst_cwd23 = {BCHST11[30], BCHST11[25:21], BCHST11[29], BCHST11[20:16], BCHST11[15:13]}
bchst_cwd24 = {BCHST12[28], BCHST12[12:8] , BCHST12[27], BCHST12[7:3], BCHST12[2:0]}
bchst_cwd25 = {BCHST12[30], BCHST12[25:21], BCHST12[29], BCHST12[20:16], BCHST12[15:13]}
bchst_cwd26 = {BCHST13[28], BCHST13[12:8] , BCHST13[27], BCHST13[7:3], BCHST13[2:0]}
bchst_cwd27 = {BCHST13[30], BCHST13[25:21], BCHST13[29], BCHST13[20:16], BCHST13[15:13]}
bchst_cwd28 = {BCHST14[28], BCHST14[12:8] , BCHST14[27], BCHST14[7:3], BCHST14[2:0]}
bchst_cwd29 = {BCHST14[30], BCHST14[25:21], BCHST14[29], BCHST14[20:16], BCHST14[15:13]}
bchst_cwd30 = {BCHST15[28], BCHST15[12:8] , BCHST15[27], BCHST15[7:3], BCHST15[2:0]}
bchst_cwd31 = {BCHST15[30], BCHST15[25:21], BCHST15[29], BCHST15[20:16], BCHST15[15:13]}

2. SPARE

SPARE includes two register-groups, SPARE0 and SPARE1. Each group has 8 registers: SPARE0_0~SPARE0_7 and SPARE1_0~SPARE1_7. Only SPARE0_0 and SPARE1_0 can be accessed via software, others are used internally.

When in bch encoding, SPARE0_0 stores system information for codeword in sram0, SPARE0_n(n=1~7) stores encode information for codeword in sram0; SPARE1_0 stores system information for codeword in sram1, SPARE1_n(n=1~7) stores encode information for codeword in sram1.

When in bch decoding, SPARE0_n(n=0~7) stores the spare data read from NAND Flash for codeword in sram0 if BCHCTL[28]=0, the bch decode status(BCHST) information which reflects the current codeword will be written into ddr instead of SPARE0_1 if BCHCTL[28]=1; SPARE1_n(n=0~7) stores the spare data read from NAND Flash for codeword in sram1 if BCHCTL[28]=0, the bch decode status(BCHST) information which reflects the current codeword will be written into ddr instead of SPARE1_1 if BCHCTL[28]=1.

12.6.2 Bus Mode Application

MTRANS_CFG[2] determines whether the data load/store between internal memory and external memory is through slave interface or master interface.

1. Slave Mode

When MTRANS_CFG[2]=0, slave is selected. i. e. , NAND Flash data load/store between

internal memory and external memory is through slave interface by cpu or external DMA.
In this mode, software should store page data into internal memory and spare data into SPARE registers before starting NAND Flash program operation; and should load page data from internal memory and spare data from SPARE registers after finishing NAND Flash read operation.
In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are unused. The transfer codeword number is determined by FLCTL[6:5], and the maximum number is 2.
The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transfer is finished.

2. Master Mode

When MTRANS_CFG[2]=1, master is selected. i. e. , NAND Flash data load/store between internal memory and external memory is through master interface.
In this mode, software should initialize page data and spare data into external memory, and set their addresses in MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting NAND Flash program operation. Similarly, software should configure MTRANS_SADDR0 and MTRANS_SADDR1 respectively before starting NAND Flash read operation and could read data from addresses in MTRANS_SADDR0 and MTRANS_SADDR1 after NANDC transfer finish.
In this mode, MTRANS_CFG, MTRANS_SADDR0 and MTRANS_SADDR1 are used. The transfer codeword number is determined by FLCTL[26:22], and the maximum number is 32.
The judgment condition for finishing data transfer is FLCTL[20]. When FLCTL[20] is high, it means that data transmission is finished.
When MTRANS_CFG[2]=1, page data and spare data are stored in the continuous space of external memory respectively.
For page data, source address is named Saddr0, specified in MTRANS_SADDR0. The space can be divided into many continuous units, and the unit size(named PUnit) is 1024 bytes.
For spare data, source address is named Saddr1, specified in MTRANS_SADDR1. The space can be divided into many continuous units, and the unit size(named SUnit) is 64 bytes for 16 bits BCH mode.

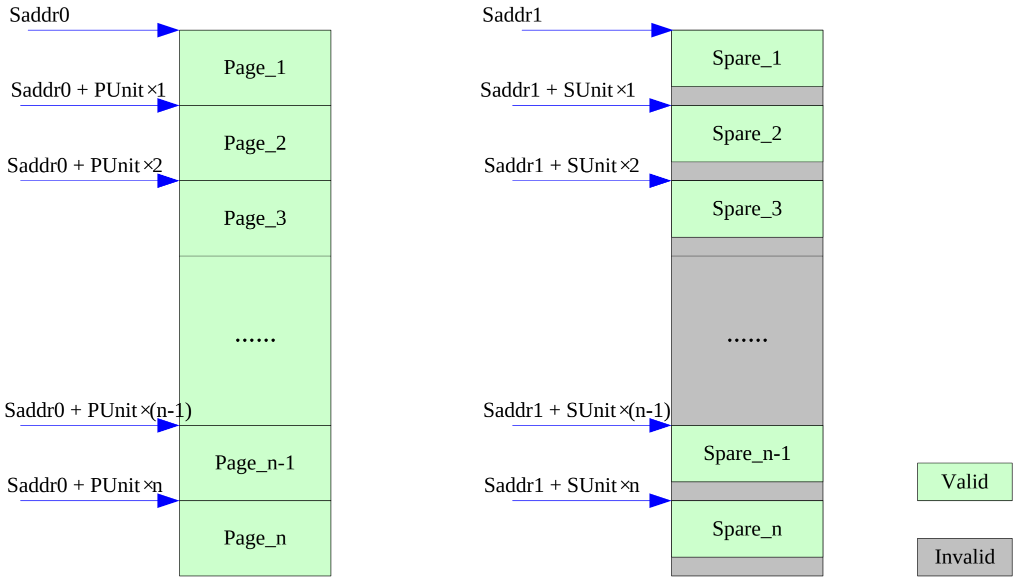


Fig. 12-2 NANDC Address Assignment

The detailed format for page data and spare data in every unit is shown in following figures.

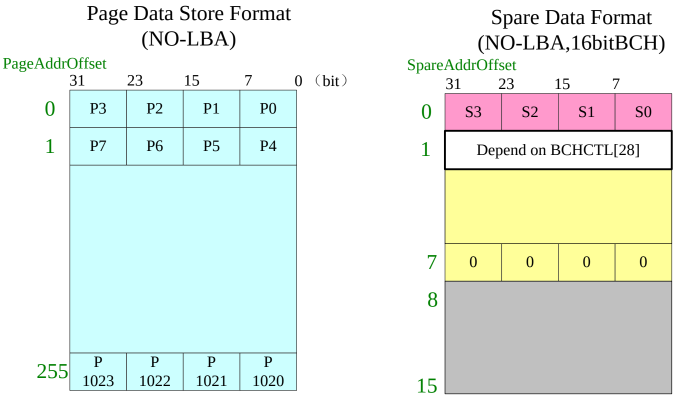


Fig. 12-3 NANDC Data Format

12.6.3 BchPage Application

BCHCTL[16] determines whether codeword size for page data is 1024 bytes or 512 bytes when FLCTL[11] is 0.

1. 1024bytes

When BCHCTL[16]=0, BchPage=0, hardware needs to write 1024 bytes page data and spare data into NAND Flash or read 1024 bytes page data and spare data from NAND Flash. All the 1024 bytes page data and spare data are encoded when writing or decoded when reading.

2. 512bytes

When BCHCTL[16]=1, BchPage=1, hardware needs to write 512 bytes page data and spare data into NAND Flash or read 512 bytes page data and spare data from NAND Flash.

In this mode, the page data unit size for BCH encoder and BCH decoder still is 1024byte. So to support BCH encoder and decoder, software should configure page data as follows:1th~512th bytes are invalid data which must be stuffed with 0xff, 513th~1024th bytes are valid page data.

12.6.4 PageSize/SpareSize Application

1. Big Page

When FLCTL[11]=0(LbaEn=0), the NAND Flash to be operated is Raw NAND Flash. Every codeword size is 1024 bytes and FLCTL[21] should always be set to 0, and the PageStep in external memory is 1024 bytes if bus mode is master mode.

At this mode, the spare size and SpareStep in external memory are determined by BCH Mode as follows:

BCH Mode=16bitBCH: spare size=(28+4)bytes , SpareStep=64bytes

2. Small Page

The NAND Flash controller will not operate in LBA mode, so small page with 512 bytes for a codeword is not supported.

12.6.5 NANDC Interrupt Application

NANDC has 1 interrupt output signal and 5 interrupt sources: dma finish interrupt source, NAND Flash ready interrupt source, bch error interrupt source, bch fail interrupt source and master idle interrupt source. When one or more of these interrupt source are enabled, NANDC interrupt is asserted if one or more interrupt source is high. Software can determine the interrupt source by reading INTST and clear interrupt by writing corresponding bit in INTCLR.

12.6.6 FIFO Application

FIFO in NANDC is used to store command, address and data temporarily that are intend to write to the external NAND Flash. The FIFO is 32-bit wide and 8-location deep, user can access it by configuring the NANDC to work on bypass(FLCTL[3]=1) and fifo enable(FLCTL[30]=1) mode.

The format of data written into fifo is as follows.

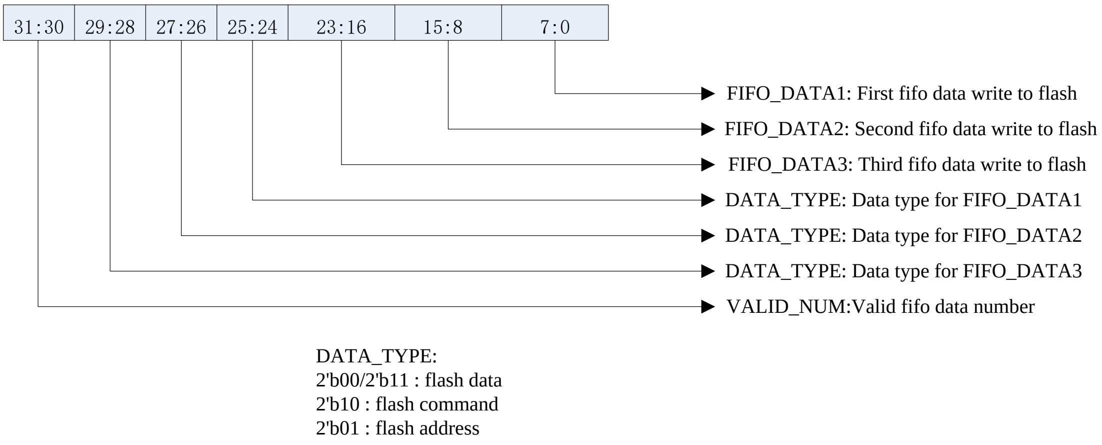

31:30
29:28
27:26
25:24
23:16
15:8
7:0
FIFO_DATA1: First fifo data write to flash
FIFO_DATA2: Second fifo data write to flash
FIFO_DATA3: Third fifo data write to flash
DATA_TYPE: Data type for FIFO_DATA1
DATA_TYPE: Data type for FIFO_DATA2
DATA_TYPE: Data type for FIFO_DATA3
VALID_NUM:Valid fifo data number
DATA_TYPE:
2'b00/2'b11 : flash data
2'b10 : flash command
2'b01 : flash address

Fig. 12-4 FIFO Data Format

Command, address or data write to external NAND Flash are pushed into fifo first and pop up automatically if the fifo is not empty. Then FIFO_DATA1, FIFO_DATA2 and FIFO_DATA3 are written to NAND Flash by NANDC in order. The method to determine the end of write operation is to read register FMCTL. If FMCTL[10] is set , it means that the fifo is empty and NAND Flash write operation is over.

Chapter 13 Embedded SRAM

13.1 Overview

There is only one embedded SRAM, SYSTEM_SRAM. The AXI slave device, which supports read and write access to provide system fast access data storage.

13.1.1 Features supported

- SYSTEM_SRAM
  - Provide 256KB 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)
  - Shares space from 32KB~256KB with vad

13.2 Block Diagram

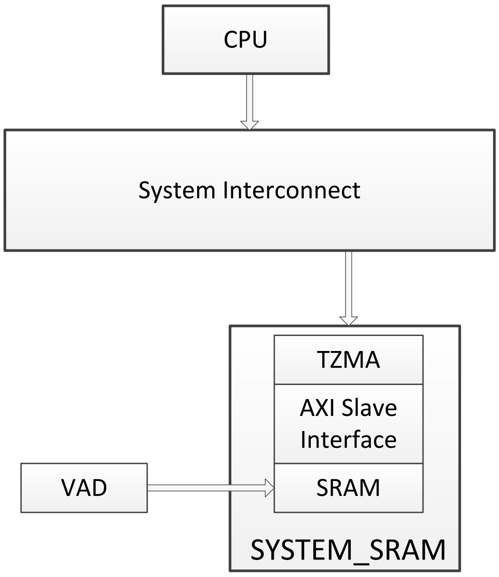


Fig. 13-1 Embedded SRAM block diagram

13.3 Function Description

13.3.1 AXI slave interface of SYSTEM_SRAM

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

13.3.2 Embedded SRAM access path

The SYSTEM_SRAM can only be accessed by CPU, DMAC0, DMAC1, CRYPTO and VOP.VAD can access SYSTEM_SRAM from 32KB to 256KB, and vad only write to SYSTEM_SRAM, and will not read from SYSTEM_SRAM.VAD can write to SYSTEM_SRAM when cpu read from/write to SYSTEM_SRAM.

Chapter 14 System Debug

14.1 Overview

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

14.1.1 Features

- Invasive debug with core halted
- SW-DP

14.1.2 Debug components address map

Base Address: 0xFF800000.
The following table shows the debug components address in memory map:

Table 14-1 Debug Address Mapping Table

Module	Address Offset
APB ROM table for the processor	0x00000 - 0x00FFF
Reserved for other debug components	0x01000 - 0x07FFF
Reserved for future expansion	0x08000 - 0x0FFFF
Core 0 Debug	0x10000 - 0x10FFF
Core 0 PMU	0x11000 - 0x11FFF
Core 1 Debug	0x12000 - 0x12FFF
Core 1 PMU	0x13000 - 0x13FFF
Core 2 Debug	0x14000 - 0x14FFF
Core 2 PMU	0x15000 - 0x15FFF
Core 3 Debug	0x16000 - 0x16FFF
Core 3 PMU	0x17000 - 0x17FFF
Core 0 CTI	0x18000 - 0x18FFF
Core 1 CTI	0x19000 - 0x19FFF
Core 2 CTI	0x1A000 - 0x1AFFF
Core 3 CTI	0x1B000 - 0x1BFFF
Core 0 Trace	0x1C000 - 0x1CFFF
Core 1 Trace	0x1D000 - 0x1DFFF
Core 2 Trace	0x1E000 - 0x1EFFF
Core 3 Trace	0x1F000 - 0x1FFFF

14.2 Block Diagram

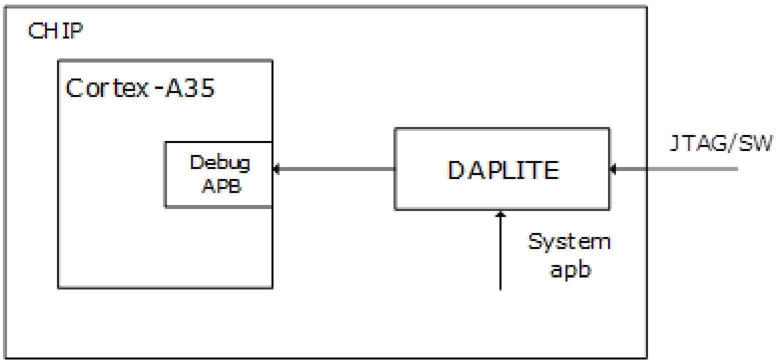


Fig. 14-1 Debug system structure

14.3 Function Description

14.3.1 DAP

The DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)

- ROM table

The debug port is the host tools interface to access the DAP-Lite. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite supports a combined debug port which includes both JTAG and Serial Wire Debug(SWD), with a mechanism that supports switching between them.

The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.

The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit[31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.

More information please refer to the document CoreSight_DAPLite_TRM.pdf for the debug detail description.

14.4 Register Description

Please refer to the document CoreSight DAP-Lite Technical Reference Manual for the debug detail description.

14.5 Interface Description

14.5.1 DAP SWJ-DP Interface

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

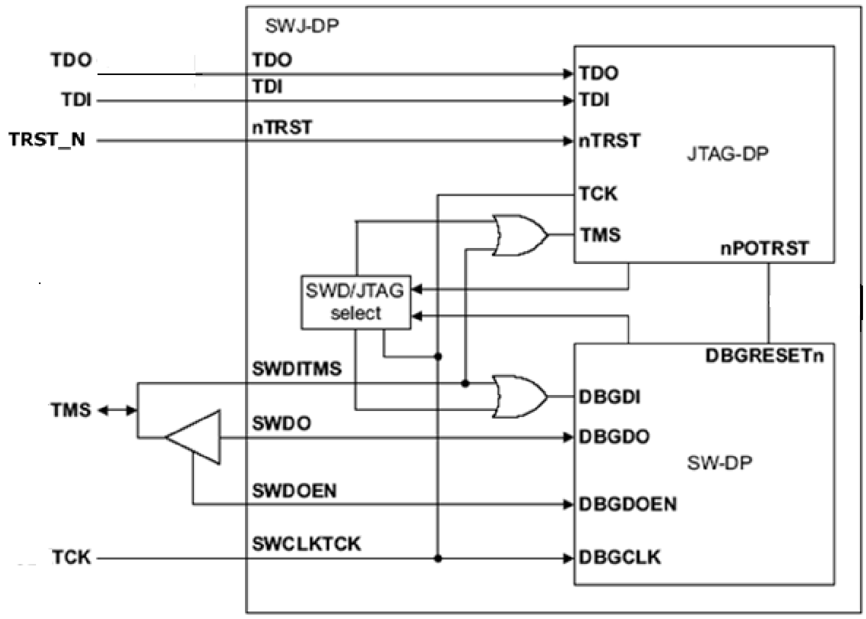


Fig. 14-2 DAP SWJ interface

14.5.2 DAP SW-DP Interface

This implementation is taken from ADIv5.1 and operates with a synchronous serial interface. This uses a single bidirectional data signal, and a clock signal.

The figure below describes the interaction between the timing of transactions on the serial wire interface, and the DAP internal bus transfers. It shows when the target responds with a

WAIT acknowledgement.

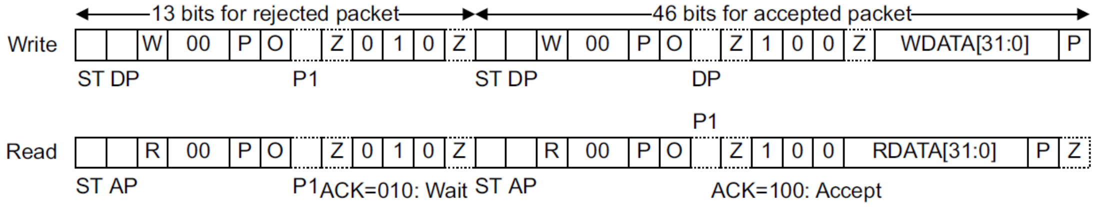


Fig. 14-3 SW-DP acknowledgement timing

Table 14-2 SW-DP Interface Description

Module pin	Direction	PIN name	IOMUX
jtag_tck	I	GPIO1_C6/UART1_CTSN/UART2_RX_M0/SPI2_MISO/JTAG_TCK	GRF_GPIO1C_IOMUX_H[7:4]=4'h4
jtag_tms	I/O	GPIO1_C7/UART1_RTSN/UART2_TX_M0/SPI2_MOSI/JTAG_TMS	GRF_GPIO1C_IOMUX_H[11:8]=4'h4

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

Chapter 15 UART

15.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.
UART Controller supports the following features:

- Support 5 independent UART controller: UART0~UART4
- contain two 64Bytes FIFOs for data receive and transmit
- support auto flow-control
- Support bit rates 115.2Kbps,460.8Kbps,921.6Kbps,1.5Mbps,3Mbps, 4Mbps
- Support programmable baud rates, even with non-integer clock divider
- Standard asynchronous communication bits (start, stop and parity)
- Support interrupt-based or DMA-based mode
- Support 5-8 bits width transfer

15.2 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

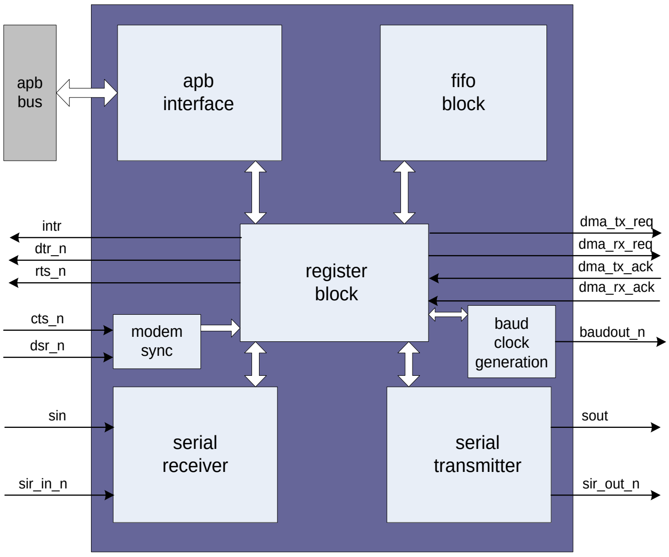


Fig. 15-1UART Block Diagram

APB INTERFACE
The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus widths of 8, 16, and 32 bits.

Register block
Be responsible for the main UART functionality including control, status and interrupt generation.

Modem Synchronization block
Synchronizes the modem input signal.

FIFO block
Be responsible for FIFO control and storage (when using internal RAM) or signaling to

control external RAM (when used).

Baud Clock Generator
Generates the transmitter and receiver baud clockalong with the output reference clock signal (baudout_n).

Serial Transmitter
Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.

Serial Receiver
Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

15.3 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.

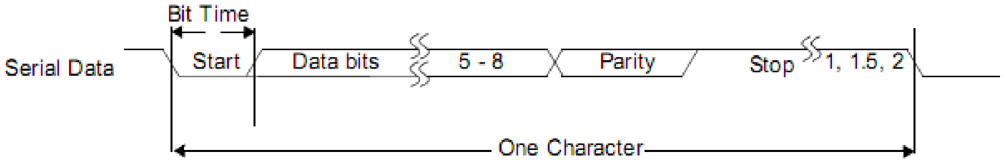


Fig. 15-2UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional datacommunications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

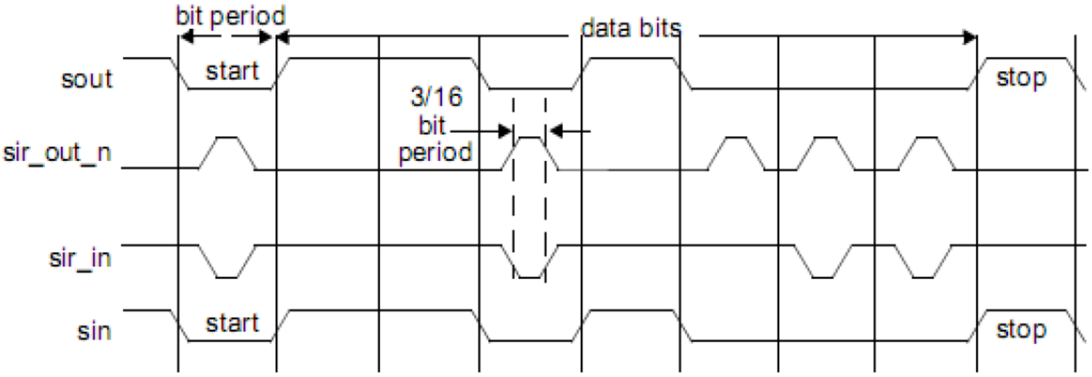


Fig. 15-3IrDA 1.0

Baud Clock

The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

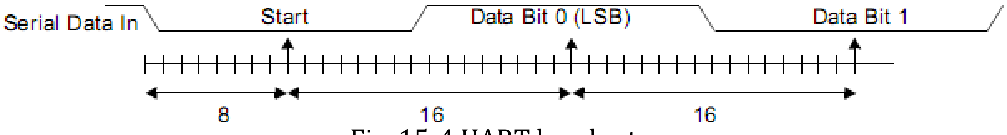


Fig. 15-4 UART baud rate

FIFO Support

1. NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2. FIFO MODE

The FIFO depth of UART0/UART1/UART2/UART3/UART4/UART5 is 64bytes. The FIFO mode of all the UART is enabled by register FCR[0].

Interrupts

The following interrupt types can be enabled with the IER register.

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode)
- Modem Status

DMA Support

The UART supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

- When the Transmitter Holding Register is empty in non-FIFO mode.
- When the transmitter FIFO is empty in FIFO mode with Programmable THRE interrupt mode disabled.
- When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

- When there is a single character available in the Receive Buffer Register in non-FIFO mode.
- When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected, it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

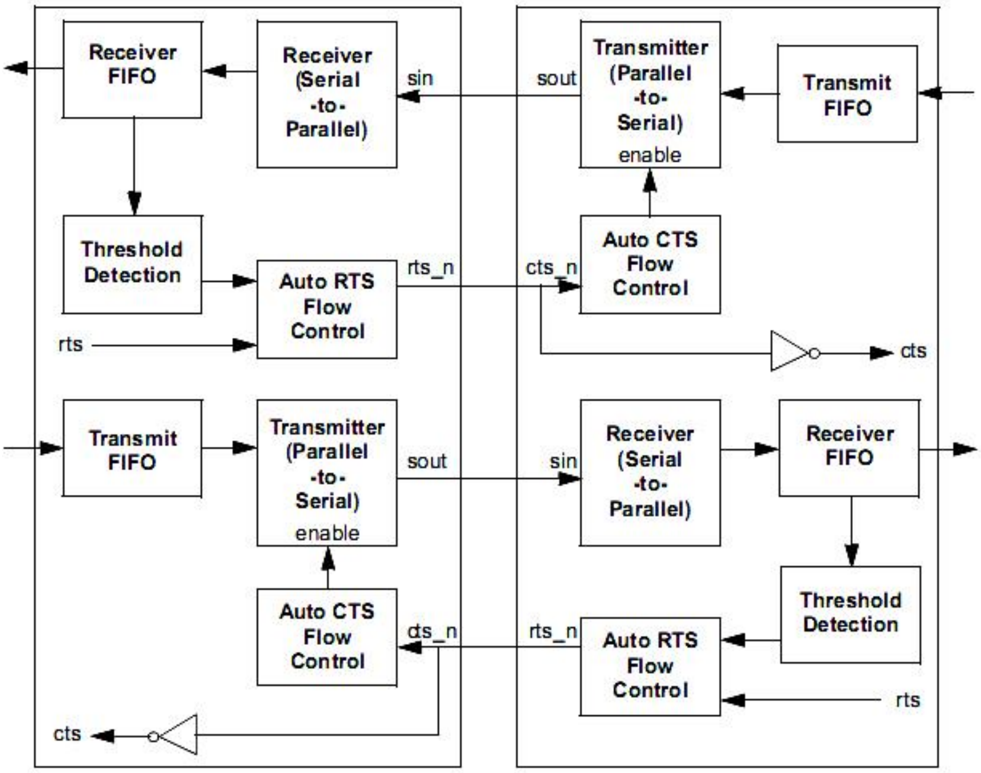


Fig. 15-5 UART Auto flow control block diagram

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)

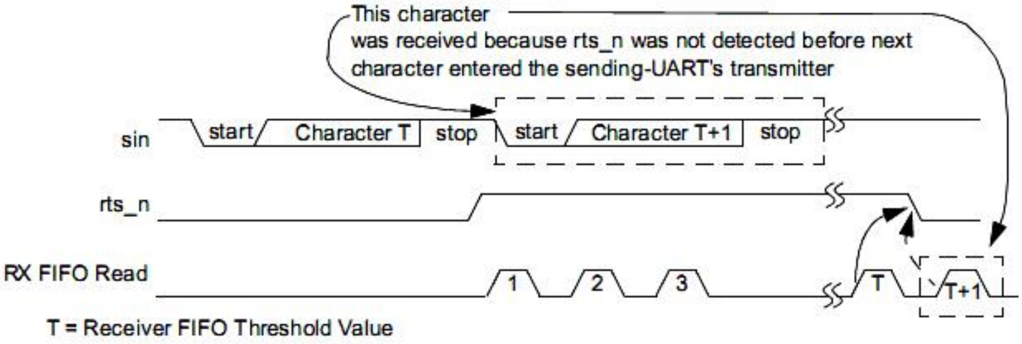

This character was received because rts_n was not detected before next character entered the sending-UART's transmitter
sin
start
Character T
stop
start
Character T+1
stop
rts_n
RX FIFO Read
1
2
3
T
T+1
T = Receiver FIFO Threshold Value

Fig. 15-6 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)

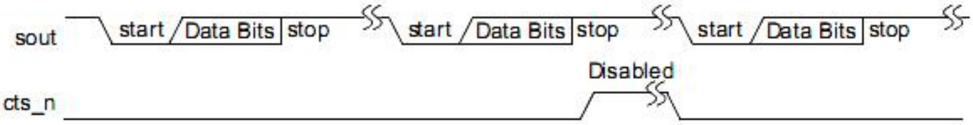


Fig. 15-7 UART AUTO CTS TIMING

15.4 Register Description

15.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000001	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000060	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000006	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x3330382a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

15.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	data_input Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_THR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	data_output Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0x00	baud_rate_divisor_L Lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest Uart clock should be allowed to pass before transmitting or receiving data.

UART_DLH

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	baud_rate_divisor_H Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	prog_thre_int_en Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 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

Bit	Attr	Reset Value	Description
1	RW	0x0	trans_hold_empty_int_en Enable Transmit Holding Register Empty Interrupt
0	RW	0x0	receive_data_available_int_en Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 1'b0: disabled 1'b1: enabled

UART_IIR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 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)

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 ful
5:4	WO	0x0	tx_empty_trigger TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 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 1
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit.
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected. Note that this bit is 'self-clearing'. It is not necessary to clear this bit.

Bit	Attr	Reset Value	Description
0	WO	0x0	fifo_en FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	div_lat_access Divisor Latch Access Bit. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.
6	RW	0x0	break_ctrl Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
5	RO	0x0	reserved
4	RW	0x0	even_parity_sel Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked
3	RW	0x0	parity_en Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 1'b0: parity disabled 1'b1: parity enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	stop_bits_num Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, twostop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit.
1:0	RW	0x0	data_length_sel Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits

UART_MCR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	sir_mode_en 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

Bit	Attr	Reset Value	Description
3	RW	0x0	out2 OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)
2	RW	0x0	out1 OUT1
1	RW	0x0	req_to_send Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data.
0	RW	0x0	data_terminal_ready Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 1'b0: dtr_n de-asserted (logic 1) 1'b1: dtr_n asserted (logic 0)

UART_LSR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	receiver_fifo_error Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 1'b0: no error in RX FIFO 1'b1: error in RX FIFO
6	RO	0x1	trans_empty Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.

Bit	Attr	Reset Value	Description
5	RO	0x1	trans_hold_reg_empty Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting
4	RO	0x0	break_int Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	RO	0x0	framing_error Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2	RO	0x0	parity_eror Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	RO	0x0	overrun_error Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read.
0	RO	0x0	data_ready Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 1'b0: no data ready 1'b1: data ready

UART_MSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	data_carrior_detect Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n.
6	RO	0x0	ring_indicator Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.
4	RO	0x0	clear_to_send Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	RO	0x0	delta_data_carrier_detect Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.
2	RO	0x0	trailing_edge_ring_indicator Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	RO	0x0	delta_data_set_ready Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	RO	0x0	delta_clear_to_send Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space

UART_SRBR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR.

UART_FAR

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	fifo_access_test_en This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 1'b0: FIFO access mode disabled 1'b1: FIFO access mode enabled

UART_TFR

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RO	0x00	trans_fifo_read Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO

UART_RFW

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	receive_fifo_framing_error Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	WO	0x0	receive_fifo_parity_error Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
7:0	WO	0x00	receive_fifo_write Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR

UART_USR

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	receive_fifo_full Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 1'b0: Receive FIFO not full 1'b1: Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.

Bit	Attr	Reset Value	Description
3	RO	0x0	receive_fifo_not_empty Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 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.
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. 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)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	trans_fifo_level Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	receive_fifo_level Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset UART Reset. This asynchronously resets the Uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_dma_mode Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	shadow_rcvr_trigger Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	shadow_tx_empty_trigger Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 1'b0: Halt TX disabled 1'b1: Halt TX enabled

UART_DMASA

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_CPR

Address: Operational Base + offset (0x00f4)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 8'h00: 0 8'h01: 16 8'h02: 32 to 8'h80: 2048 8'h81- 8'hff: reserved
15:14	RO	0x0	reserved
13	RO	0x0	DMA_EXTRA 1'b0: FALSE 1'b1: TRUE
12	RO	0x0	UART_ADD_ENCODED_PARAMS 1'b0: FALSE 1'b1: TRUE
11	RO	0x0	SHADOW 1'b0: FALSE 1'b1: TRUE
10	RO	0x0	FIFO_STAT 1'b0: FALSE 1'b1: TRUE
9	RO	0x0	FIFO_ACCESS 1'b0: FALSE 1'b1: TRUE
8	RO	0x0	NEW_FEAT 1'b0: FALSE 1'b1: TRUE
7	RO	0x0	SIR_LP_MODE 1'b0: FALSE 1'b1: TRUE
6	RO	0x0	SIR_MODE 1'b0: FALSE 1'b1: TRUE
5	RO	0x0	THRE_MODE 1'b0: FALSE 1'b1: TRUE
4	RO	0x0	AFCE_MODE 1'b0: FALSE 1'b1: TRUE
3:2	RO	0x0	reserved
1:0	RO	0x0	APB_DATA_WIDTH 2'b00: 8 bits 2'b01: 16 bits 2'b10: 32 bits 2'b11: reserved

UART_UCV
Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:0	RO	0x3330382a	ver ASCII value for each number in the version.

UART_CTR
Address: Operational Base + offset (0x00fc)

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

15.5 Interface Description

Table 15-1 UART Interface Description

Modulepin	Dir	Pin name	IOMUX
UART0 Interface			
uart0_sin	I	GPIO2_A0/UART0_RX/SPI0_MISO	GRF_GPIO2A_IOMUX[1:0]=2’b01
uart0_sout	O	GPIO2_A1/UART0_TX/SPI0_MOSI	GRF_GPIO2A_IOMUX[3:2]=2’b01
uart0_cts_n	I	GPIO2_A2/UART0_CTSN/SPI0_CLK/I2C2_SDA	GRF_GPIO2A_IOMUX[5:4]=2’b01
uart0_rts_n	O	GPIO2_A3/UART0_RTSN/SPI0_CSN0/I2C2_SCL	GRF_GPIO2A_IOMUX[7:6]=2’b01
UART1 Interface			
uart1_sin	I	GPIO1_D0/UART1_RX/I2C0_SDA/SPI2_CLK	GRF_GPIO1D_IOMUX[1:0]= 2’b01
uart1_sout	O	GPIO1_D1/UART1_TX/I2C0_SCL/SPI2_CSN0	GRF_GPIO1D_IOMUX[3:2]= 2’b01
uart1_cts_n	I	GPIO1_C6/UART1_CTSN/UART2_RX_M0/SPI2_MISO/JTAG_TCK	GRF_GPIO1C_IOMUX_H[7:4]= 4’b0001
uart1_rts_n	O	GPIO1_C7/UART1_RTSN/UART2_TX_M0/SPI2_MOSI/JTAG_TMS	GRF_GPIO1C_IOMUX_H[11:8]=4’b0001
UART2m0 Interface			
uart2m0_sin	I	GPIO1_C6/UART1_CTSN/UART2_RX_M0/SPI2_MISO/JTAG_TCK	GRF_GPIO1C_IOMUX_H[7:4]=4’b0010
uart2m0_sout	O	GPIO1_C7/UART1_RTSN/UART2_TX_M0/SPI2_MOSI/JTAG_TMS	GRF_GPIO1C_IOMUX_H[11:8]=4’b0010
UART2m1 Interface			
uart2m1_sin	I	GPIO4_D2/SDMMC_D2/UART2_RX_M1	GRF_GPIO4D_IOMUX[5:4]=2’b10
uart2m1_sout	O	GPIO4_D3/SDMMC_D3/UART2_TX_M1	GRF_GPIO4D_IOMUX[7:6]=2’b10
UART3 Interface			

Modulepin	Dir	Pin name	IOMUX
uart3_sin	I	GPIO3_B4/FLASH_RDY/I2C3_SDA_M1/SPI1_MOSI/UART3_RX	GRF_GPIO3B_IOMUX[11:8]=4'b0100
uart3_sout	O	GPIO3_B5/FLASH_CSN0/I2C3_SCL_M1/SPI1_CSN0/UART3_TX	GRF_GPIO3B_IOMUX[15:12]=4'b0100
UART4 Interface			
uart4_sin	I	GPIO4_B0/UART4_RX	GRF_GPIO4B_IOMUX[1:0]=2'b01
uart4_sout	O	GPIO4_B1/UART4_TX	GRF_GPIO4B_IOMUX[3:2]= 2'b01
uart4_cts_n	I	GPIO4_A6/UART4_CTSN	GRF_GPIO4A_IOMUX[13:12]=2'b01
uart4_rts_n	O	GPIO4_A7/UART4_RTSN	GRF_GPIO4A_IOMUX[15:14]=2'b01

The I/O interface of UART2 can be chosen by setting GRF_SOC_CON5[3:2]bit, if those bit is set to 2'b01, UART2 uses the UART2m1 I/O interface,if those bit is set to 2'b10,UART2 uses the internal usbphy uart debug port.

15.6 Application Notes

15.6.1 None FIFO Mode Transfer Flow

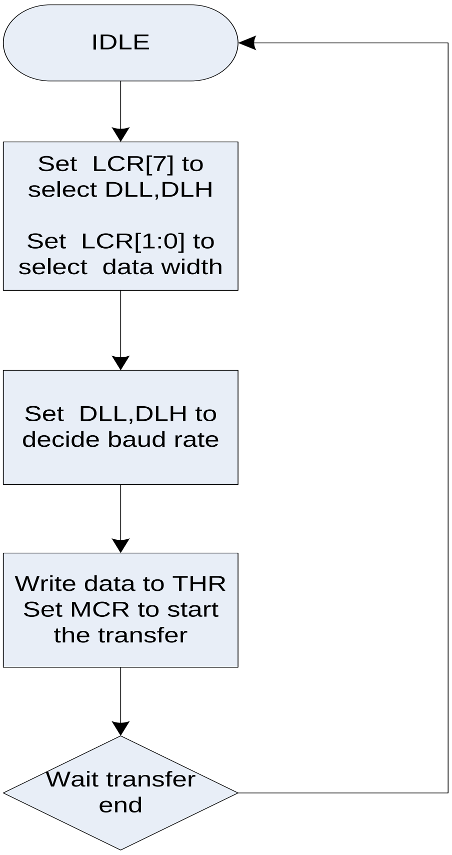

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set DLL,DLH to decide baud rate
Write data to THR
Set MCR to start the transfer
Wait transfer end

Fig. 15-8 UART none fifo mode

15.6.2 FIFO Mode Transfer Flow

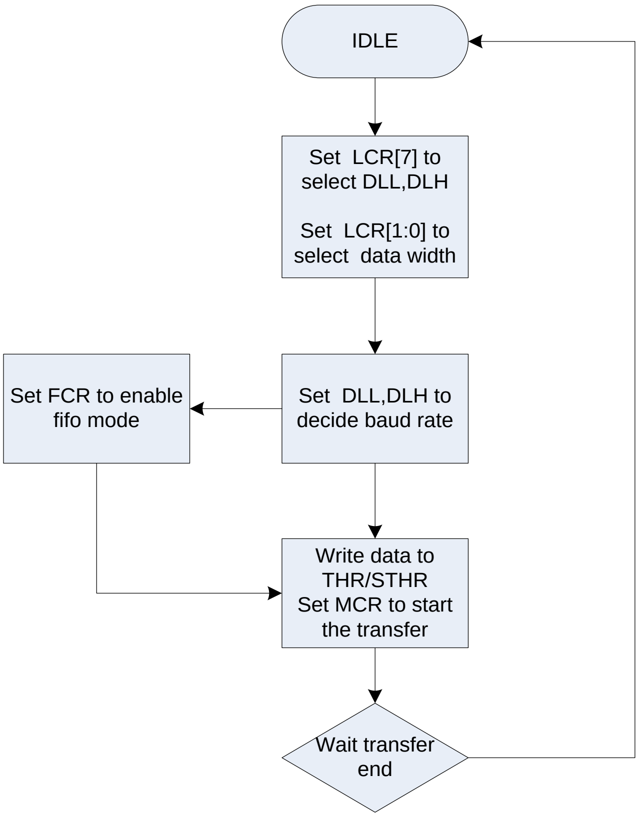

IDLE
Set LCR[7] to select DLL,DLH
Set LCR[1:0] to select data width
Set FCR to enable fifo mode
Set DLL,DLH to decide baud rate
Write data to THR/STHR
Set MCR to start the transfer
Wait transfer end

Fig. 15-9 UART Serial protocol

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.

15.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.UART0~4 source clocks can be selected from five PLL outputs (XIN_OSC0_FUNC/DPLL_CLK_MUX/USBPHY_480M/ VPLL1_CLK_MUX/VPLL0_MUX).
UART clocks can be generated by 1 to 32 division of its source clock, or can be fractionally divided again.

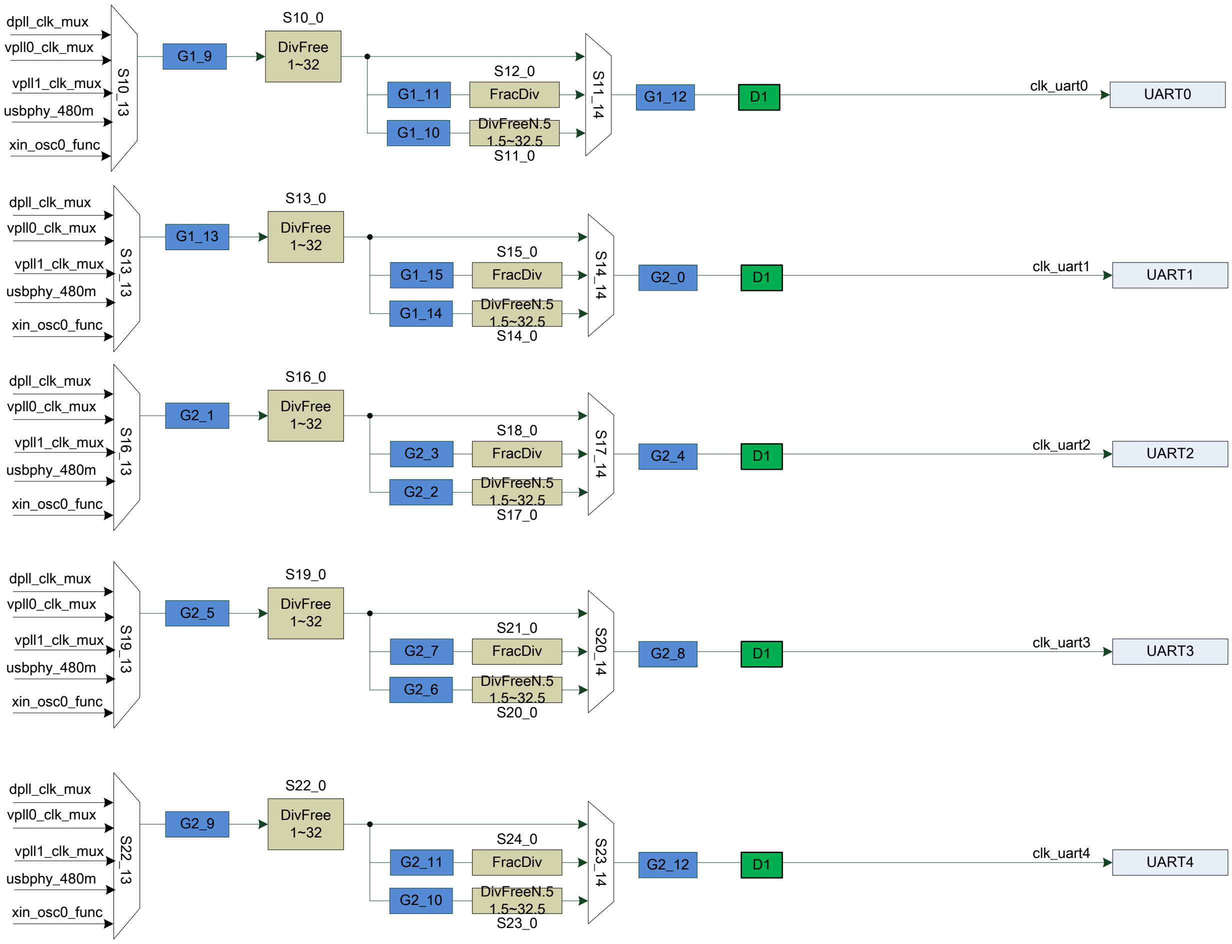


Fig. 15-10 UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 15-2 UART baud rate configuration

Baud Rate	Reference Configuration

115.2 Kbps	Configure DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/72 to get 1.8432MHz clock;
	Configure UART_DLL to 1.

460.8 Kbps	Configure DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/288 to get 7.3728MHz clock;
	Configure UART_DLL to 1.

921.6 Kbps	Configure DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 46875/576 to get 14.7456MHz clock;
	Configure UART_DLL to 1.

1.5 Mbps	Choose DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 50 to get 24MHz clock;
	Configure UART_DLL to 1.

3 Mbps	Choose DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 1200/48 to get 48MHz clock;
	Configure UART_DLL to 1.

4 Mbps	Configure DPLL to get 1200MHz clock output;
	Divide 1200MHz clock by 480/7.5 to get 64MHz clock;
	Configure UART_DLL to 1.

15.6.4 CTS_n and RTS_n Polarity Configurable

The polarity of cts_n and rts_n ports can be configured by GRF registers.

- When grf_uart_cts_sel[*] is configured as 1'b1, cts_n is high active. Otherwise, lowactive.
- When grf_uart_rts_sel[*] is configured as 1'b1, rts_n is high active. Otherwise, lowactive.

Table 15-3 UART baud rate configuration

UART	GRF_UART_CTS_SEL	GRF_UART_RTS_SEL
UART0	GRF_SOC_CON3[0]	GRF_SOC_CON3[5]
UART1	GRF_SOC_CON3[1]	GRF_SOC_CON3[6]
UART2	GRF_SOC_CON3[2]	GRF_SOC_CON3[7]
UART3	GRF_SOC_CON3[3]	GRF_SOC_CON3[8]
UART4	GRF_SOC_CON3[4]	GRF_SOC_CON3[9]

15.6.5 UART tx dma_req enable signal Configurable

Because uart tx dma_req signal auto drive high after system power up,and DMAC will take it as a valid request, so RK3308 add a grf bit to mask uart tx dma_req when system power up.

●When grf_uart_dma_req_ctrl[*] is configured as 1'b0, dma_req is active. Otherwise, dma_req is masked.This bit default value is 1, and it must be set to 0 after uart is configured, otherwise, uart will not work correctly.

Table 15-4 UART dma_req enable configuration

UART	GRF_UART_DMA_REQ_CTRL
UART0	GRF_SOC_CON3[10]
UART1	GRF_SOC_CON3[11]
UART2	GRF_SOC_CON3[12]
UART3	GRF_SOC_CON3[13]
UART4	GRF_SOC_CON3[14]

Chapter 16 Pulse Width Modulation (PWM)

16.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.

The PWM Module supports the following features:

- 4-built-in PWM channels
- Configurable to operate in capture mode
  - Measures the high/low polarity effective cycles of this input waveform
  - Generates a single interrupt at the transition of input waveform polarity
  - 32-bit high polarity capture register
  - 32-bit low polarity capture register
  - 32-bit current value register
  - The capture result can be stored in a FIFO, and the depth of FIFO is 8. The data of FIFO can be read by CPU or DMA
  - Channel 3 support 32-bits power key capture mode
  - Support a input filter to remove glitch
- Configurable to operate in continuous mode or one-shot mode
  - 32-bit period counter
  - 32-bit duty register
  - 32-bit current value register
  - Configurable PWM output polarity in inactive state and duty period pulse polarity
  - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
  - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
  - 8-bit repeat counter for one-shot operation. One-shot operation will produce N + 1 periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
  - Continuous mode generates the waveform continuously, and does not generates any interrupts
- pre-scaled operation to clk_pwm and then further scaled
- Available low-power mode to reduce power consumption when the channel is inactive.

16.2 Block Diagram

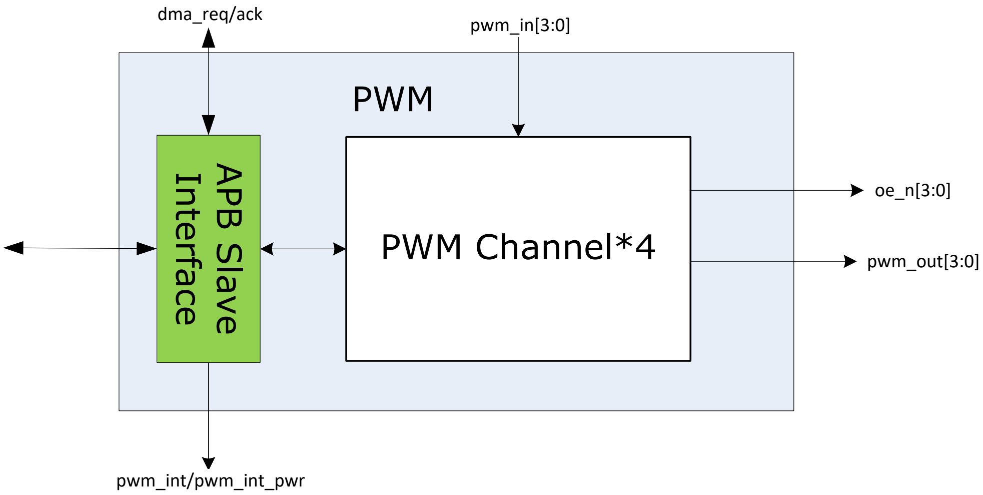


Fig. 16-1 PWM Block Diagram

The host processor gets access to PWM Register Block through the APB slave interface with 32-bit bus width, and asserts the active-high level interrupt. PWM only supports one interrupt output, please refer to interrupt register to know the raw interrupt status when an

interrupt is asserted.
PWM Channel is the control logic of PWM module, and controls the operation of PWM module according to the configured working mode.

16.3 Function Description

The PWM supports three operation modes: capture mode, one-shot mode and continuous mode. For the one-shot mode and the continuous mode, the PWM output can be configured as the left-aligned mode or the center-aligned mode.

16.3.1 Capture mode

The capture mode is used to measure the PWM channel input waveform high/low effective cycles with the PWM channel clock, and asserts an interrupt when the polarity of the input waveform changes. The number of the high effective cycles is recorded in the PWMx_PERIOD_HPC register, while the number of the low effective cycles is recorded in the PWMx_DUTY_LPC register.

Notes: the PWM input waveform is doubled buffered when the PWM channel is working in order to filter unexpected shot-time polarity transition, and therefore the interrupt is asserted several cycles after the input waveform polarity changes, and so does the change of the values of PWMx_PERIOD_HPC and PWMx_DUTY_LPC.

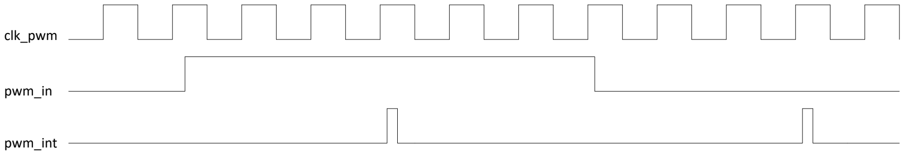


Fig. 16-2 PWM Capture Mode

The capture result also can be stored in a FIFO. The FIFO has an almost full indicator. The indicator can chose to use as an interrupt or DMA request. When it is used as an interrupt, the data in FIFO can be read by CPU. When it is used as a DMA request, the data in FIFO can be read through DMA. It also supports timeout interrupt when the data in FIFO has not been read in a time threshold.
The PWM (only channel 3) support 32-bits power key capture mode. User can configure 10 power key to match, the interrupt will be asserted when the capture value match any one.

16.3.2 Continuous mode

The PWM channel generates a series of the pulses continuously as expected once the channel is enabled with continuous mode.
In the continuous mode, the PWM output waveforms can be in one form of the two output mode: left-aligned mode or center-aligned mode.
For the left-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. After the period number (PWMx_PERIOD_HPC) is reached, the output is again switched to the opposite polarity to start another period of desired pulse.

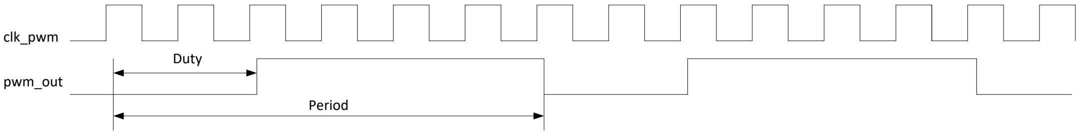


Fig. 16-3 PWM Continuous Left-aligned Output Mode

For the center-aligned output mode, the PWM channel firstly starts the duty cycle with the configured duty polarity (PWMx_CTRL.duty_pol). Once one half of duty cycle number (PWMx_DUTY_LPC) is reached, the output is switched to the opposite polarity. Then if there is one half of duty cycle left for the whole period, the output is again switched to the opposite polarity. Finally after the period number (PWMx_PERIOD_HPC) is reached, the output starts another period of desired pulse.

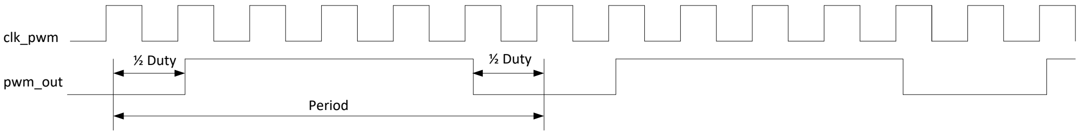


Fig. 16-4 PWM Continuous Center-aligned Output Mode

Once disable the PWM channel, the channel stops generating the output waveforms and output polarity is fixed as the configured inactive polarity (PWMx_CTRL.inactive_pol).

16.3.3 One-shot mode

Unlike the continuous mode, the PWM channel generates the output waveforms within the configured periods (PWM_CTRL.rpt + 1), and then stops. At the same times, an interrupt is asserted to inform that the operation has been finished.

There are also two output modes for the one-shot mode: the left-aligned mode and the center-aligned mode.

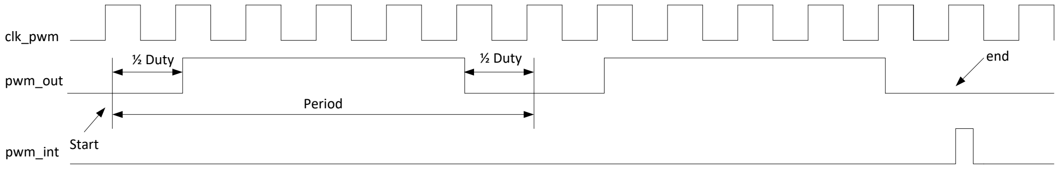


Fig. 16-5 PWM One-shot Center-aligned Output Mode

16.4 Register Description

16.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PWM_PWM0_CNT	0x0000	W	0x00000000	PWM Channel 0 Counter Register
PWM_PWM0_PERIOD_HPR	0x0004	W	0x00000000	PWM Channel 0 Period Register/High Polarity Capture Register
PWM_PWM0_DUTY_LPR	0x0008	W	0x00000000	PWM Channel 0 Duty Register/Low Polarity Capture Register
PWM_PWM0_CTRL	0x000c	W	0x00000000	PWM Channel 0 Control Register
PWM_PWM1_CNT	0x0010	W	0x00000000	PWM Channel 1 Counter Register
PWM_PWM1_PERIOD_HPR	0x0014	W	0x00000000	PWM Channel 1 Period Register/High Polarity Capture Register
PWM_PWM1_DUTY_LPR	0x0018	W	0x00000000	PWM Channel 1 Duty Register/Low Polarity Capture Register
PWM_PWM1_CTRL	0x001c	W	0x00000000	PWM Channel 1 Control Register
PWM_PWM2_CNT	0x0020	W	0x00000000	PWM Channel 2 Counter Register
PWM_PWM2_PERIOD_HPR	0x0024	W	0x00000000	PWM Channel 2 Period Register/High Polarity Capture Register
PWM_PWM2_DUTY_LPR	0x0028	W	0x00000000	PWM Channel 2 Duty Register/Low Polarity Capture Register
PWM_PWM2_CTRL	0x002c	W	0x00000000	PWM Channel 2 Control Register
PWM_PWM3_CNT	0x0030	W	0x00000000	PWM Channel 3 Counter Register

Name	Offset	Size	Reset Value	Description
PWM_PWM3_PERIOD_HPR	0x0034	W	0x00000000	PWM Channel 3 Period Register/High Polarity Capture Register
PWM_PWM3_DUTY_LPR	0x0038	W	0x00000000	PWM Channel 3 Duty Register/Low Polarity Capture Register
PWM_PWM3_CTRL	0x003c	W	0x00000000	PWM Channel 3 Control Register
PWM_INTSTS	0x0040	W	0x00000000	Interrupt Status Register
PWM_INT_EN	0x0044	W	0x00000000	Interrupt Enable Register
PWM_FIFO_CTRL	0x0050	W	0x00000000	PWM Channel 3 FIFO Mode Control Register
PWM_FIFO_INTSTS	0x0054	W	0x00000010	FIFO Interrupts Status Register
PWM_FIFO_TOUTTHR	0x0058	W	0x00000000	FIFO Timeout Threshold Register
PWM_VERSION_ID	0x005c	W	0x02120b34	PWM Version ID Register
PWM_FIFO	0x0060	W	0x00000000	FIFO Register
PWM_PWRMATCH_CTRL	0x0080	W	0x00000000	Power Key Match Control Register
PWM_PWRMATCH_LPRE	0x0084	W	0x238c22c4	Power Key Match Of Low Preload Register
PWM_PWRMATCH_HPRE	0x0088	W	0x11f81130	Power Key Match Of High Preload Register
PWM_PWRMATCH_LD	0x008c	W	0x029401cc	Power Key Match Of Low Data Register
PWM_PWRMATCH_HD_ZERO	0x0090	W	0x029401cc	Power Key Match Of High Data For Zero Register
PWM_PWRMATCH_HD_ONE	0x0094	W	0x06fe0636	Power Key Match Of High Data For One Register
PWM_PWRMATCH_VALUE0	0x0098	W	0x00000000	Power Key Match Value 0 Register
PWM_PWRMATCH_VALUE1	0x009c	W	0x00000000	Power Key Match Value 1 Register
PWM_PWRMATCH_VALUE2	0x00a0	W	0x00000000	Power Key Match Value 2 Register
PWM_PWRMATCH_VALUE3	0x00a4	W	0x00000000	Power Key Match Value 3 Register
PWM_PWRMATCH_VALUE4	0x00a8	W	0x00000000	Power Key Match Value 4 Register
PWM_PWRMATCH_VALUE5	0x00ac	W	0x00000000	Power Key Match Value 5 Register
PWM_PWRMATCH_VALUE6	0x00b0	W	0x00000000	Power Key Match Value 6 Register
PWM_PWRMATCH_VALUE7	0x00b4	W	0x00000000	Power Key Match Value 7 Register
PWM_PWRMATCH_VALUE8	0x00b8	W	0x00000000	Power Key Match Value 8 Register
PWM_PWRMATCH_VALUE9	0x00bc	W	0x00000000	Power Key Match Value 9 Register
PWM_PWM3_PWRCAPTURE_VALUE	0x00cc	W	0x00000000	Channel 3 Power Key Capture Value Register
PWM_FILTER_CTRL	0x00d0	W	0x00000000	Filter Control Register

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

16.4.2 Detail Register Description

PWM_PWM0_CNT
Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT The 32-bit indicates current value of PWM Channel 0 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_PERIOD_HPR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_DUTY_LPR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM0_CTRL

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.

Bit	Attr	Reset Value	Description
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel 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	force_clk_en 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en 0: disabled 1: enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 0: disable lock 1: enable lock
5	RW	0x0	output_mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt . 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM1_CNT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_PERIOD_HPR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_DUTY_LPR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM1_CTRL

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel 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	force_clk_en 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en 0: disabled 1: enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 0: disable lock 1: enable lock

Bit	Attr	Reset Value	Description
5	RW	0x0	output_mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol 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 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM2_CNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_PERIOD_HPR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_DUTY_LPR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM2_CTRL

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RW	0x0	clk_sel 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	force_clk_en 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en 0: disabled 1: enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 0: disable lock 1: enable lock
5	RW	0x0	output_mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol 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 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_PWM3_CNT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_PERIOD_HPR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_DUTY_LPR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to (2^32-1).

PWM_PWM3_CTRL

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.

Bit	Attr	Reset Value	Description
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11:10	RO	0x0	reserved
9	RW	0x0	clk_sel 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	force_clk_en 0: disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1: enabled, the clk_pwm to PWM Clock prescale module is always enable.
7	RW	0x0	ch_cnt_en 0: disabled 1: enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 0: disable lock 1: enable lock
5	RW	0x0	output_mode 0: left aligned mode 1: center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 0: negative 1: positive
3	RW	0x0	duty_pol This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 0: negative 1: positive

Bit	Attr	Reset Value	Description
2:1	RW	0x0	pwm_mode 00: One shot mode. PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 01: Continuous mode. PWM produces the waveform continuously 10: Capture mode. PWM measures the cycles of high/low polarity of input waveform. 11: reserved
0	RW	0x0	pwm_en 0: disabled 1: enabled. If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation

PWM_INTSTS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RO	0x0	CH3_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM3_PERIOD_HPR to know the effective high cycle of Channel 3 input waveform. Otherwise, please refer to PWM3_PERIOD_LPR to know the effective low cycle of Channel 3 input waveform. Write 1 to CH3_IntSts will clear this bit.
10	RO	0x0	CH2_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM2_PERIOD_HPR to know the effective high cycle of Channel 2 input waveform. Otherwise, please refer to PWM2_PERIOD_LPR to know the effective low cycle of Channel 2 input waveform. Write 1 to CH2_IntSts will clear this bit.
9	RO	0x0	CH1_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM1_PERIOD_HPR to know the effective high cycle of Channel 1 input waveform. Otherwise, please refer to PWM1_PERIOD_LPR to know the effective low cycle of Channel 1 input waveform. Write 1 to CH1_IntSts will clear this bit.

Bit	Attr	Reset Value	Description
8	RO	0x0	CH0_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7	W1C	0x0	CH3_pwr_IntSts 0: Channel 3 power key Interrupt not generated 1: Channel 3 power key Interrupt generated
6	W1C	0x0	CH2_pwr_IntSts 0: Channel 2 power key Interrupt not generated 1: Channel 2 power key Interrupt generated
5	W1C	0x0	CH1_pwr_IntSts 0: Channel 1 power keyInterrupt not generated 1: Channel 1 power key Interrupt generated
4	W1C	0x0	CH0_pwr_IntSts 0: Channel 0 power key Interrupt not generated 1: Channel 0 power key Interrupt generated
3	W1C	0x0	CH3_IntSts 0: Channel 3 Interrupt not generated 1: Channel 3 Interrupt generated
2	W1C	0x0	CH2_IntSts 0: Channel 2 Interrupt not generated 1: Channel 2 Interrupt generated
1	W1C	0x0	CH1_IntSts 0: Channel 1 Interrupt not generated 1: Channel 1 Interrupt generated
0	W1C	0x0	CH0_IntSts 0: Channel 0 Interrupt not generated 1: Channel 0 Interrupt generated

PWM_INT_EN

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	CH3_pwr_Int_en 0: Channel 3 power key Interrupt disabled 1: Channel 3 power key Interrupt enabled
6:4	RO	0x0	reserved
3	RW	0x0	CH3_Int_en 0: Channel 3 Interrupt disabled 1: Channel 3 Interrupt enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	CH2_Int_en 0: Channel 2 Interrupt disabled 1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en 0: Channel 1 Interrupt disabled 1: Channel 1 Interrupt enabled
0	RW	0x0	CH0_Int_en 0: Channel 0 Interrupt disabled 1: Channel 0 Interrupt enabled

PWM_FIFO_CTRL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:12	RW	0x0	dma_ch_sel 2'b00: Select PWM0 2'b01: Select PWM1 2'b10: Select PWM2 2'b11: Select PWM3
11	RO	0x0	reserved
10	RW	0x0	dma_ch_sel_en 1'b1: Enable, use dma_ch_sel to select the channel to FIFO mode and DMA mode. 1'b0: Disable, select the channel PWM3 to FIFO mode and DMA mode.
9	RW	0x0	timeout_en fifo timeout enable
8	RW	0x0	dma_mode_en 1'b1: enable 1'b0: disable
7	RO	0x0	reserved
6:4	RW	0x0	almost_full_watermark Almost full Watermark level
3	RW	0x0	watermark_int_en Watermark full interrupt
2	RW	0x0	overflow_int_en When high, an interrupt asserts when the fifo overflow
1	RW	0x0	full_int_en When high, an interrupt asserts when the FIFO is full.
0	RW	0x0	fifo_mode_sel When high, PWM FIFO mode is activated

PWM_FIFO_INTSTS

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x1	fifo_empty_status This bit indicates the FIFO is empty
3	W1C	0x0	timieout_intsts Timeout interrupt
2	W1C	0x0	fifo_watermark_full_intsts This bit indicates the FIFO is Watermark Full
1	W1C	0x0	fifo_overflow_intsts This bit indicates the FIFO is overflow
0	W1C	0x0	fifo_full_intsts This bit indicates the FIFO is full

PWM_FIFO_TOUTTHR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	timeout_threshold FIFO Timeout value(unit pwm clock)

PWM_VERSION_ID

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:24	RW	0x02	main_version 8'h0:Base version 8'h1:Support DMA mode 8'h2:Support DMA mode and Power key mode
23:16	RW	0x12	minor_version
15:0	RW	0x0b34	svn_version

PWM_FIFO

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	pol This bit indicates the polarity of the lower 31-bit counter. 0: Low 1: High
30:0	RO	0x00000000	cycle_cnt This 31-bit counter indicates the effective cycles of high/low waveform.

PWM_PWRMATCH_CTRL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	CH3_pwrkey_int_ctrl 0: Assert interrupt after key capture with power key match 1: Assert interrupt after key capture without power key match
14:12	RO	0x0	reserved
11	RW	0x0	CH3_pwrkey_capture_ctrl 0: Capture the value after interrupt 1: Capture the value directly
10:8	RO	0x0	reserved
7	RW	0x0	CH3_pwrkey_polarity 0: pwm in polarity is positive 1: pwm in polarity is negative
6:4	RO	0x0	reserved
3	RW	0x0	CH3_pwrkey_enable 0: Disabled 1: Enabled
2:0	RO	0x0	reserved

PWM_PWRMATCH_LPRE

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x238c	cnt_max The maximum counter value
15:0	RW	0x22c4	cnt_min The minimum counter value

PWM_PWRMATCH_HPRE

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x11f8	cnt_max The maximum counter value
15:0	RW	0x1130	cnt_min The minimum counter value

PWM_PWRMATCH_LD

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM_PWRMATCH_HD_ZERO

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM_PWRMATCH_HD_ONE

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x06fe	cnt_max The maximum counter value
15:0	RW	0x0636	cnt_min The minimum counter value

PWM_PWRMATCH_VALUE0

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE1

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE2

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE3

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE4

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE5

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE6

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE7

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE8

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWRMATCH_VALUE9

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value Power key match value

PWM_PWM3_PWRCAPTURE_VALUE

Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_FILTER_CTRL

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:4	RW	0x000	filter_number Filter window number

Bit	Attr	Reset Value	Description
3	RW	0x0	CH3_input_filter_enable 0: Disabled 1: Enabled
2	RW	0x0	CH2_input_filter_enable 0: Disabled 1: Enabled
1	RW	0x0	CH1_input_filter_enable 0: Disabled 1: Enabled
0	RW	0x0	CH0_input_filter_enable 0: Disabled 1: Enabled

16.5 Interface Description

Table 16-1 PWM Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
PWM0	I/O	GPIO0_B5/PWM0	GRF_GPIO0B_IOMUX[11:10]=2'b01
PWM1	I/O	GPIO0_B6/PWM1	GRF_GPIO0B_IOMUX[13:12]=2'b01
PWM2	I/O	GPIO0_B7/PWM2/I2C3_SDA_M0	GRF_GPIO0B_IOMUX[15:14]=2'b01
PWM3	I/O	GPIO0_C0/PWM3/I2C3_SCL_M0	GRF_GPIO0C_IOMUX[1:0]=2'b01

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

16.6 Application Notes

16.6.1 PWM Capture Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation.
5. Set PWM_FILTER_CTRL. filter_number, then Enable the PWM_FILTER_CTRL. CHx_input_filter_enable(Optional).
6. Enable the channel by writing '1' to PWM_PWMx_CTRL.pwm_en bit to start the channel.
7. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status. If the corresponding polarity flag is set, turn to PWM_PWMx_PERIOD_HPC register to know the effective high cycles of input waveforms, otherwise turn to PWM_PWMx_DUTY_LPC register to know the effective low cycles.
8. Write '0' to PWM_PWMx_CTRL.pwm_en to disable the channel.

16.6.2 PWM Capture DMA Mode Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Configure the channel 3 to work in the capture mode.
4. Configure the PWM_FIFO_CTRL.dma_mode_en and PWM_FIFO_CTRL.fifo_mode_sel to enable the DMA mode. Configure PWM_FIFO_CTRL.almost_full_watermark at appropriate value.
5. Configure DMAC_BUS to tansfer data from PWM to DDR.

6. Set PWM_FILTER_CTRL. filter_number, then Enable the PWM_FILTER_CTRL. CHx_input_filter_enable(Optional).
7. Enable the channel by writing '1' to PWM_PWMx_CTRL.pwm_en bit to start the channel.
8. When a dma_req is asserted, DMAC_BUS transfer the data of effective high cycles and low cycles of input waveforms to DDR.
9. Write '0' to PWM_PWMx_CTRL.pwm_en to disable the channel.

16.6.3 PWM Power key Capture Mode Standard Usage Flow

1. Set PWM_PWM3_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for clk_pwm by programming PWM_PWM3_CTRL.prescale and PWM_PWM3_CTRL.scale, and select the clock needed by setting PWM_PWM3_CTRL.clk_sel. The clock should be 1 Mhz after division.
3. Configure the channel to work in the capture mode.
4. Enable the PWM_INT_EN.CH3_int_pwr to enable the interrupt generation.
5. Set the PWM_PWRMATCH_VALUE0~9 registers for the 10 power key match value.
6.Set max_cnt and min_cnt of follow register: PWM_PWRMATCH_LPRE, PWM_PWRMATCH_HPRE, PWM_PWRMATCH_LD, PWM_PWRMATCH_HD_ZERO, PWM_PWRMATCH_HD_ONE. It doesn't need to set these registers when the default value can meet the requirement.
7.Set PWM_PWRMATCH_CTRL.CH3_pwrkey_polarity for the polarity of power key signal, the default value is 0. Enable the PWM_PWRMATCH_CTRL.CH3_pwrkey_enable.
8. Set PWM_FILTER_CTRL. filter_number, then Enable the PWM_FILTER_CTRL. CH3_input_filter_enable(Optional).
9. Enable the channel by writing '1' to PWM_PWM3_CTRL.pwm_en bit to start the channel.
10. When an interrupt is asserted, refer to INTSTS register to know the raw interrupt status, and refer to PWM_PWM3_PWRCAPTURE_VALUE to know the power key capture value.
11. Write '0' to PWM_PWM3_CTRL.pwm_en to disable the channel.

16.6.4 PWM One-shot Mode/Continuous Standard Usage Flow

1. Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
2. Choose the prescale factor and the scale factor for pclk by programming PWM_PWMx_CTRL.prescale and PWM_PWMx_CTRL.scale, and select the clock needed by setting PWM_PWMx_CTRL.clk_sel.
3. Choose the output mode by setting PWM_PWMx_CTRL.output_mode, and set the duty polarity and inactive polarity by programming PWM_PWMx_CTRL.duty_pol and PWM_PWMx_CTRL.inactive_pol.
4. Set the PWM_PWMx_CTRL.rpt if the channel is desired to work in the one-shot mode.
5. Configure the channel to work in the one-shot mode or the continuous mode.
6. Enable the PWM_INT_EN.chx_int_en to enable the interrupt generation if if the channel is desired to work in the one-shot mode.
7. If the channel is working in the one-shot mode, an interrupt is asserted after the end of operation, and the PWM_PWMx_CTRL.pwm_en is automatically cleared. Whatever mode the channel is working in, write '0' to PWM_PWMx_CTRL.pwm_en bit to disable the PWM channel.

16.6.5 Low-power Usage Flow

The default value of PWM_PWMx_CTRL.force_clk_en is '0' which make the channel enter the low-power mode. In low-power mode, When the PWM channel is inactive, the clk_pwm to the clock prescale module is gated in order to reduce the power consumption. User can set PWM_PWMx_CTRL.force_clk_en to '1' which will make the channel quit the low-power mode. After the setting, the clk_pwm to the clock prescale module is always enable.

16.6.6 Other notes

When the channel is active to produce waveforms, it is free to program the PWM_PWMx_PERIOD_HPC and PWM_PWMx_DUTY_LPC register. User can use PWM_PWMx_CTRL.conlock to take period and duty effect at the same time. The usage flow is as follow:
1. Set PWM_PWMx_CTRL.conlock to '1'.

2. Set PWM_PWMx_PERIOD_HPC and PWM_PWMx_DUTY_LPC.
3. Set PWM_PWMx_CTRL.conlock to ‘0’, others bits in PWM_PWMx_CTRL should be appropriate.

After above configuration, the change will not take effect immediately until the current period ends.

An active channel can be changed to another operation mode without disable the PWM channel. However, during the transition of the operation mode there may be some irregular output waveforms. So does changing the clock division factor when the channel is active.

Chapter 17 I2C Interface

17.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.

I2C Controller supports the following features:

- Support 4 independent I2C: I2C0, I2C1, I2C2, I2C3
- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- Fiter out glitch on SCL and SDA

17.2 Block Diagram

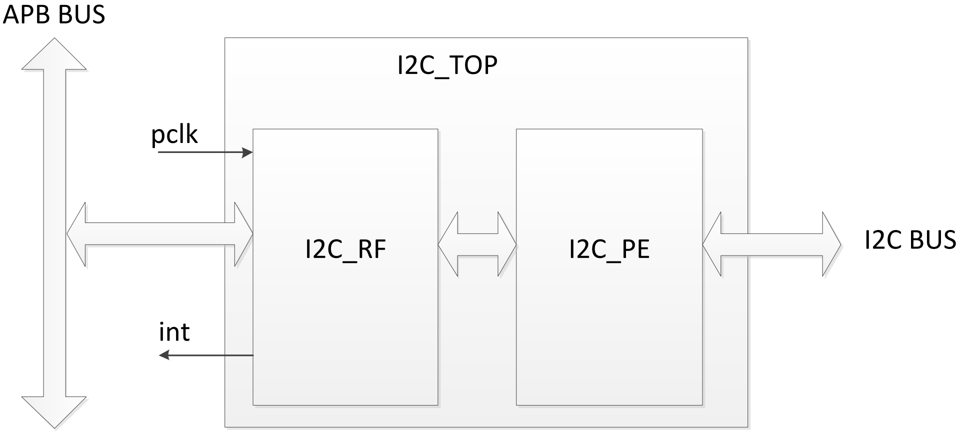


Fig. 17-1 I2C architecture

17.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

17.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

17.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

17.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller supports only Masterfunction. Itsupports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

17.3.1 Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting and configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the working clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

17.3.2 Master Mode Programming

- SCL Clock
  When the I2C controller is programmed in Master mode, the SCL frequency is determined by I2C_CLKDIV register. The SCL frequency is calculated by the following formula:
  SCL Divisor = 8*(CLKDIVL + 1 + CLKDIVH + 1)
  SCL = PCLK/ SCLK Divisor

- Data Receiver Register Access
  When the I2C controller received MRXCNT bytes data， CPU can get the data through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 bytes' data in one transaction.
  When MRXCNT register is written, the I2C controller will start to drive SCL to receive data.

- Transmit Transmitter Register
  Data to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 bytes' data in one transaction. The lower byte will be transmitted first.
  When MTXCNT register is written, the I2C controller will start to transmit data.

- Start Command
  Write 1 to I2C_CON[3], the controller will send I2C start command.

- Stop Command
  Write 1 to I2C_CON[4], the controller will send I2C stop command

- I2C Operation mode
  There are four i2c operation modes.
  - When I2C_CON[2:1] is 2'b00, the controller transmit all valid data in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.
  - When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.
  - When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will trigger clock to read MRXCNT byte data.
  - When I2C_CON[2:1] is 2'b11, the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last, the controller enter receive mode.

- Read/Write Command
  - When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decided by controller itself.

  - In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].
  - In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

- Master Interrupt Condition
  There are 7 interrupt bits in I2C_ISR register related to master mode.
  - Byte transmitted finish interrupt (Bit 0): The bit is asserted when Master completed transmitting a byte.
  - Byte received finish interrupt (Bit 1): The bit is asserted when Master completed receiving a byte.
  - MTXCNT bytes data transmitted finish interrupt (Bit 2): The bit is asserted when Master completed transmitting MTXCNT bytes.
  - MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master completed receiving MRXCNT bytes.
  - Start interrupt (Bit 4): The bit is asserted when Master finished asserting start command to I2C bus.
  - Stop interrupt (Bit 5): The bit is asserted when Master finished asserting stop command to I2C bus.
  - NAK received interrupt (Bit 6): The bit is asserted when Master received a NAK handshake.

- Last byte acknowledge control
  - If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.
  - If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

- How to handle NAK handshake received
  - If I2C_CON[6] is 1, the I2C controller will stop all transactions when NAK handshake received. And the software should take responsibility to handle the problem.
  - If I2C_CON[6] is 0, the I2C controller will ignore all NAK handshake received.

- I2C controller data transfer waveform
  - Bit transferring
    - Data Validity
      The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

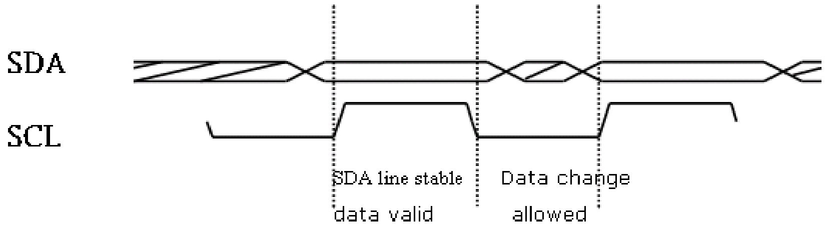


Fig. 17-2 I2C DATA Validity

    - START and STOP conditions
      START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

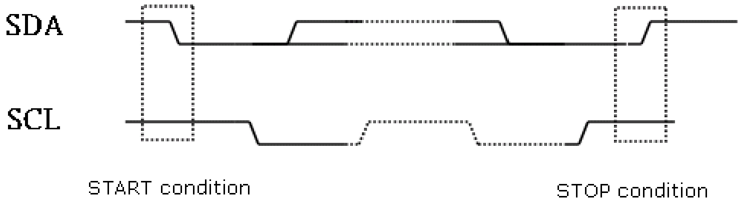


Fig. 17-3 I2C Start and stop conditions

- Data transfer
  - Acknowledge
    After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means "ACK", on the contrary, it's "NOT ACK".

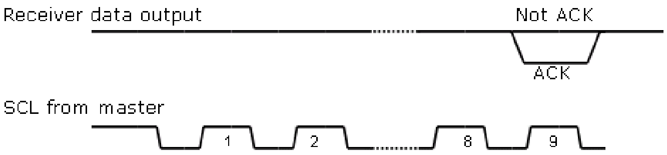


Fig. 17-4 I2C Acknowledge

  - Byte transfer
    The master own I2C bus might initiate multi byte to transfer to a slave.The transfer starts from a "START" command and ends in a "STOP"command. After every byte transfer, the receiver must reply an ACK to transmitter.

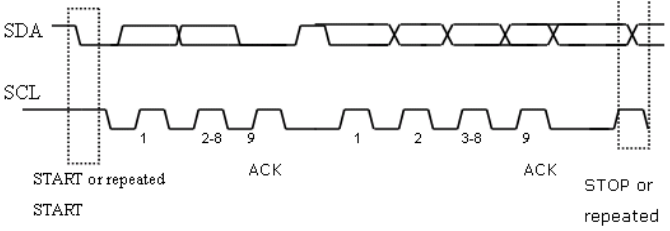


Fig. 17-5 I2C byte transfer

17.4 Register Description

17.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00030300	control register
RKI2C_CLKDIV	0x0004	W	0x00060006	clock divider register，I2C CLK = PCLK / (16*CLKDIV)
RKI2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master rx mode
RKI2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master rx mode
RKI2C_MTXCNT	0x0010	W	0x00000000	master transmit count.specify the total bytes to be transmit (0~32)
RKI2C_MRXCNT	0x0014	W	0x00000000	master rx count.specify the total bytes to be recieved(0~32)
RKI2C_IEN	0x0018	W	0x00000000	interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	interrupt pending register

Name	Offset	Size	Reset Value	Description
RKI2C_FCNT	0x0020	W	0x00000000	finished count: the count of data which has been transmitted or receivedfor debug purpose
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	slave hold debounce configure register
RKI2C_TXDATA0	0x0100	W	0x00000000	I2C tx data register 0
RKI2C_TXDATA1	0x0104	W	0x00000000	I2C tx data register 1
RKI2C_TXDATA2	0x0108	W	0x00000000	I2C tx data register 2
RKI2C_TXDATA3	0x010c	W	0x00000000	I2C tx data register 3
RKI2C_TXDATA4	0x0110	W	0x00000000	I2C tx data register 4
RKI2C_TXDATA5	0x0114	W	0x00000000	I2C tx data register 5
RKI2C_TXDATA6	0x0118	W	0x00000000	I2C tx data register 6
RKI2C_TXDATA7	0x011c	W	0x00000000	I2C tx data register 7
RKI2C_RXDATA0	0x0200	W	0x00000000	I2C rx data register 0
RKI2C_RXDATA1	0x0204	W	0x00000000	I2C rx data register 1
RKI2C_RXDATA2	0x0208	W	0x00000000	I2C rx data register 2
RKI2C_RXDATA3	0x020c	W	0x00000000	I2C rx data register 3
RKI2C_RXDATA4	0x0210	W	0x00000000	I2C rx data register 4
RKI2C_RXDATA5	0x0214	W	0x00000000	I2C rx data register 5
RKI2C_RXDATA6	0x0218	W	0x00000000	I2C rx data register 6
RKI2C_RXDATA7	0x021c	W	0x00000000	I2C rx data register 7
RKI2C_ST	0x0220	W	0x00000003	status debug register
RKI2C_DBGCTRL	0x0224	W	0x00000f00	Debug config register

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

17.4.2 Detail Register Description

RKI2C_CON

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0003	version rki2c version information
15:14	RW	0x0	stop_setup stop setup config: TSU;sto = (stop_setup + 1) * T(SCL_HIGH) + Tclk_i2c
13:12	RW	0x0	start_setup start setup config: TSU;sta = (start_setup + 1) * T(SCL_HIGH) + Tclk_i2c THD;sta = (start_setup + 2) * T(SCL_HIGH) - Tclk_i2c
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	data_upd_st SDA update point config: Used to config sda change state when scl is low, used to adjust setup/hold time 4'bn:Thold = (n + 1) * Tclk_i2c Note: 0 <= n <= 5
7	RO	0x0	reserved
6	RW	0x0	act2nak operation when NAK handshake is received: 1'b0: ignored 1'b1: stop transaction
5	RW	0x0	ack last byte acknowledge control in master receive mode: 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop stop enable, when this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start start enable, when this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode i2c mode select: 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable: 1'b0:not enable 1'b1:enable

RKI2C_CLKDIV

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count: T(SCL_HIGH) = Tclk_i2c * (CLKDIVH + 1) * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count: T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8

RKI2C_MRXADDR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid: 1'b0:invalid 1'b1:valid
25	RW	0x0	addmvld address middle byte valid: 1'b0:invalid 1'b1:valid
24	RW	0x0	addlvld address low byte valid: 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	saddr master address register. the lowest bit indicate write or read 24 bits address register

RKI2C_MRXRADDR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid: 1'b0:invalid 1'b1:valid
25	RW	0x0	sraddmvld address middle byte valid: 1'b0:invalid 1'b1:valid
24	RW	0x0	sraddlvld address low byte valid: 1'b0:invalid 1'b1:valid
23:0	RW	0x000000	sraddr slave register address accessed. 24 bits register address

RKI2C_MTXCNT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mtxcnt master transmit count. 6 bits counter

RKI2C_MRXCNT

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master rx count. 6 bits counter

RKI2C_IEN

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclen slave hold scl interrupt enable: 1'b0:disable 1'b1:enable
6	RW	0x0	nakrcvien NAK handshake received interrupt enable: 1'b0:disable 1'b1:enable
5	RW	0x0	stopien stop operation finished interrupt enable: 1'b0:disable 1'b1:enable
4	RW	0x0	startien start operation finished interrupt enable: 1'b0:disable 1'b1:enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable: 1'b0:disable 1'b1:enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable: 1'b0:disable 1'b1:enable
1	RW	0x0	brfien byte rx finished interrupt enable: 1'b0:disable 1'b1:enable
0	RW	0x0	btfien byte tx finished interrupt enable: 1'b0:disable 1'b1:enable

RKI2C_IPD

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	slavehdsclipd slave hold scl interrupt pending bit: 1'b0:no interrupt available 1'b1:slave hold scl interrupt appear, write 1 to clear
6	W1C	0x0	nakrcvipd NAK handshake received interrupt pending bit: 1'b0:no interrupt available 1'b1:NAK handshake received interrupt appear, write 1 to clear
5	W1C	0x0	stopipd stop operation finished interrupt pending bit: 1'b0:no interrupt available 1'b1:stop operation finished interrupt appear, write 1 to clear
4	W1C	0x0	startipd start operation finished interrupt pending bit: 1'b0:no interrupt available 1'b1:start operation finished interrupt appear, write 1 to clear
3	W1C	0x0	mbrfipd MRXCNT data received finished interrupt pending bit: 1'b0:no interrupt available 1'b1:MRXCNT data received finished interrupt appear, write 1 to clear
2	W1C	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit: 1'b0:no interrupt available 1'b1:MTXCNT data transfer finished interrupt appear, write 1 to clear
1	W1C	0x0	brfipd byte rx finished interrupt pending bit: 1'b0:no interrupt available 1'b1:byte rx finished interrupt appear, write 1 to clear
0	W1C	0x0	btfipd byte tx finished interrupt pending bit: 1'b0:no interrupt available 1'b1:byte tx finished interrupt appear, write 1 to clear

RKI2C_FCNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	fcnt the count of data which has been transmitted or received for debug purpose

RKI2C_SCL_OE_DB

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x20	scl_oe_db slave hold scl debounce. cycles for debounce (unit: Tclk_i2c)

RKI2C_TXDATA0

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0 data0 to be transmitted. 32 bits data

RKI2C_TXDATA1

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1 data1 to be transmitted. 32 bits data

RKI2C_TXDATA2

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2 data2 to be transmitted. 32 bits data

RKI2C_TXDATA3

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3 data3 to be transmitted. 32 bits data

RKI2C_TXDATA4

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4 data4 to be transmitted. 32 bits data

RKI2C_TXDATA5

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5 data5 to be transmitted. 32 bits data

RKI2C_TXDATA6

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6 data6 to be transmitted. 32 bits data

RKI2C_TXDATA7

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7 data7 to be transmitted. 32 bits data

RKI2C_RXDATA0

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata0 data0 received. 32 bits data

RKI2C_RXDATA1

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata1 data1 received. 32 bits data

RKI2C_RXDATA2

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata2 data2 received. 32 bits data

RKI2C_RXDATA3
Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata3 data3 received. 32 bits data

RKI2C_RXDATA4
Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata4 data4 received. 32 bits data

RKI2C_RXDATA5
Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata5 data5 received. 32 bits data

RKI2C_RXDATA6
Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata6 data6 received. 32 bits data

RKI2C_RXDATA7
Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	rxdata7 data7 received. 32 bits data

RKI2C_ST
Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	scl_st scl status: 1'b0: scl status low 1'b0: scl status high

Bit	Attr	Reset Value	Description
0	RO	0x0	sda_st sda status: 1'b0: sda status low 1'b0: sda status high

RKI2C_DBGCTRL

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	h0_check_scl 0: Check if scl been pull down by slave at the whole SCL_HIGH. 1: Check if scl been pull down by slave only at the h0 of SCL_HIGH(SCL_HIGH including h0~h7).
13	RW	0x0	nak_release_scl 0: Hold scl as low when recieved nack 1: Release scl as high when recieved nack
12	RW	0x0	flt_en SCL edage glitch filter enable 0: disable 1: enable
11:8	RW	0x0	slv_hold_scl_th Slave hold scl threshold = slv_hold_scl_th * Tclk_i2c
7:4	RW	0x0	flt_r Filter scl rising edge glitches of width less than flt_r * Tclk_i2c
3:0	RW	0x0	flt_f Filter scl falling edge glitches of width less than flt_f * Tclk_i2c

17.5 Interface Description

Table 17-1I2C Interface Description

Module pin	Direction	Pin name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	GPIO1_D0/UART1_RX/I2C0_SDA/SPI2_CLK	GRF_GPIO1D_IOMUX[1:0]=2’b10
i2c0_scl	I/O	GPIO1_D1/UART1_TX/I2C0_SCL/SPI2_CSN0	GRF_GPIO1D_IOMUX[3:2]=2’b10
I2C1 Interface			
i2c1_sda	I/O	GPIO0_B3/I2C1_SDA	GRF_GPIO0B_IOMUX[7:6]=2’b01
i2c1_scl	I/O	GPIO0_B4/I2C1_SCL	GRF_GPIO0B_IOMUX[9:8]=2’b01
I2C2 Interface			
i2c2_sda	I/O	GPIO2_A2/UART0_CTSN/SPI0_CLK/I2C2_SDA	GRF_GPIO2A_IOMUX[5:4]=2’b11
i2c2_scl	I/O	GPIO2_A3/UART0_RTSN/SPI0_CSN0/I2C2_SCL	GRF_GPIO2A_IOMUX[7:6]=2’b11

I2C3 M0 Interface			
i2c3m0_sda	I/O	GPIO0_B7/PWM2/I2C3_SDA_M0	GRF_GPIO0B_IOMUX[15:14]=2'b10
i2c3m0_scl	I/O	GPIO0_C0/PWM3/I2C3_SCL_M0	GRF_GPIO0C_IOMUX[1:0]=2'b10
I2C3 M1 Interface			
i2c3m1_sda	I/O	GPIO3_B4/FLASH_RDY/I2C3_SDA_M1/SPI1_MOSI/UART3_RX	GRF_GPIO3B_IOMUX[11:8]=4'b0010
i2c3m1_scl	I/O	GPIO3_B5/FLASH_CSN0/I2C3_SCL_M1/SPI1_CSN0/UART3_TX 3scl_GPIO1B5vccio0	GRF_GPIO3B_IOMUX[15:12]=4'b0010

The I/O interface of I2C3 can be chosen by setting GRF_SOC_CON5[4] bit, if this bit is set to 1, I2C3 uses the I2C3m1 I/O interface, if those bit is set to 0, I2C3 uses the I2C3m0 I/O interface.

17.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode. Users are strongly advised to follow

- Transmit only mode (I2C_CON[1:0]=2'b00)

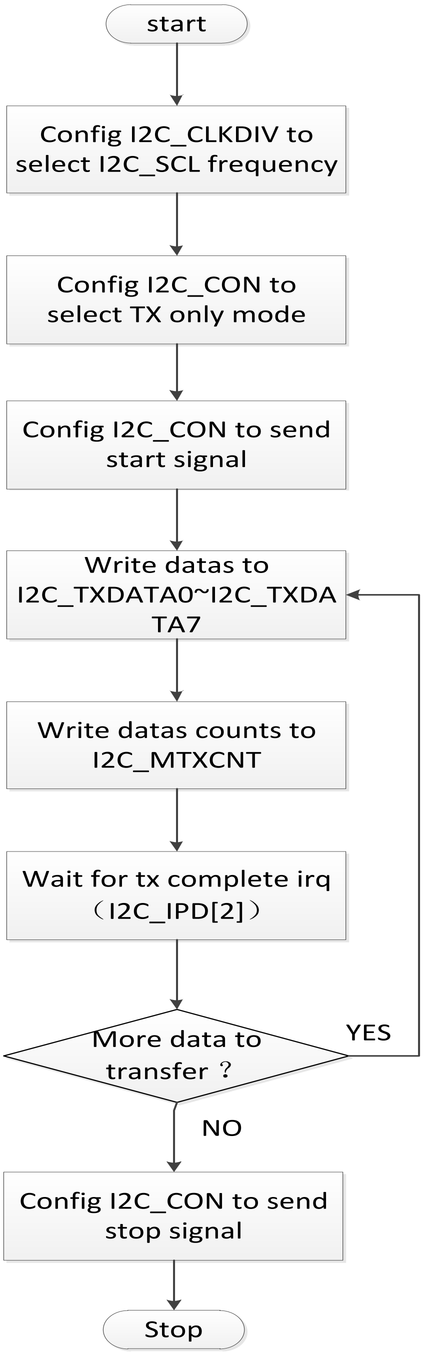

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select TX only mode
Config I2C_CON to send start signal
Write datas to I2C_TXDATA0~I2C_TXDATA7
Write datas counts to I2C_MTXCNT
Wait for tx complete irq（I2C_IPD[2]）
More data to transfer ？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig. 17-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

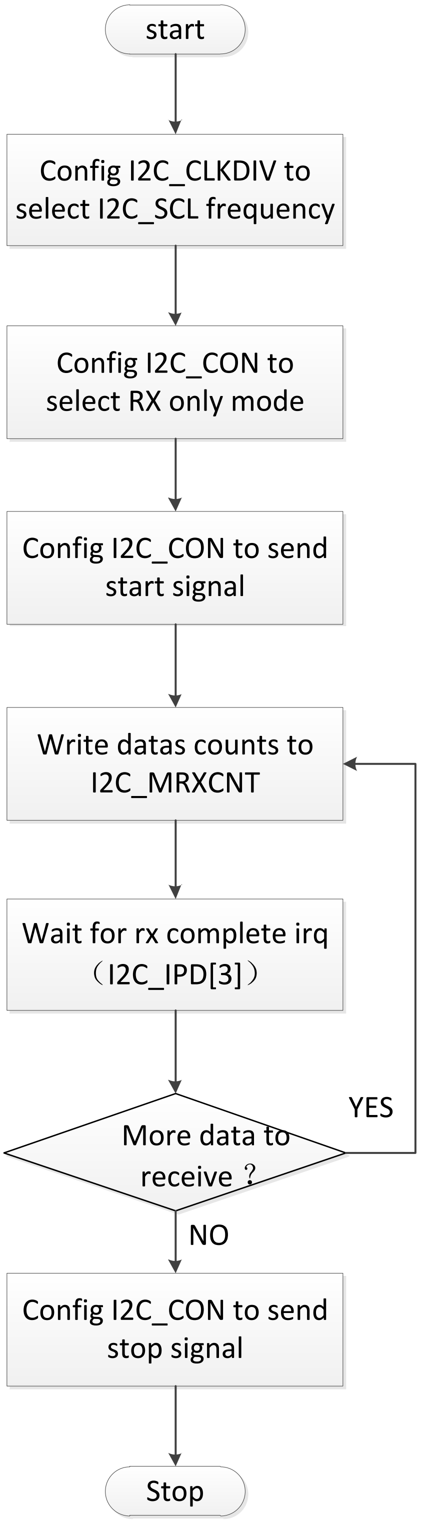

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select RX only mode
Config I2C_CON to send start signal
Write datas counts to I2C_MRXCNT
Wait for rx complete irq （I2C_IPD[3]）
More data to receive？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig. 17-7 I2C Flow chat for receive only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b11)

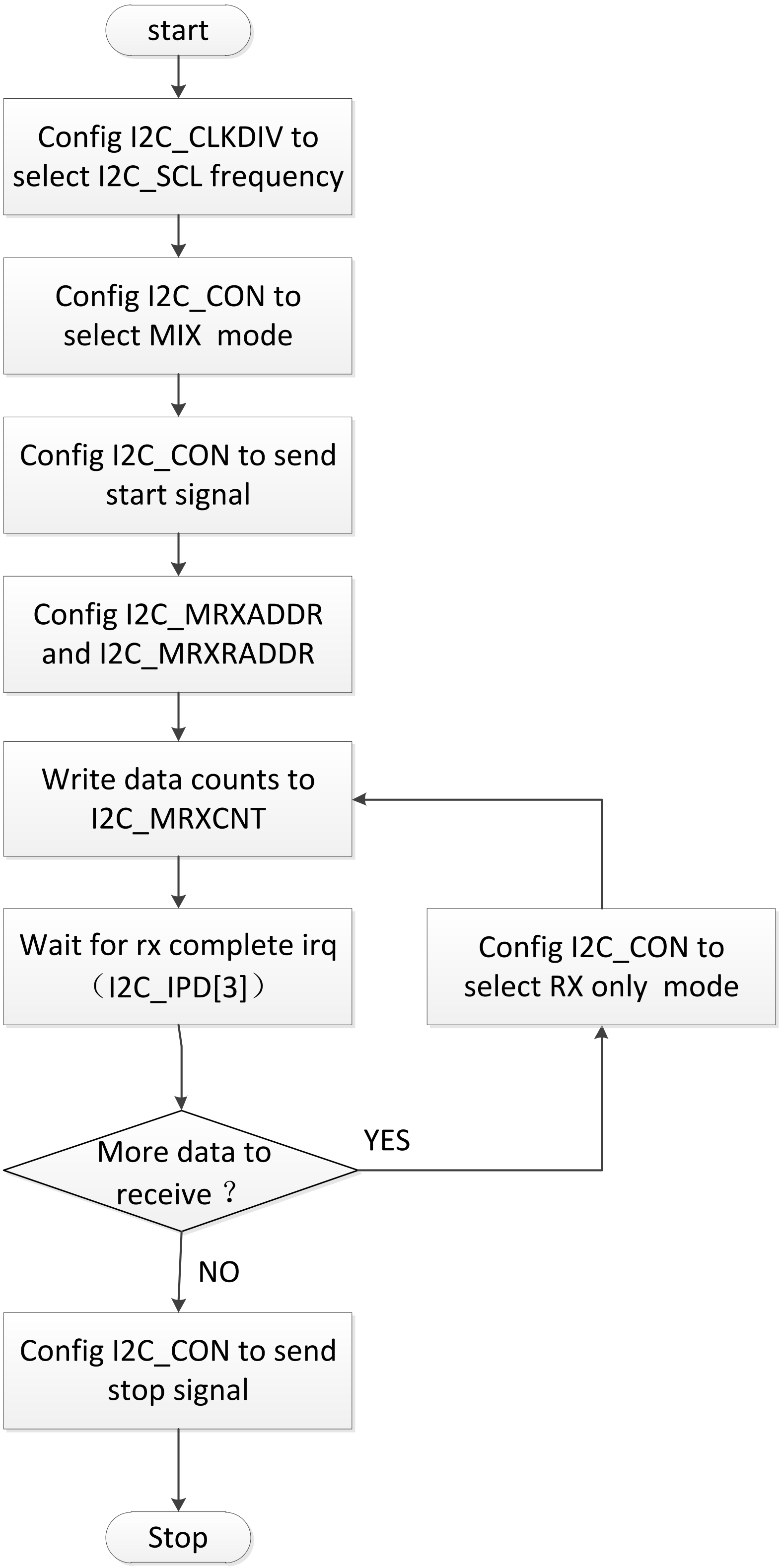

start
Config I2C_CLKDIV to select I2C_SCL frequency
Config I2C_CON to select MIX mode
Config I2C_CON to send start signal
Config I2C_MRXADDR and I2C_MRXRADDR
Write data counts to I2C_MRXCNT
Wait for rx complete irq （I2C_IPD[3]）
Config I2C_CON to select RX only mode
More data to receive？
YES
NO
Config I2C_CON to send stop signal
Stop

Fig. 17-8 I2C Flow chat for mix mode

Chapter 18 I2S 2-channel

18.1 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 is invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

I2S bus is widely used in the devices such as ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

18.1.1 Features

Not only I2S but also PCM mode stereo audio output and input are supported in I2S/PCM controller.

- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving audio data
- Support AHB bus interface
- Support 16~32 bits audio data transfer
- Support master and slave mode
- Support DMA handshaking interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combined interrupt output
- Support 2-channel audio transmitting in I2S mode and PCM mode
- Support 2-channel 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 2 independent LRCK signals, one for receiving and the other for transmitting audio data
- Support configurable SCLK and LRCK polarity

18.2 Block Diagram

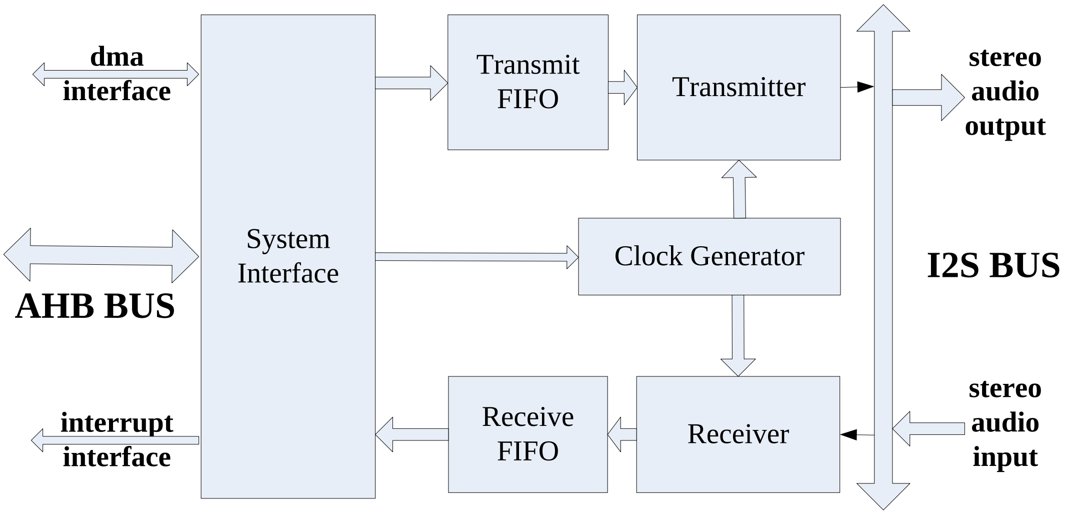


Fig. 18-1 I2S/PCM controller (2 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S or PCM mode stereo serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either a master or a slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.Function Description

18.3 Function description

In the I2S/PCM controller, there are four types: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

In broadcasting application, the I2S/PCM controller is used as a transmitter and external or internal audio CODEC is used as a receiver. In recording application, the I2S/PCM controller is used as a receiver and external or internal audio CODEC is used as a transmitter. Either the I2S/PCM controller or the audio CODEC can act as a master or a slave, but if one is master, the other must be slave.

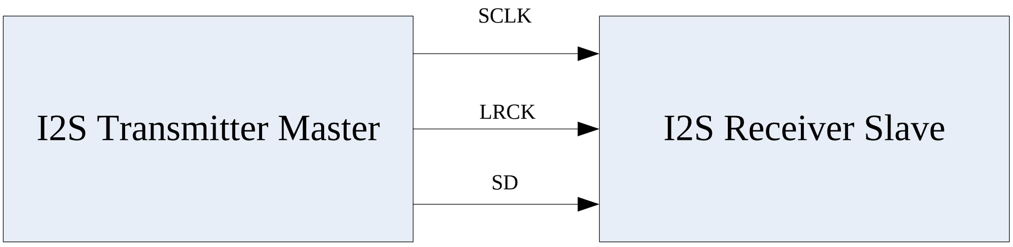


Fig. 18-2 I2S transmitter-master & receiver-slave condition

When the transmitter acts as a master, it sends all signals to the receiver (the slave), and CPU controls when to send clock and data to the receiver. When acts as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from the receiver (the master) to the transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when the transmitter to send data.

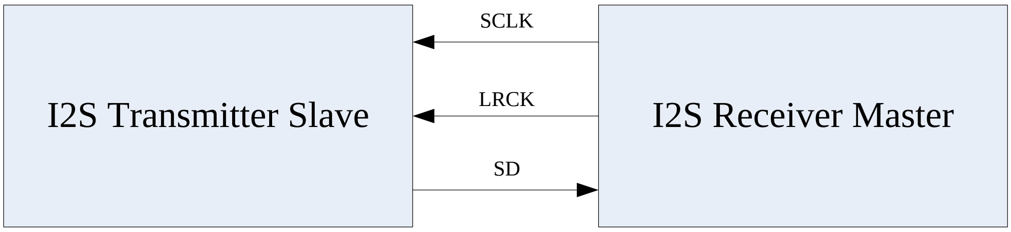


Fig. 18-3 I2S transmitter-slave & receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (the slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

18.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s_sdo, i2s_sdi) signal, it starts sending the first bit (MSB or LSB) one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

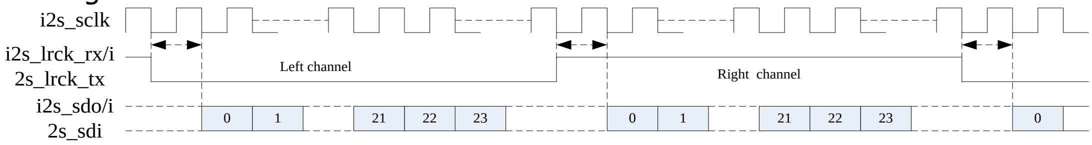


Fig. 18-4 I2S normal mode timing format

18.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s_lrck_rx / i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

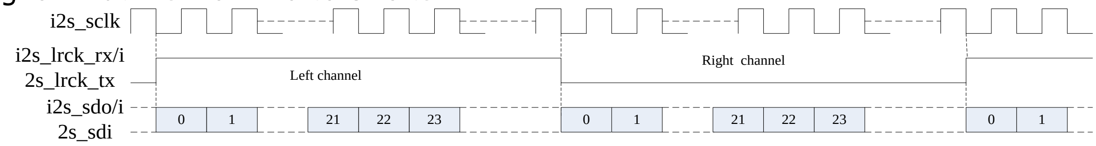


Fig. 18-5 I2S left justified mode timing format

18.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s_lrck_rx/ i2s_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s_sdo, i2s_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode, the last bit of the transferred data is aligned to the transition edge of the LRCK signal while one bit is transferred at one SCLK cycle. The range of SD signal width is from 16 to 32bits.

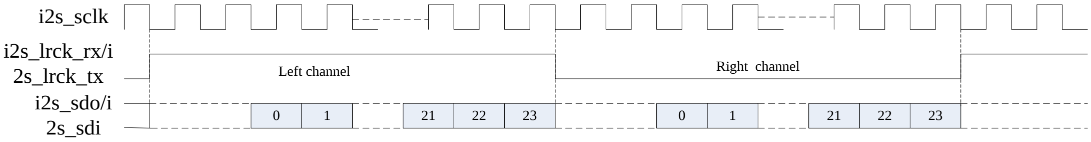


Fig. 18-6 I2S right justified mode timing format

18.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

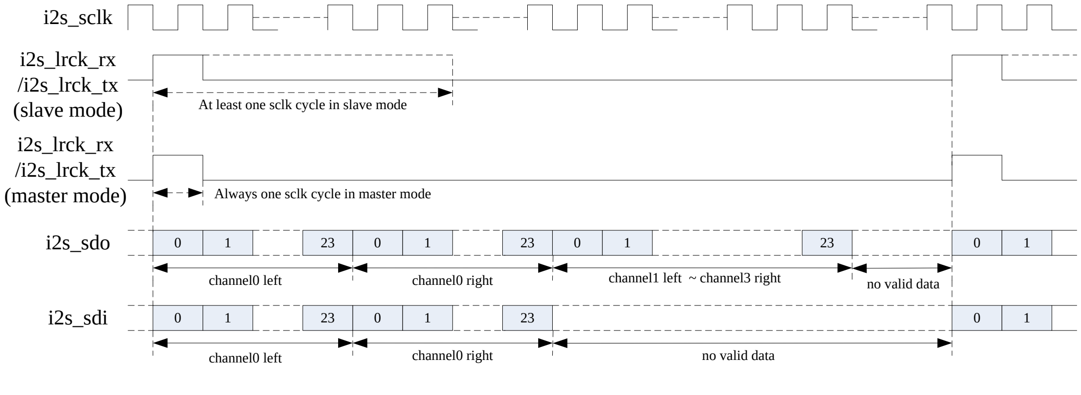


Fig. 18-7 PCM early mode timing format

18.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it sends the first bit (MSB or LSB) one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

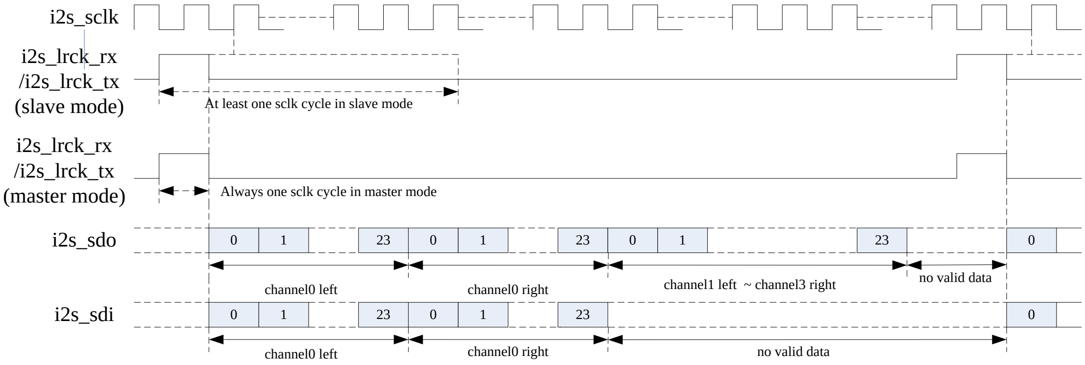

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
0 1 23 0 1 23 0 1 23 0
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s_sdi
0 1 23 0 1 23 0
channel0 left
channel0 right
no valid data

Fig. 18-8 PCM late1 mode timing format

18.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it sends the first bit (MSB or LSB)two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

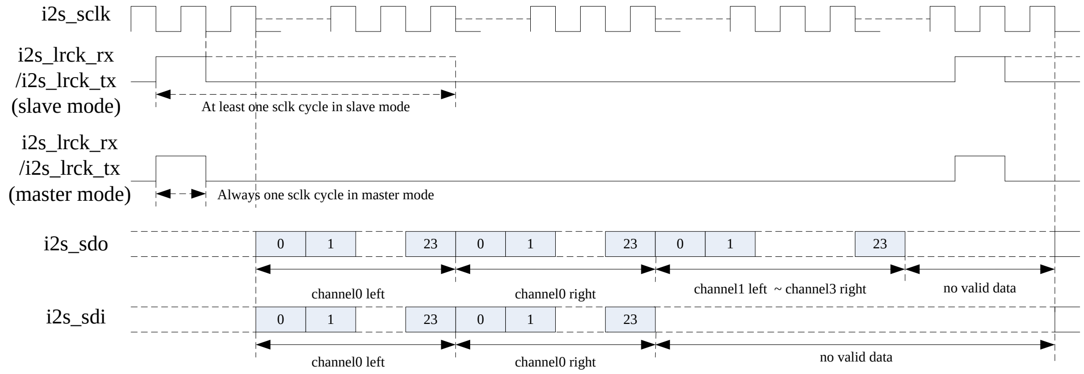

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
0 1 23 0 1 23 0 1 23
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s_sdi
0 1 23 0 1 23
channel0 left
channel0 right
no valid data

Fig. 18-9 PCM late2 mode timing format

18.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s_lrck_rx/i2s_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s_sdo, i2s_sdi) signal, it sends the first bit (MSB or LSB) three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

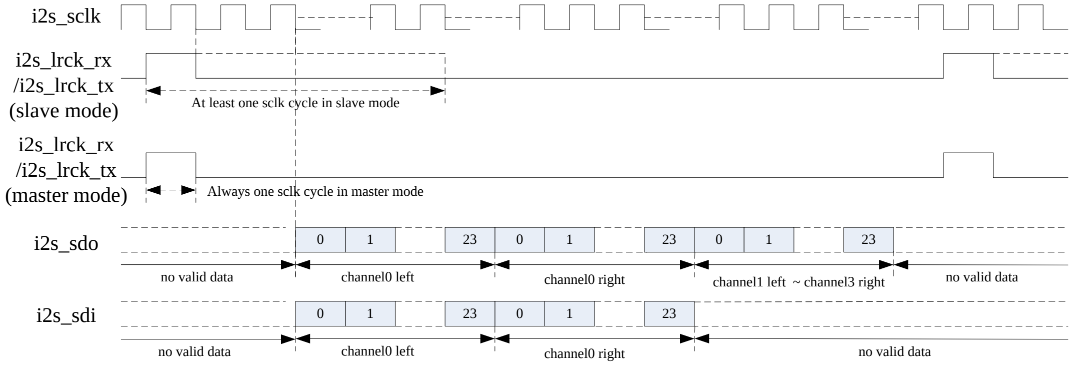

i2s_sclk
i2s_lrck_rx /i2s_lrck_tx (slave mode)
At least one sclk cycle in slave mode
i2s_lrck_rx /i2s_lrck_tx (master mode)
Always one sclk cycle in master mode
i2s_sdo
0 1 23 0 1 23 0 1 23
no valid data
channel0 left
channel0 right
channel1 left ~ channel3 right
no valid data
i2s_sdi
0 1 23 0 1 23
no valid data
channel0 left
channel0 right
no valid data

Fig. 18-10 PCM late3 mode timing format

18.4 Register Description

18.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x00000000	Transmit Operation Control Register
I2S_RXCR	0x0004	W	0x00000000	Receive Operation Control Register
I2S_CKR	0x0008	W	0x00001f00	Clock Generation Register
I2S_FIFOLR	0x000c	W	0x00000000	FIFO Level Register
I2S_DMACR	0x0010	W	0x001f0000	DMA Control Register
I2S_INTCR	0x0014	W	0x00000000	Interrupt Control Register
I2S_INTSR	0x0018	W	0x00000000	Interrupt Status Register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	SCLK Domain Logic Clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register

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

18.4.2 Detail Register Description

I2S_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT Right Jusitified Counter (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave tx mode is selected. Start to transmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel Select Register 2'b00: 2 channel 2'b01~2'b11: reserved
14	RW	0x0	HWT Halfword Word Transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store Justified Mode (Can be written only when XFER[0] bit is 0.) 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. 1'b0: right justified 1'b1: left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	IBM I2S Bus Mode (Can be written only when XFER[0] bit is 0.) 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: reserved

8:7	RW	0x0	PBM PCM Bus Mode (Can be written only when XFER[0] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer Format Select (Can be written only when XFER[0] bit is 0.) 1'b0: I2S format 1'b1: PCM format
4:0	RW	0x00	VDW Valid Data Width (Can be written only when XFER[0] bit is 0.) 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit …… 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	HWT Halfword Word Transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	SJM Store Justified Mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 1'b0: right justified 1'b1: left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	IBM I2S Bus Mode (Can be written only when XFER[1] bit is 0.) 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: reserved
8:7	RW	0x0	PBM PCM Bus Mode (Can be written only when XFER[1] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer Format Select (Can be written only when XFER[1] bit is 0.) 1'b0: i2s 1'b1: pcm

Bit	Attr	Reset Value	Description
4:0	RW	0x00	VDW Valid Data Width (Can be written only when XFER[1] bit is 0.) 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit …… 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM TX/RX LRCK as common 2'b00/2'b11: tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01: only tx_lrck is used as synchronous signal for TX and RX. 2'b10: only rx_lrck is used as synchronous signal for TX and RX. Note: When set to 2'b01 or 2'b10 , if user wants to use both transmitting and receiving in master mode, user should configure as following: a. The value of TSD and RSD should be same. b. User should start TX transfer and RX transfer at the same time.
27	RW	0x0	MSS Master/Slave Mode Select (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: master mode(sclk output) 1'b1: slave mode(sclk input)
26	RW	0x0	CKP Sclk Polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: sample data at posedge sclk and drive data at negedge sclk 1'b1: sample data at negedge sclk and drive data at posedge sclk

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.) 1'b0: normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1'b1: opposite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
24	RW	0x0	TLP Transmit Lrck Polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: normal polarity (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal: high valid) 1'b1: opposite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal: low valid)
23:16	RW	0x00	MDIV Mclk Divider (Can be written only when XFER[1] or XFER[0] bit is 0.) mclk divider = (mclk/sclk)-1. For example, if mclk divider is 5, then the frequency of sclk is mclk/6
15:8	RW	0x1f	RSD Receive Sclk Divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: reserved 8'h1f~8'hff: frequency of sclk = (RSD>>1 +1)*2*frequency of rx_lrck
7:0	RW	0x00	TSD Transmit Sclk Divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: reserved 8'h1f~8'hff: frequency of sclk = (TSD>>1 + 1)*2*frequency of tx_lrck

I2S_FIFOLR

Address: Operational Base + offset (0x000c)

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.
23:6	RO	0x0	reserved
5:0	RW	0x00	TFL Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO.

I2S_DMACR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO is equal to or below this field value.

I2S_INTCR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX Overrun Interrupt Clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX Overrun Interrupt Enable 1'b0: disable 1'b1: enable
16	RW	0x0	RXFIE RX Full Interrupt Enable 1'b0: disable 1'b1: enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX Underrun Interrupt Clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX Underrun Interrupt Enable 1'b0: disable 1'b1: enable
0	RW	0x0	TXEIE TX empty Interrupt Enable 1'b0: disable 1'b1: enable

I2S_INTSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX Overrun Interrupt 1'b0: inactive 1'b1: active

Bit	Attr	Reset Value	Description
16	RO	0x0	RXFI RX Full Interrupt 1'b0: inactive 1'b1: active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX Underrun Interrupt 1'b0: inactive 1'b1: active
0	RO	0x0	TXEI TX Empty Interrupt 1'b0: inactive 1'b1: active

I2S_XFER

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer Start Bit 1'b0: stop RX transfer 1'b1: start RX transfer
0	RW	0x0	TXS TX Transfer Start Bit 1'b0: stop TX transfer 1'b1: start TX transfer

I2S_CLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX Logic Clear This is a self-cleared bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX Logic Clear This is a self-cleared bit. Write 1 to clear all transmit logic.

I2S_TXDR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TXDR Transmit FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S_RXDR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

18.5 Interface Description

There are two 2-channel I2S in total. The I2S02CH is for transmitting and receiving audio data while the I2S12CH is only for receiving audio data. Signal i2s_lrck_tx will be shared if the I2S02CH works at transmitting and receiving modes at the same time.

Table 18-1 I2S02CH Interface Description

Module Pin	Direction	PIN Name	IOMUX Setting
i2s02ch_clk	O	I2S0_2CH_MCLK	GRF_GPIO4B_IOMUX[9:8]=2'b01
i2s02ch_sclk	I/O	I2S0_2CH_SCLK	GRF_GPIO4B_IOMUX[11:10]=2'b01
i2s02ch_lrck_tx	I/O	I2S0_2CH_LRCK_TX	GRF_GPIO4B_IOMUX[13:12]=2'b01
i2s02ch_sdo	O	I2S0_2CH_SDO	GRF_GPIO4B_IOMUX[15:14]=2'b01
i2s02ch_sdi	I	I2S0_2CH_SDI	GRF_GPIO4C_IOMUX[1:0]=2'b01

The I2S12CH is connected to the audio codec which is used to receive audio data. It works at slave mode and receives audio data from audio codec by configuring GRF_SOC_CON1.

Table 18-2 I2S12CH Interface Description

Module Pin	Direction	Module Pin	IOMUX Setting
pin_adc_sck_o	O	i2s12ch_sclk	
pin_adc_ws_o	O	i2s12ch_lrck_rx	
pin_adc_sd_o_0	O	i2s12ch_sdi	GRF_SOC_CON1[13:12]=2'b00
pin_adc_sd_o_1	O		GRF_SOC_CON1[13:12]=2'b01
pin_adc_sd_o_2	O		GRF_SOC_CON1[13:12]=2'b10
pin_adc_sd_o_3	O		GRF_SOC_CON1[13:12]=2'b11

18.6 Application Notes

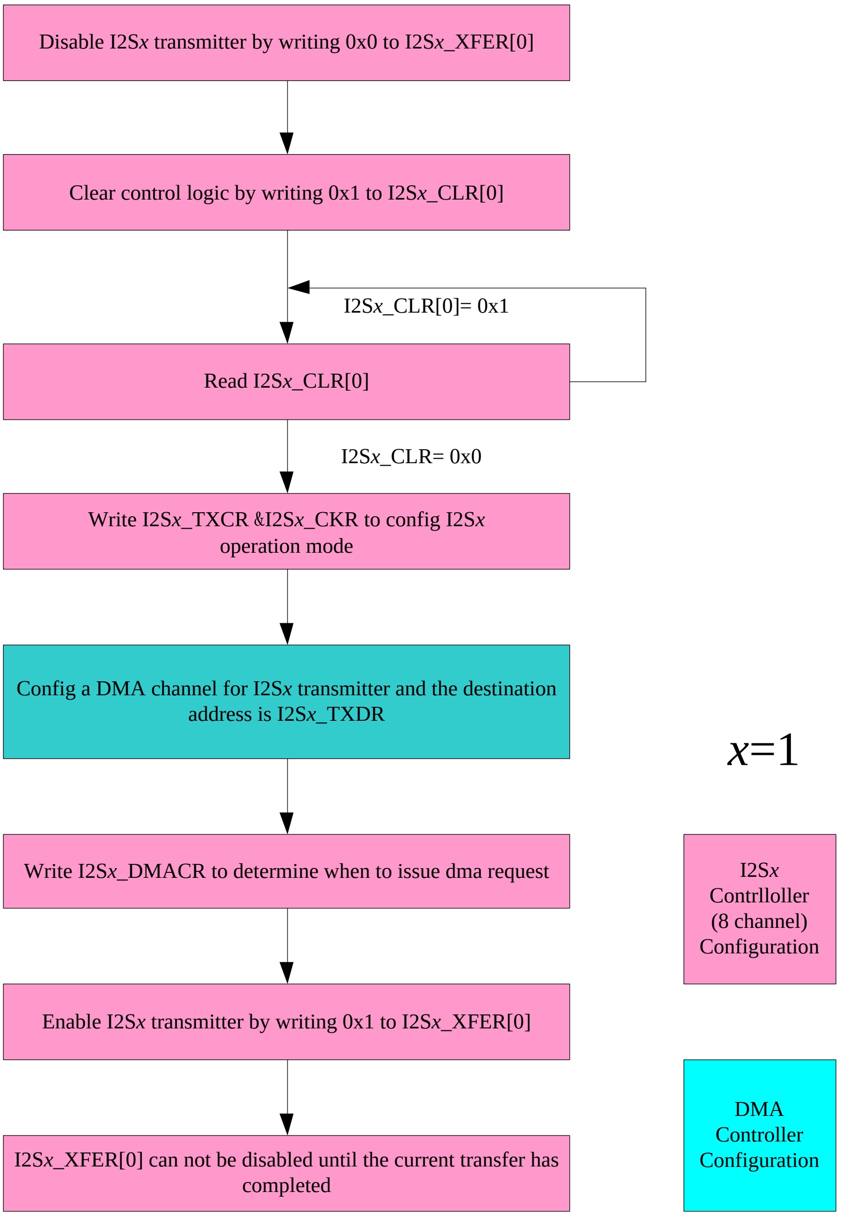

Disable I2Sx transmitter by writing 0x0 to I2Sx_XFER[0]
Clear control logic by writing 0x1 to I2Sx_CLR[0]
I2Sx_CLR[0]= 0x1
Read I2Sx_CLR[0]
I2Sx_CLR= 0x0
Write I2Sx_TXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx transmitter and the destination address is I2Sx_TXDR
x=1
Write I2Sx_DMACR to determine when to issue dma request
I2Sx Contrlloller (8 channel) Configuration
Enable I2Sx transmitter by writing 0x1 to I2Sx_XFER[0]
DMA Controller Configuration
I2Sx_XFER[0] can not be disabled until the current transfer has completed

Fig. 18-11 I2S/PCM controller transmit operation flow chart

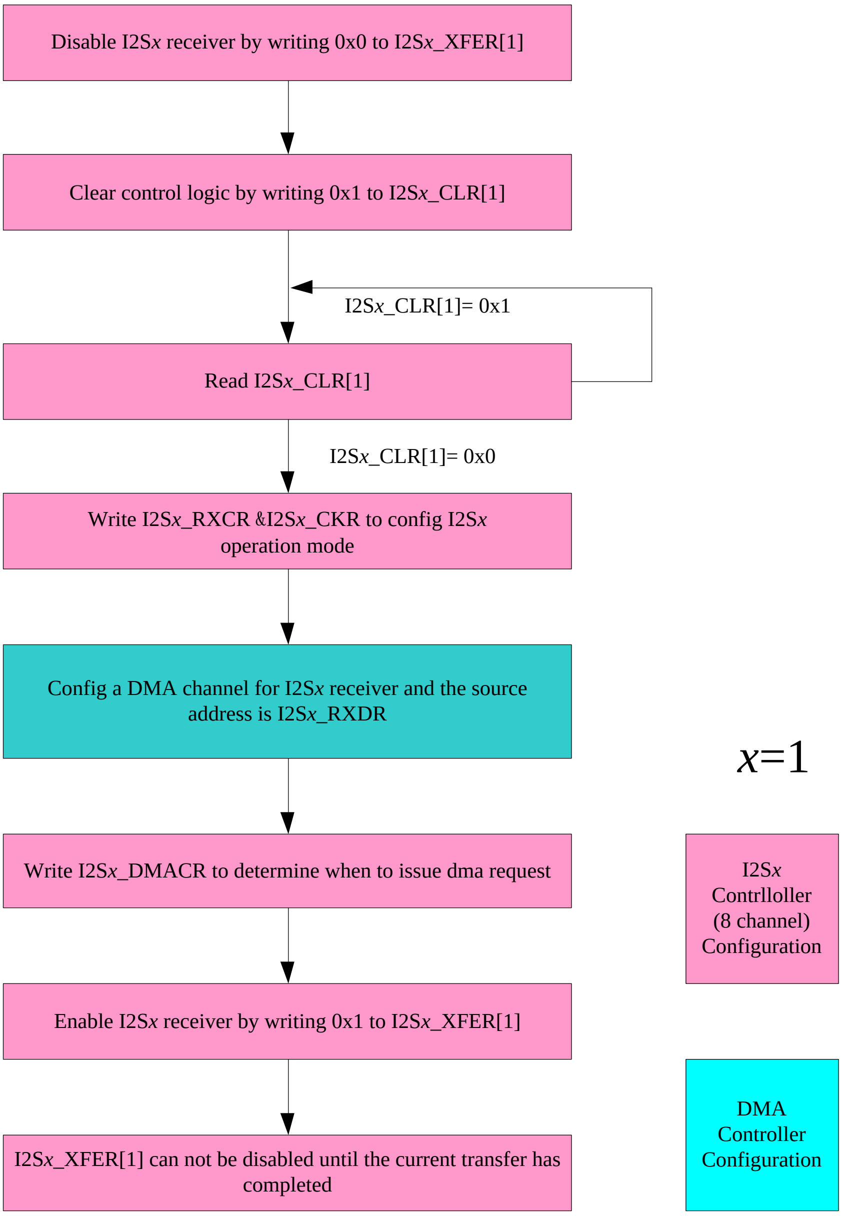

Disable I2Sx receiver by writing 0x0 to I2Sx_XFER[1]
Clear control logic by writing 0x1 to I2Sx_CLR[1]
I2Sx_CLR[1]= 0x1
Read I2Sx_CLR[1]
I2Sx_CLR[1]= 0x0
Write I2Sx_RXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx receiver and the source address is I2Sx_RXDR
Write I2Sx_DMACR to determine when to issue dma request
Enable I2Sx receiver by writing 0x1 to I2Sx_XFER[1]
I2Sx_XFER[1] can not be disabled until the current transfer has completed
x=1
I2Sx Contrllolle r (8 channel) Configuration
DMA Controller Configuration

Fig. 18-12 I2S/PCM controller receive operation flow chart

There are 3 operation modes which are normal mode, TX common mode and RX common mode. Following shows the normal mode connections, I2S_CKR[29:28] should be set to 2’b00.

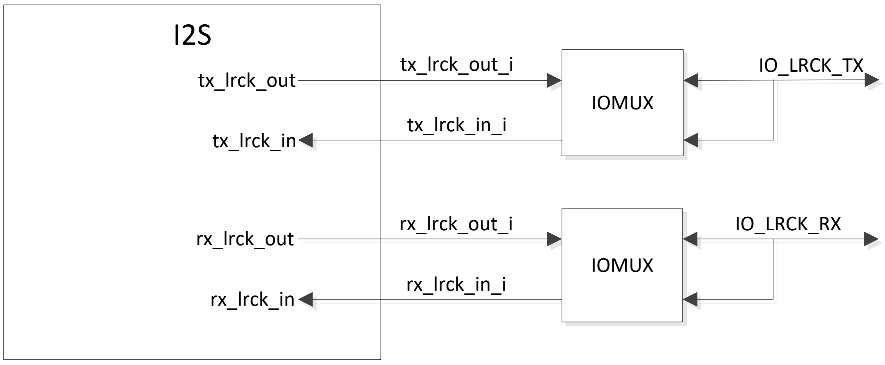


Fig. 18-13 I2S/PCM normal mode

For TX common mode, only IO_LRCK_TX is used and I2S_CKR[29:28] should be set to 2’b01. TX/RX should work at the same time and at the same sample rate.

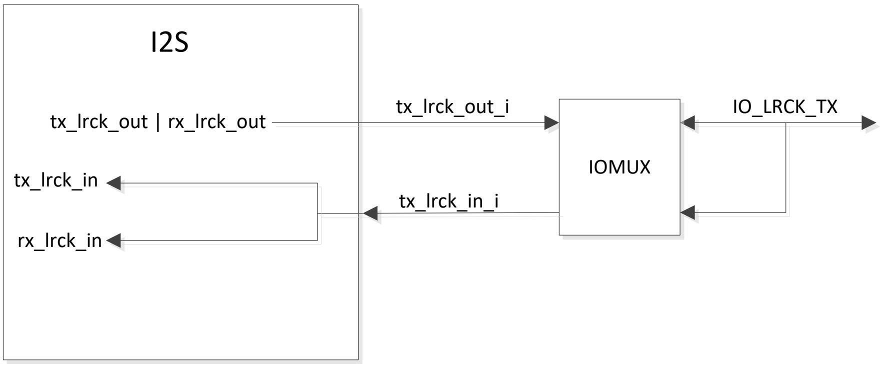


Fig. 18-14 I2S/PCM TX common mode

For RX common mode, only IP_LRCK_RX is used and I2S_CKR[29:28] should be set to 2'b10. TX/RX should work at the same time and at the same sample rate.

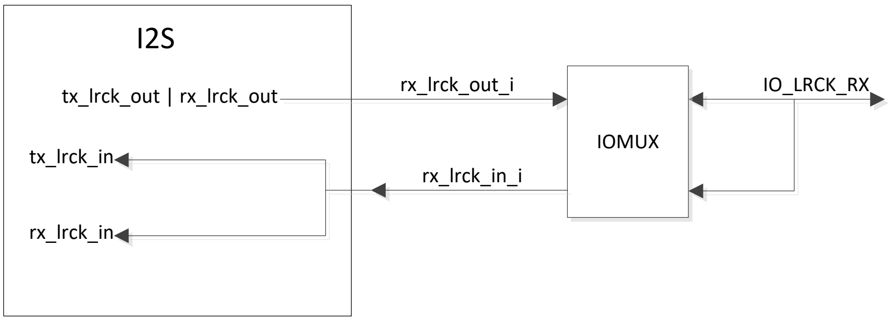


Fig. 18-15 I2S/PCM RX common mode

Chapter 19 I2S 8-channel

19.1 Overview

The I2S/PCM/TDM controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and is invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

I2S bus is widely used in the devices such as ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

19.1.1 Features

The I2S/PCM/TDM controller supports I2S, PCM and TDM mode stereo audio output and input.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and the other for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshaking interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combined interrupt output
- Support 8-channel audio transmitting in I2S/TDM mode and 2-channel in PCM mode.
- Support 8-channel audio receiving in I2S/TDM mode and 2 channel in PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support TDM normal,1/2 cycle left shift ,1 cycle left shift,2 cycle left shift, right shift mode serial audio data transfer.
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bits audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and the other for transmitting audio data. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity
- Support TDM programmable slot bit width: 16~32bits
- Support TDM programmable frame width: 32~512bits
- Support TDM programmable FSYNC width
- Support SDI, SDO IOMUX.

19.2 Block Diagram

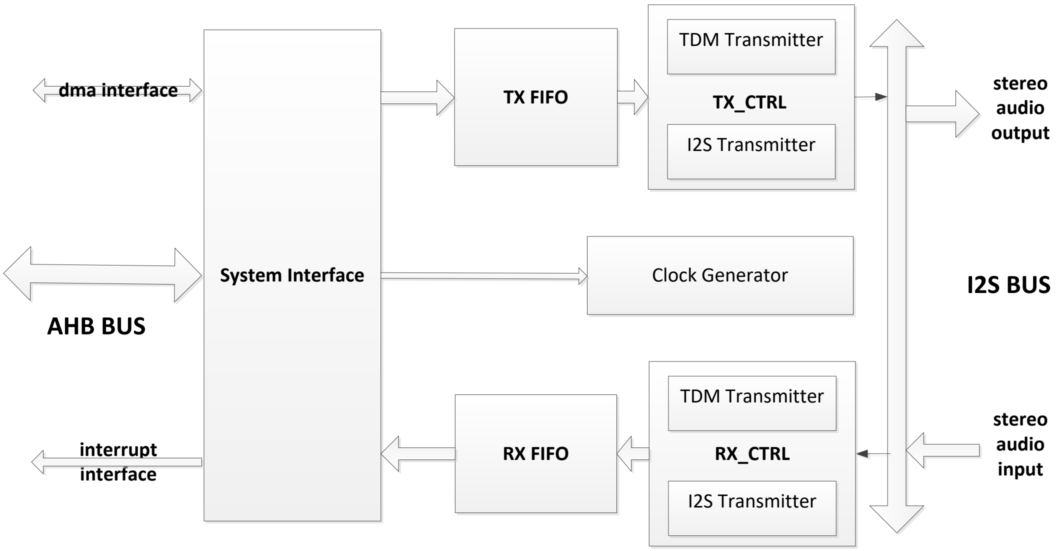


Fig. 19-1 I2S/PCM/TDM controller (8 channel) Block Diagram

System Interface
The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Generator
The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters
The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S, PCM or TDM mode surround serial audio interface.

Receiver
The Receiver implements receive operation. The receiver can act as either a master or a slave, with I2S, PCM or TDM mode stereo serial audio interface.

Transmit FIFO
The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO
The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

19.3 Function description

In the I2S/PCM/TDM controller, there are four types: transmitter-master &receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.
In broadcasting application, the I2S/PCM/TDM controller is used as a transmitter and external or internal audio CODEC is used as a receiver. In recording application, the I2S/PCM/TDM controller is used as a receiver and external or internal audio CODEC is used as a transmitter. Either the I2S/PCM/TDM controller or the audio CODEC can act as a master or a slave, but if one is master, the other must be slave.

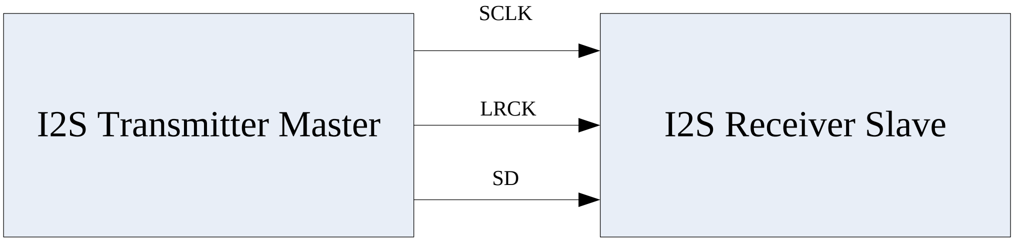


Fig. 19-2 I2S transmitter-master & receiver-slave condition

When the transmitter acts as a master, it sends all signals to the receiver (the slave), and CPU controls when to send clock and data to the receiver. When acts as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from the receiver (the master) to the transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when the transmitter to send data.

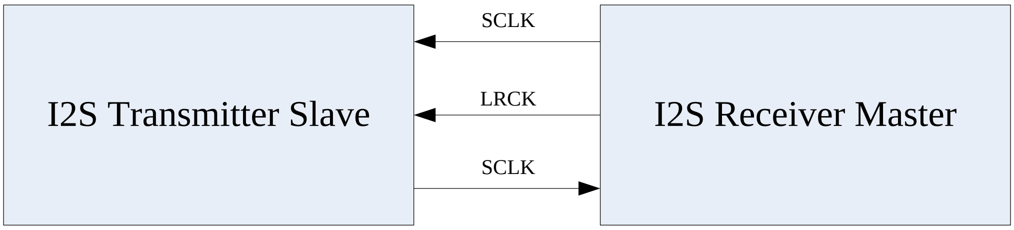


Fig. 19-3 I2S transmitter-slave & receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (the slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

19.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

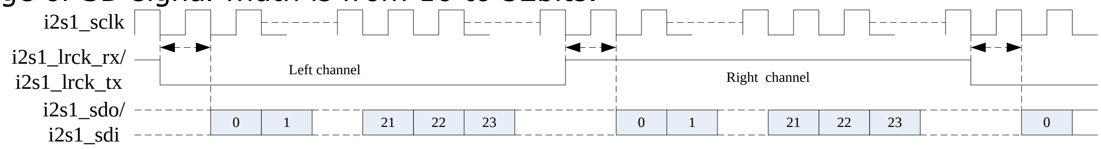


Fig. 19-4 I2S normal mode timing format

19.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s1_lrck_rx / i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

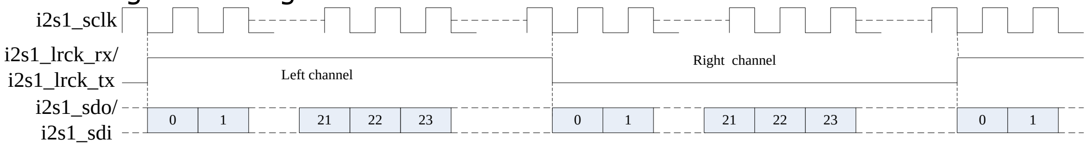


Fig. 19-5 I2S left justified mode timing format

19.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s1_lrck_rx/ i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode, the last bit of the transferred data is aligned to the transition edge of the LRCK signal while one bit is transferred at one SCLK cycle. The range of SD signal width is from 16 to 32bits.

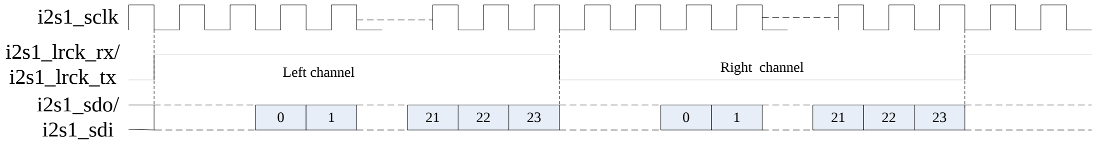


Fig. 19-6 I2S right justified mode timing format

19.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

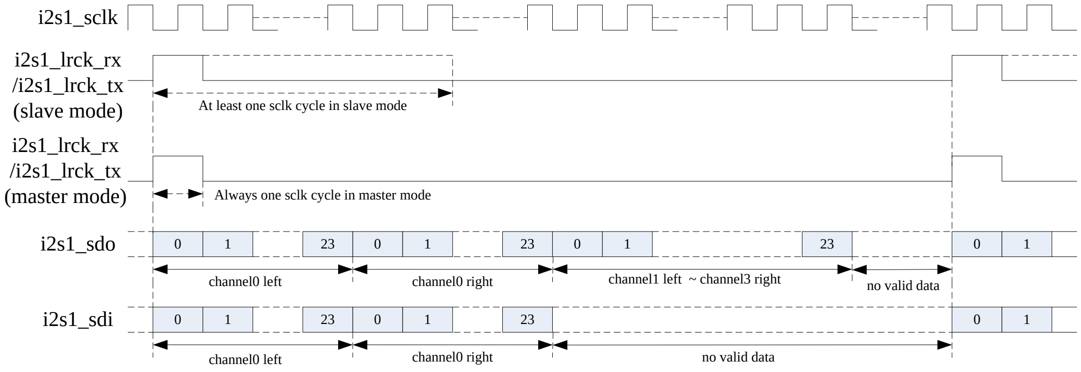


Fig. 19-7 PCM early mode timing format

19.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

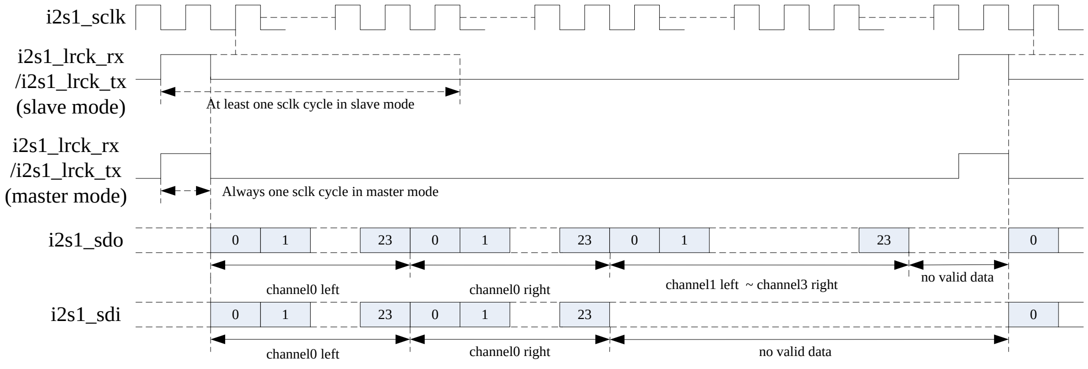


Fig. 19-8 PCM late1 mode timing format

19.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB)two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

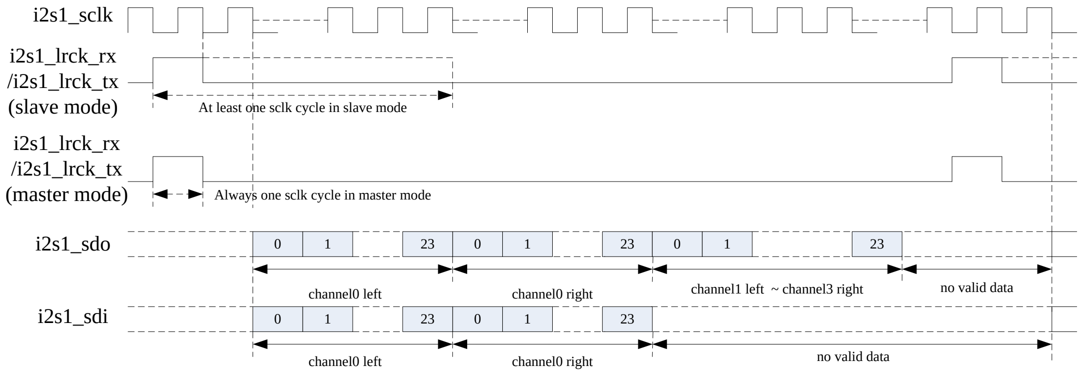


Fig. 19-9 PCM late2 mode timing format

19.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

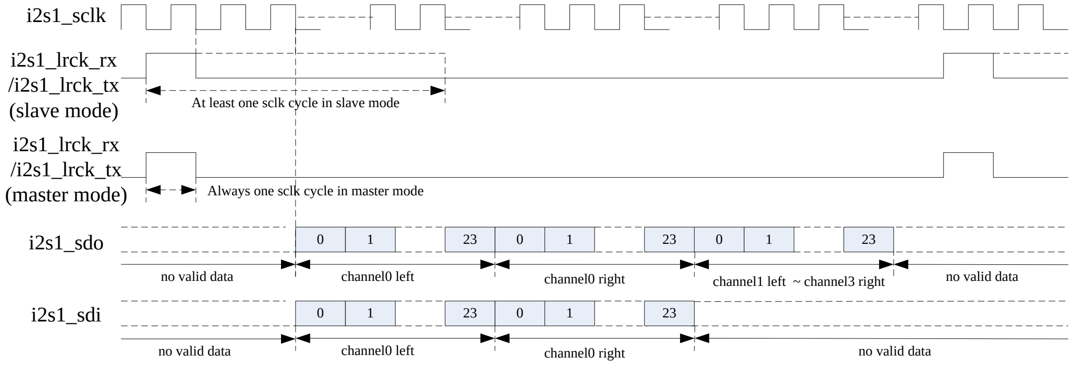
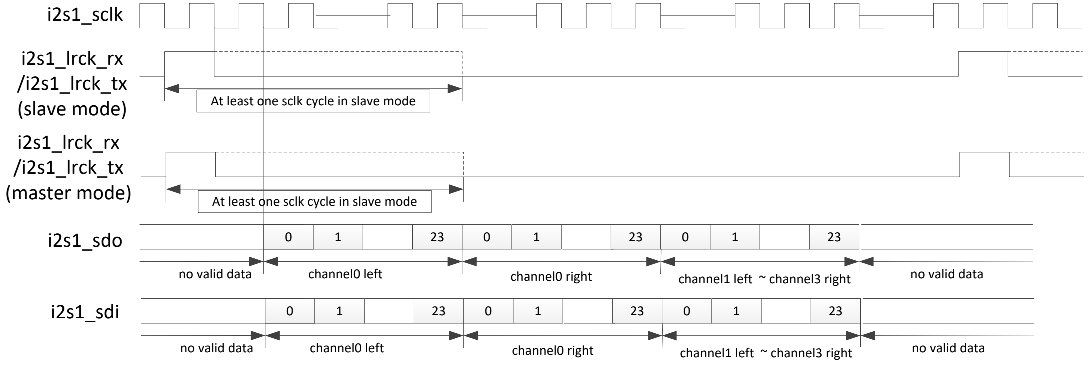


Fig. 19-10 PCM late3 mode timing format

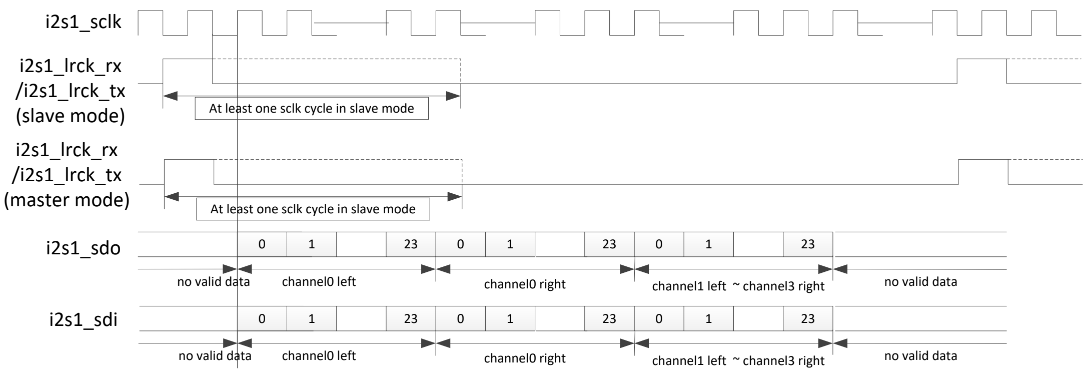
19.3.8 TDM normal mode (PCM format)

This is the waveform of TDM normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the second falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.



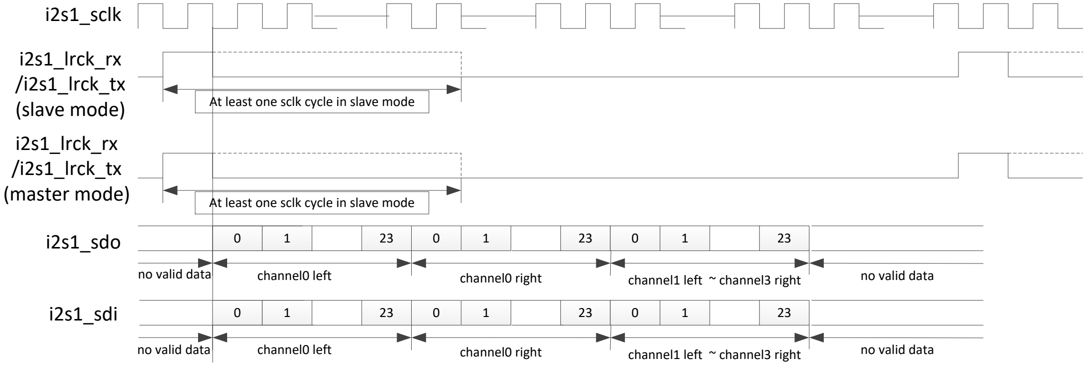
19.3.9 TDM left shift mode0 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the second rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.



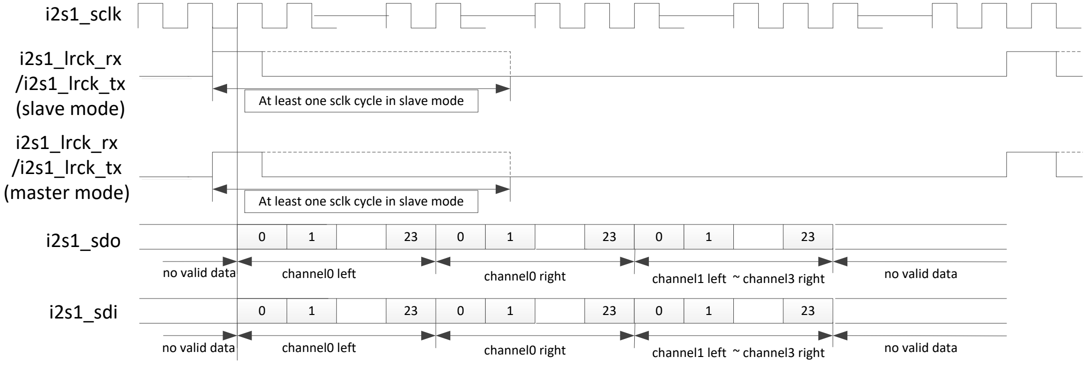
19.3.10 TDM left shift mode1 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the first falling edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.



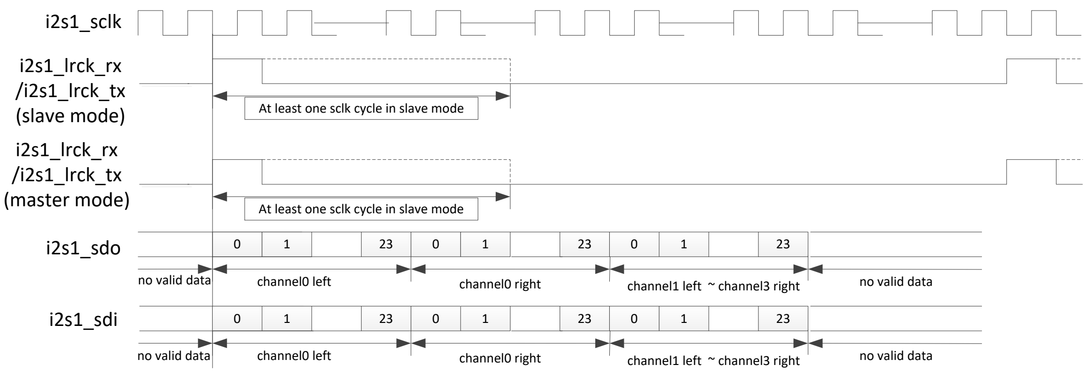
19.3.11 TDM left shift mode2 (PCM format)

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) on the first rising edge of SCLK after LRCK goes high. The range of SD signal width is from 16 to 32bits.



19.3.12 TDM left shift mode3 (PCM format)

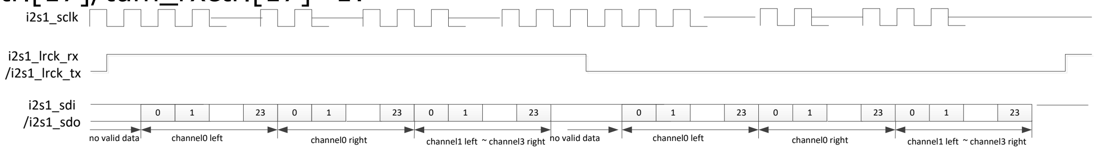
This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it sends the first bit (MSB or LSB) at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.



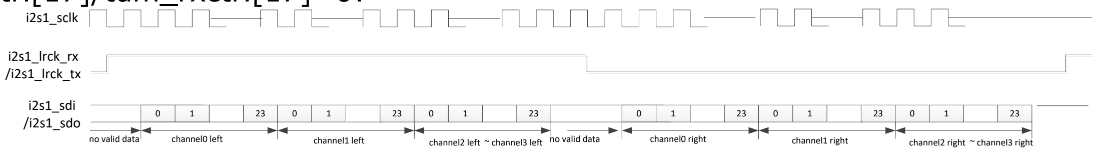
19.3.13 TDM normal mode (I2S format)

This is the waveform of I2S normal mode. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) on the first falling edge of SCLK after LRCK changes. The range of SD signal width is from 16 to 32bits.

tdm_txctrl[17]/tdm_rxctrl[17]=1:

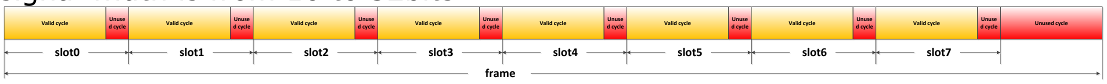


tdm_txctrl[17]/tdm_rxctrl[17]=0:

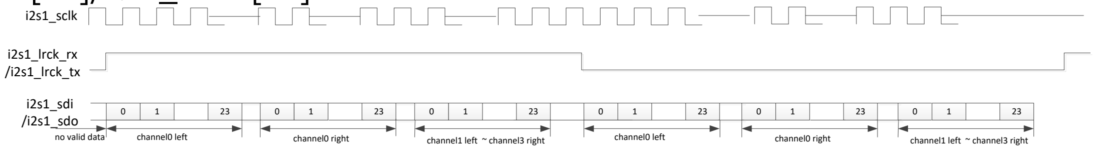


19.3.14 TDM left justified mode (I2S format)

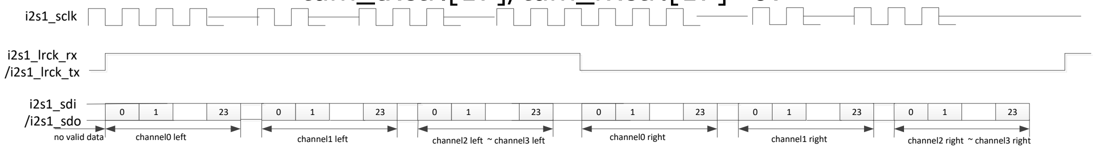
This is the waveform of I2S left justified mode. For SD (i2s1_sdo, i2s1_sdi) signal, it starts sending the first bit (MSB or LSB) at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.



tdm_txctrl[17]/tdm_rxctrl[17]=1:

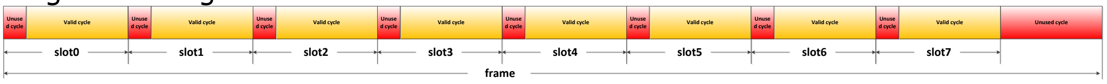


tdm_txctrl[17]/tdm_rxctrl[17]=0:

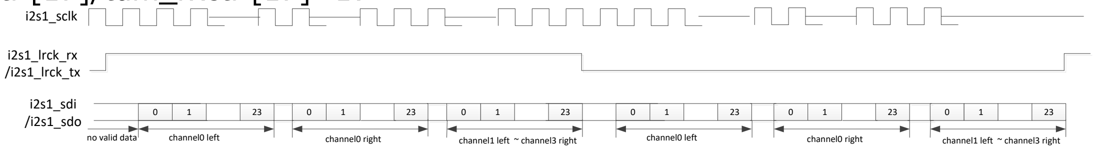


19.3.15 TDM right justified mode (I2S format)

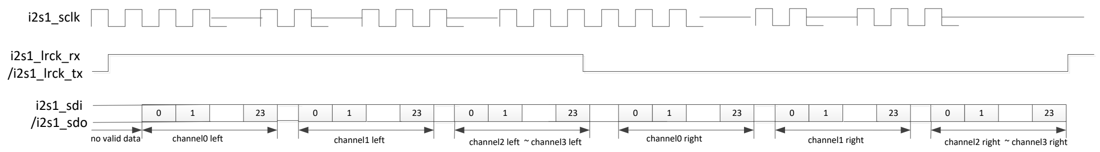
This is the waveform of I2S right justified mode. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but what is different from I2S normal or left justified mode.The range of SD signal width is from 16 to 32bits.



tdm_txctrl[17]/tdm_rxctrl[17]=1:



tdm_txctrl[17]/tdm_rxctrl[17]=0:



19.4 Register description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_8CH_TXCR	0x0000	W	0x7200000f	Transmit operation control register.
I2S_8CH_RXCR	0x0004	W	0x01c8000f	Receive operation control register
I2S_8CH_CKR	0x0008	W	0x00001f1f	Clock generation register
I2S_8CH_TXFIFOLR	0x000c	W	0x00000000	TX FIFO level register
I2S_8CH_DMACR	0x0010	W	0x001f0000	DMA control register
I2S_8CH_INTCR	0x0014	W	0x01f00000	Interrupt control register
I2S_8CH_INTSR	0x0018	W	0x00000000	Interrupt status register
I2S_8CH_XFER	0x001c	W	0x00000000	Transfer start register
I2S_8CH_CLR	0x0020	W	0x00000000	SCLK domain logic clear register
I2S_8CH_TXDR	0x0024	W	0x00000000	Transmit FIFO data register
I2S_8CH_RXDR	0x0028	W	0x00000000	Receive FIFO data register
I2S_8CH_RXFIFOLR	0x002c	W	0x00000000	RX FIFO level register
I2S_8CH_TDM_TXCTRL	0x0030	W	0x00003eff	TDM mode transmit operation control register
I2S_8CH_TDM_RXCTRL	0x0034	W	0x00003eff	TDM mode receive operation control register
I2S_8CH_CLKDIV	0x0038	W	0x00000707	Clock divider register
I2S_8CH_VERSION	0x003c	W	0x20150001	I2S version register
I2S_8CH_INCR_RXDR	0x0800	W	0x00000000	Increment address 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_8CH_TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:29	RW	0x3	tx_path_select3 Tx path select: 2'b00: sdo3 output data from path0; 2'b01: sdo3 output data from path1; 2'b10: sdo3 output data from path2; 2'b11: sdo3 output data from path3; Note: when TDM mode, only path0 enable.

Bit	Attr	Reset Value	Description
28:27	RW	0x2	tx_path_select2 Tx path select: 2'b00: sdo2 output data from path0; 2'b01: sdo2 output data from path1; 2'b10: sdo2 output data from path2; 2'b11: sdo2 output data from path3; Note: when TDM mode, only path0 enable.
26:25	RW	0x1	tx_path_select1 Tx path select: 2'b00: sdo1 output data from path0; 2'b01: sdo1 output data from path1; 2'b10: sdo1 output data from path2; 2'b11: sdo1 output data from path3; Note: when TDM mode, only path0 enable.
24:23	RW	0x0	tx_path_select0 Tx path select: 2'b00: sdo0 output data from path0; 2'b01: sdo0 output data from path1; 2'b10: sdo0 output data from path2; 2'b11: sdo0 output data from path3; Note: when TDM mode, only path0 enable.
22:17	RW	0x00	RCNT 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 transmit data RCNT sclk cycles after left channel valid. Note: Only function when TX TFS[1]=0;
16:15	RW	0x0	TCSR TX Channel select register 2'b00: two channel 2'b01: four channel 2'b10: six channel 2'b11: eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
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. 1'b0: right justified 1'b1: left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: reserved Note: Only function when TX TFS[1:0] is 0;
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode Note: function when TX TFS[1:0] is 1;
6:5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 2'b00: I2S format 2'b01: PCM format 2'b10: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data Width (Can be written only when XFER[0] bit is 0.) 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit …… 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_8CH_RXCR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

24:23	RW	0x3	rx_path_select3
			Rx path select:
			2'b00: path3 data from sdi0;
			2'b01: path3 data from sdi1;
			2'b10: path3 data from sdi2;
			2'b11: path3 data from sdi3;
			Note: inoperative at TDM mode.

22:21	RW	0x2	rx_path_select2
			Rx path select:
			2'b00: path2 data from sdi0;
			2'b01: path2 data from sdi1;
			2'b10: path2 data from sdi2;
			2'b11: path2 data from sdi3;
			Note: inoperative at TDM mode.

20:19	RW	0x1	rx_path_select1
			Rx path select:
			2'b00: path1 data from sdi0;
			2'b01: path1 data from sdi1;
			2'b10: path1 data from sdi2;
			2'b11: path1 data from sdi3;
			Note: inoperative at TDM mode.

18:17	RW	0x0	rx_path_select0
			Rx path select:
			2'b00: path0 data from sdi0;
			2'b01: path0 data from sdi1;
			2'b10: path0 data from sdi2;o
			2'b11: path0 data from sdi3;

Bit	Attr	Reset Value	Description
16:15	RW	0x0	RCSR RX Channel select register 2'b00: two channel 2'b01: four channel 2'b10: six channel 2'b11: eight channel
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: low 16bit data valid to AHB/APB bus, high 16 bit data invalid.id.
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. 1'b0: right justified 1'b1: left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 1'b0: MSB 1'b1: LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 2'b00: I2S normal 2'b01: I2S Left justified 2'b10: I2S Right justified 2'b11: reserved Note: Only function when RX TFS[1:0] is 0;
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode Note: Only function when RX TFS[1:0] is 1;

Bit	Attr	Reset Value	Description
6:5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 2'b00: I2S format 2'b01: PCM format 2'b10: TDM format 0 (PCM mode) 2'b11: TDM format 1 (I2S mode)
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 5'b00000~5'b01110: reserved 5'b01111: 16bit 5'b10000: 17bit 5'b10001: 18bit 5'b10010: 19bit …… 5'b11100: 29bit 5'b11101: 30bit 5'b11110: 31bit 5'b11111: 32bit

I2S_8CH_CKR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	TRCM TX/RX LRCK as common 2'b00/2'b11: tx_lrck/rx_lrck are used as synchronous signal for TX /RX respectively. 2'b01: only tx_lrck is used as synchronous signal for TX and RX. 2'b10: only rx_lrck is used as synchronous signal for TX and RX. Note: When set to 2'b01 or 2'b10 , if user wants to use both transmitting and receiving in master mode, user should configure as following: a. mclk_rx and mclk_tx should source from same clock. b. The value of TX_MDIV/RX MDIV should be same, The value of TSD and RSD should be same. c. Before start TX/RX transfer, user should assert mresetn_tx and mresetn_rx, then clear them at the same time after a delay. d. user should start TX transfer and RX transfer at the same time.
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: master mode(sclk output) 1'b1: slave mode(sclk input)

Bit	Attr	Reset Value	Description
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: sample data at posedge sclk and drive data at negedge sclk 1'b1: sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: normal 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'b1: 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.) 1'b0: 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'b1: oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)
23:16	RO	0x0	reserved
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: reserved 8'h1f~8'hff: frequency of sclk = (RSD>>1 +1)*2*frequency of rx_lrck Note: function when RX TFS[1:0] is 0 or 1;
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: reserved 8'h1f~8'hff: frequency of sclk = (TSD>>1 + 1)*2*frequency of tx_lrck Note: function when TX TFS[1:0] is 0 or 1;

I2S_8CH_TXFIFOLR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RW	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.
17:12	RW	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RW	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S_8CH_DMACR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 1'b0 : Receive DMA disabled 1'b1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 1'b0 : Transmit DMA disabled 1'b1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2S_8CH_INTCR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 1'b0: disable 1'b1: enable
16	RW	0x0	RXFIE RX full interrupt enable 1'b0: disable 1'b1: enable
15:9	RO	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.
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 1'b0: disable 1'b1: enable
0	RW	0x0	TXEIE TX empty interrupt enable 1'b0: disable 1'b1: enable

I2S_8CH_INTSR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 1'b0: inactive 1'b1: active
16	RO	0x0	RXFI RX full interrupt 1'b0: inactive 1'b1: active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 1'b0: inactive 1'b1: active
0	RO	0x0	TXEI TX empty interrupt 1'b0: inactive 1'b1: active

I2S_8CH_XFER

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 1'b0: stop RX transfer. 1'b1: start RX transfer
0	RW	0x0	TXS TX Transfer start bit 1'b0: stop TX transfer. 1'b1: start TX transfer

I2S_8CH_CLR

Address: Operational Base + offset (0x0020)

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_8CH_TXDR

Address: Operational Base + offset (0x0024)

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_8CH_RXDR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO data register. When the register is read, data in the receive FIFO is accessed.

I2S_8CH_RXFIFOLR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:18	RW	0x00	RFL3 Receive FIFO3 Level. Contains the number of valid data entries in the Receive FIFO3.
17:12	RW	0x00	RFL2 Receive FIFO2 Level. Contains the number of valid data entries in the Receive FIFO2.
11:6	RW	0x00	RFL1 Receive FIFO1 Level . Contains the number of valid data entries in the Receive FIFO1.
5:0	RW	0x00	RFL0 Receive FIFO0 Level. Contains the number of valid data entries in the Receive FIFO0.

I2S_8CH_TDM_TXCTRL

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:18	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL1 Tx TDM FSYNC width selection, working at TDM format. (Can be written only when XFER[0] is 0.) 3'b000: single period of the ASP_CLK. 3'b001: 2 period of the ASP_CLK. n: n+1 period of the ASP_CLK. 3'b110: 7 period of the ASP_CLK. 3'b111: the width is equivalent to a channel block Note: function when TX TFS[1:0] is 2 or 3;

Bit	Attr	Reset Value	Description
17	RW	0x0	TX_TDM_FSYNC_WIDTH_SEL0 I2S tx TDM FSYNC width selection, working at TDM format. (Can be written only when XFER[0] is 0.) 1'b0: 1/2 frame width. Aspc_ctrl1[8:0] should be set to an even number 1'b1: frame width
16:14	RW	0x0	TDM_TX_SHIFT_CTRL TDM tx shift control. Working at TDM mode. (Can be written only when XFER[0] is 0.) 3'b000: PCM format: normal mode, sample data on the third rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: normal mode 3'b001: PCM format: 1/2 cycle shift left, sample data on second falling rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: left justified mode 3'b010: PCM format: 1 cycle shift left, sample data on second rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: right justified mode 3'b011: PCM format: 3/2 cycle shift left, sample data on first falling edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b100: PCM format: 2 cycle shift left, sample data on first rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b101~3'b111 not support Note: function when TX TFS[1:0] is 2 or 3;
13:9	RW	0x1f	TDM_TX_SLOT_BIT_WIDTH TDM tx slot bit width. Working at TDM mode. (Can be written only when XFER[0] is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 31:32bit Note: function when TX TFS[1:0] is 2 or 3;

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_TX_FRAME_WIDTH TDM tx frame width. Working at TDM mode. (Can be written only when XFER[0] is 0.) 0~30:reserved 31:32bit 32:33bit 33:34bit 34:35bit …… n:(n+1)bit …… 511:512bit Note: functional when TX TFS[1:0] is 2 or 3;

I2S_8CH_TDM_RXCTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:18	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL1 Tx TDM FSYNC width selection, working at TDM format. (Can be written only when XFER[0] is 0.) 3'b000: single period of the ASP_CLK. 3'b001: 2 period of the ASP_CLK. n: n+1 period of the ASP_CLK. 3'b110: 7 period of the ASP_CLK. 3'b111: the width is equivalent to a channel block Note: function when RX TFS[1:0] is 2 or 3;
17	RW	0x0	RX_TDM_FSYNC_WIDTH_SEL0 I2S rx TDM FSYNC width selection, working at TDM format. (Can be written only when XFER[0] is 0.) 0: 1/2 frame width. Aspc_ctrl1[8:0] should be set to an even number 1: frame width

Bit	Attr	Reset Value	Description
16:14	RW	0x0	TDM_RX_SHIFT_CTRL TDM rx shift control. Working at TDM mode. (Can be written only when XFER[0] is 0.) 3'b000: PCM format: normal mode, sample data on the third rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: normal mode 3'b001: PCM format: 1/2 cycle shift left, sample data on second falling rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: left justified mode 3'b010: PCM format: 1 cycle shift left, sample data on second rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: right justified mode 3'b011: PCM format: 3/2 cycle shift left, sample data on first falling edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b100: PCM format: 2 cycle shift left, sample data on first rising edge of TDM_CLK after rising edge of ASPC_FSYNC. I2S format: not support 3'b101~3'b111 not support Note: function when RX TFS[1:0] is 2 or 3;
13:9	RW	0x1f	TDM_RX_SLOT_BIT_WIDTH TDM rx slot bit width. Working at TDM mode. (Can be written only when XFER[0] is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… n:(n+1)bit …… 31:32bit Note: function when RX TFS[1:0] is 2 or 3;

Bit	Attr	Reset Value	Description
8:0	RW	0x0ff	TDM_RX_FRAME_WIDTH TDM rx frame width. Working at TDM mode. (Can be written only when XFER[0] is 0.) 0~30:reserved 31:32bit 32:33bit 33:34bit 34:35bit …… n:(n+1)bit …… 511:512bit Note: functional when RX TFS[1:0] is 2 or 3;

I2S_8CH_CLKDIV

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

15:8	RW	0x07	RX_MDIV
			Mclk rx divider
			(Can be written only XFER[0] bit is 0.)
			Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequecy / txsclk frequecy-1)
			0 :Fmclk=Ftxsclk;
			1 :Fmclk=2*Ftxsclk;
			2,3 :Fmclk=4*Ftxsclk;
			4,5 :Fmclk=6*Ftxsclk;
			……
			2n,2n+1:Fmclk=(2n+2)*Ftxsclk;
			……
			60,61:Fmclk=62*Ftxsclk;
			62,63:Fmclk=64*Ftxsclk;
			……
			252,253:Fmclk=254*Ftxsclk;
			254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description

7:0	RW	0x07	TX_MDIV
			Mclk tx divider
			(Can be written only when XFER[1] bit is 0.)
			Serial Clock Divider = Fmclk / Ftxsclk-1.(mclkfrequecy / txsclk frequecy-1)
			0 :Fmclk=Ftxsclk;
			1 :Fmclk=2*Ftxsclk;
			2,3 :Fmclk=4*Ftxsclk;
			4,5 :Fmclk=6*Ftxsclk;
			……
			2n,2n+1:Fmclk=(2n+2)*Ftxsclk;
			……
			60,61:Fmclk=62*Ftxsclk;
			62,63:Fmclk=64*Ftxsclk;
			……
			252,253:Fmclk=254*Ftxsclk;
			254,255:Fmclk=256*Ftxsclk;

I2S_8CH_VERSION

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RO	0x20150001	I2S_VERSION i2s_version

I2S_8CH_INCR_RXDR

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	INCR_RXDR Increment address receive FIFO data register. When the register is read, data in the receive FIFO is accessed. This register is used when the access address is Increment.

19.5 Interface Description

Table 19-1 I2S Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
i2s0_8ch_mclk	I/O	GPIO2_A4/I2S0_8CH_MCLK/PDM_8CH_CLK_M_M2	GRF_GPIO2A_IOMUX[9:8]=2’b01
i2s0_8ch_sclk_tx	I/O	GPIO2_A5/I2S0_8CH_SCLK_TX	GRF_GPIO2A_IOMUX[11:10]=2’b01
i2s0_8ch_sclk_rx	I/O	GPIO2_A6/I2S0_8CH_SCLK_RX/PDM_8CH_CLK_S_M2	GRF_GPIO2A_IOMUX[13:12]=2’b01
i2s0_8ch_lrck_tx	I/O	GPIO2_A7/I2S0_8CH_LRCK_TX	GRF_GPIO2A_IOMUX[15:14]=2’b01
i2s0_8ch_lrck_rx	I/O	GPIO2_B0/I2S0_8CH_LRCK_RX	GRF_GPIO2B_IOMUX[1:0]=2’b01
i2s0_8ch_sdo0	O	GPIO2_B1/I2S0_8CH_SDO0	GRF_GPIO2B_IOMUX[3:2]=2’b01
i2s0_8ch_sdo1	O	GPIO2_B2/I2S0_8CH_SDO1	GRF_GPIO2B_IOMUX[5:4]=2’b01
i2s0_8ch_sdo2	O	GPIO2_B3/I2S0_8CH_SDO2	GRF_GPIO2B_IOMUX[7:6]=2’b01
i2s0_8ch_sdo3	O	GPIO2_B4/I2S0_8CH_SDO3	GRF_GPIO2B_IOMUX[9:8]=2’b01

Module Pin	Direction	Pin Name	IOMUX Setting
i2s0_8ch_sdi0	I	GPIO2_B5/I2S0_8CH_SDI0/PDM_8CH_SDI0_M2	GRF_GPIO2B_IOMUX[11:10]=2’b01
i2s0_8ch_sdi1	I	GPIO2_B6/I2S0_8CH_SDI1/PDM_8CH_SDI1_M2	GRF_GPIO2B_IOMUX[13:12]=2’b01
i2s0_8ch_sdi2	I	GPIO2_B7/I2S0_8CH_SDI2/PDM_8CH_SDI2_M2	GRF_GPIO2B_IOMUX[15:14]=2’b01
i2s0_8ch_sdi3	I	GPIO2_C0/I2S0_8CH_SDI3/PDM_8CH_SDI3_M2	GRF_GPIO2C_IOMUX[1:0]=2’b01
i2s1_8ch_mclk	I/O	GPIO1_A2/LCDC_VSYNC/I2S1_8CH_MCLK_M0	GRF_GPIO1A_IOMUX[5:4]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_B4/LCDC_D8/I2S1_8CH_MCLK_M1/MAC_CLK	GRF_GPIO1B_IOMUX_L[9:8]=2’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sclk_tx	I/O	GPIO1_A3/LCDC_DEN/I2S1_8CH_SCLK_TX_M0	GRF_GPIO1A_IOMUX[7:6]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_B5/LCDC_D9/I2S1_8CH_SCLK_TX_M1/MAC_MDC	GRF_GPIO1B_IOMUX_L[11:10]=2’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sclk_rx	I/O	GPIO1_A4/LCDC_D0/I2S1_8CH_SCLK_RX_M0/PDM_8CH_CLK_M0	GRF_GPIO1A_IOMUX[9:8]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_B6/LCDC_D10/I2S1_8CH_SCLK_RX_M1/PDM_8CH_CLK_M1/MAC_MDIO	GRF_GPIO1B_IOMUX_L[15:12]=4’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_lrck_tx	I/O	GPIO1_A5/LCDC_D1/I2S1_8CH_LRCK_TX_M0	GRF_GPIO1A_IOMUX[11:10]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_B7/LCDC_D11/I2S1_8CH_LRCK_TX_M1/MAC_RXER	GRF_GPIO1B_IOMUX_H[1:1]=2’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_lrck_rx	I/O	GPIO1_A6/LCDC_D2/I2S1_8CH_LRCK_RX_M0	GRF_GPIO1A_IOMUX[13:12]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_C0/LCDC_D12/I2S1_8CH_LRCK_RX_M1/MAC_RXDV	GRF_GPIO1C_IOMUX_L[1:0]=2’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sdo0	O	GPIO1_A7/LCDC_D3/I2S1_8CH_SDO0_M0	GRF_GPIO1A_IOMUX[15:14]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_C1/LCDC_D13/I2S1_8CH_SDO0_M1/MAC_TXEN	GRF_GPIO1C_IOMUX_L[3:2]=2’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sdo1/sdi3	I/O	GPIO1_B0/LCDC_D4/I2S1_8CH_SDO1_SDI3_M0/PDM_8CH_SDI3_M0	GRF_GPIO1B_IOMUX_L[1:0]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_C2/LCDC_D14/I2S1_8CH_SDO1_SDI3_M1/PDM_8CH_SDI3_M1/MAC_TXD0	GRF_GPIO1C_IOMUX_L[7:4]=4’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sdo2/sdi2	I/O	GPIO1_B1/LCDC_D5/I2S1_8CH_SDO2_SDI2_M0/PDM_8CH_SDI2_M0	GRF_GPIO1B_IOMUX_L[3:2]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_C3/LCDC_D15/I2S1_8CH_SDO2_SDI2_M1/PDM_8CH_SDI2_M1/MAC_TXD1	GRF_GPIO1C_IOMUX_L[11:8]=4’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sdo3/sdi1	I/O	GPIO1_B2/LCDC_D6/I2S1_8CH_SDO3_SDI1_M0/PDM_8CH_SDI1_M0	GRF_GPIO1B_IOMUX_L[5:4]=2’b10 GRF_SOC_CON2[3]=1’b0
		GPIO1_C4/LCDC_D16/I2S1_8CH_SDO3_SDI1_M1/PDM_8CH_SDI1_M1/MAC_RXD0	GRF_GPIO1C_IOMUX_L[15:12]=4’b10 GRF_SOC_CON2[3]=1’b1
i2s1_8ch_sdi0	I	GPIO1_B3/LCDC_D7/I2S1_8CH_SDI0_M0/PDM_8CH_SDI0_M0	GRF_GPIO1B_IOMUX_L[7:6]=2’b10 GRF_SOC_CON2[3]=1’b0

Module Pin	Direction	Pin Name	IOMUX Setting
		GPIO1_C5/LCDC_D17/I2S1_8CH_SDI0_M1/PDM_8CH_SDI0_M1/MAC_RXD1	GRF_GPIO1C_IOMUX_H[3:0]=4'b10 GRF_SOC_CON2[3]=1'b1

Notes:
1. I=input, O=output, I/O=input/output, bidirectional
2. The direction of i2s0_8ch_mclk *is controlled by CRU register, it's controlled by CRU_CLKGATE_CON13[4] when GRF_SOC_CON2[10] is set to 1'b0, it's controlled by CRU_CLKGATE_CON13[5] when GRF_SOC_CON2[10] is set to 1'b1:*
1'b1: input
1'b0: output
3. The direction of i2s1_8ch_mclk *is controlled by CRU register, it's controlled by CRU_CLKGATE_CON13[6] when GRF_SOC_CON2[2] is set to 1'b0, it's controlled by CRU_CLKGATE_CON13[7] when GRF_SOC_CON2[10] is set to 1'b1:*
1'b1: input
1'b0: output
4. The direction of i2s1_8ch_sdo1/sdi3 is controlled by *GRF_SOC_CON2[5]*
1'b1: sdo1 output
1'b0: sdi3 input
5. The direction of i2s1_8ch_sdo2/sdi2 is controlled by *GRF_SOC_CON2[6]:*
1'b1: sdo2 output
1'b0: sdi2 input
6. The direction of i2s1_8ch_sdo3/sdi1 is controlled by *GRF_SOC_CON2[7]:*
1'b1: sdo3 output
1'b0: sdi1 input

19.6 Application Notes

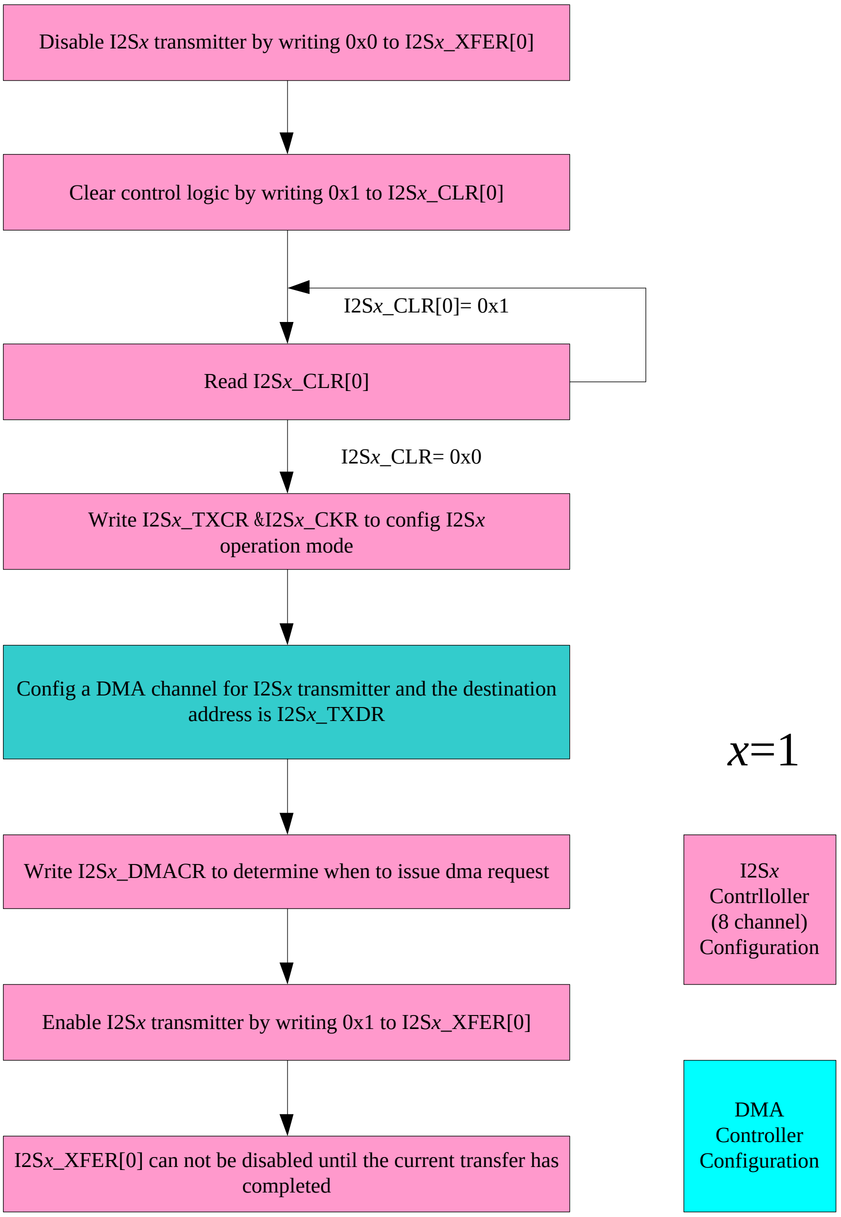

Disable I2Sx transmitter by writing 0x0 to I2Sx_XFER[0]
Clear control logic by writing 0x1 to I2Sx_CLR[0]
I2Sx_CLR[0]= 0x1
Read I2Sx_CLR[0]
I2Sx_CLR= 0x0
Write I2Sx_TXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx transmitter and the destination address is I2Sx_TXDR
x=1
Write I2Sx_DMACR to determine when to issue dma request
I2Sx Contrlloller (8 channel) Configuration
Enable I2Sx transmitter by writing 0x1 to I2Sx_XFER[0]
DMA Controller Configuration
I2Sx_XFER[0] can not be disabled until the current transfer has completed

Fig. 19-11 I2S/PCM/TDM controller transmit operation flow chart

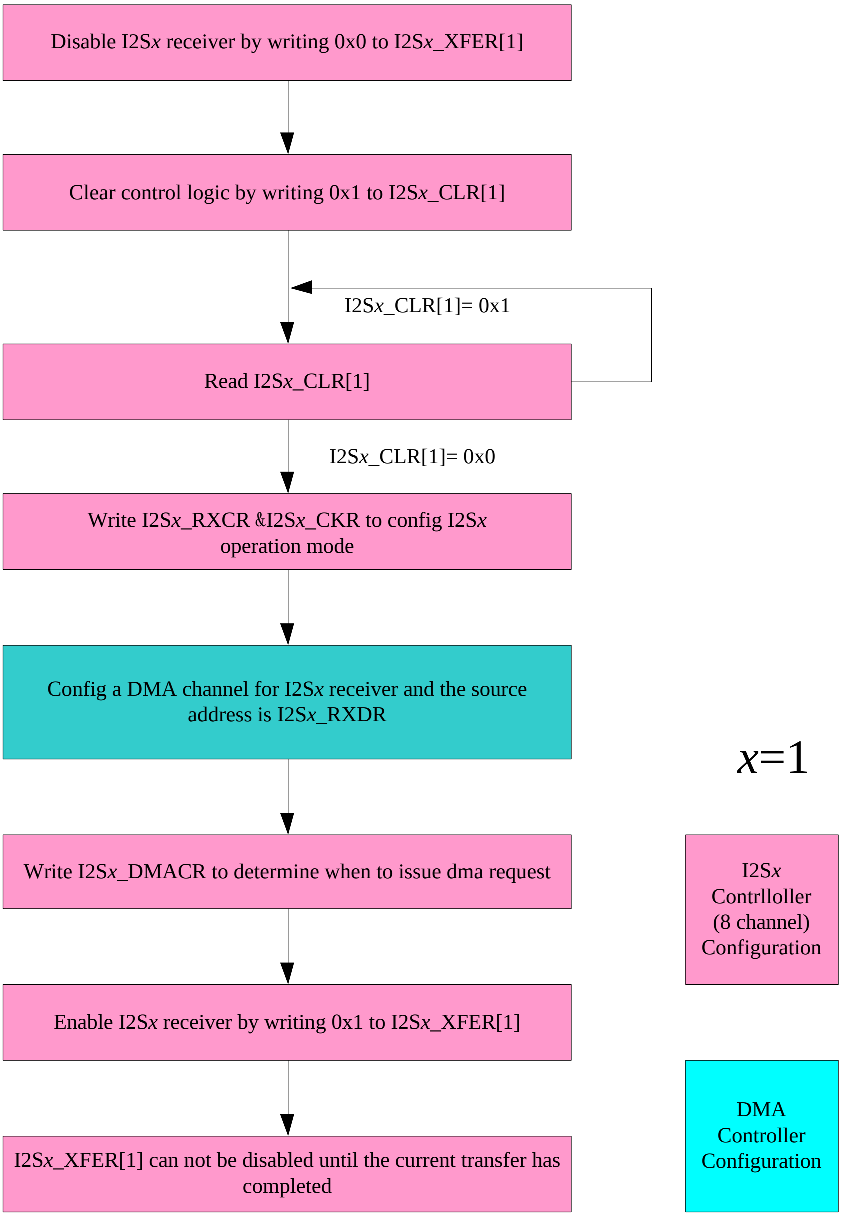

Disable I2Sx receiver by writing 0x0 to I2Sx_XFER[1]
Clear control logic by writing 0x1 to I2Sx_CLR[1]
I2Sx_CLR[1]= 0x1
Read I2Sx_CLR[1]
I2Sx_CLR[1]= 0x0
Write I2Sx_RXCR &I2Sx_CKR to config I2Sx operation mode
Config a DMA channel for I2Sx receiver and the source address is I2Sx_RXDR
Write I2Sx_DMACR to determine when to issue dma request
Enable I2Sx receiver by writing 0x1 to I2Sx_XFER[1]
I2Sx_XFER[1] can not be disabled until the current transfer has completed
x=1
I2Sx Contrlloller (8 channel) Configuration
DMA Controller Configuration

Fig. 19-12 I2S/PCM/TDM controller receive operation flow chart

Note: User should clear TX/RX logical by CLR[0]/CLR[1] and wait clear operation done before configure the other registers.

There are 3 operation modes which are normal mode, TX common mode and RX common mode. Following shows the normal mode connections, I2S_CKR[29:28] should be set to 2'b00.

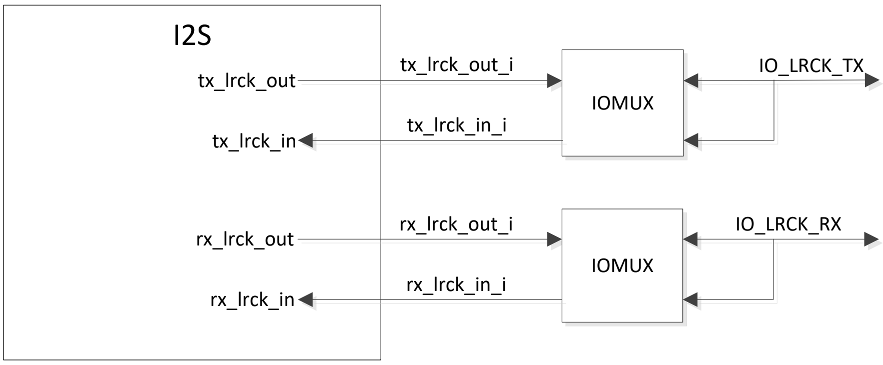


Fig. 19-13 I2S/PCM normal mode

For TX common mode, only IO_LRCK_TX is used and I2S_CKR[29:28] should be set to 2'b01. TX/RX should work at the same time and at the same sample rate.

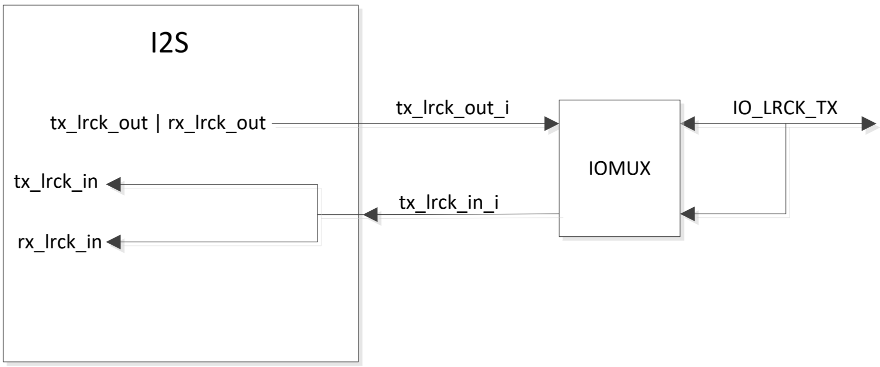


Fig. 19-14 I2S/PCM TX common mode

For RX common mode, only IP_LRCK_RX is used and I2S_CKR[29:28] should be set to 2'b10. TX/RX should work at the same time and at the same sample rate.

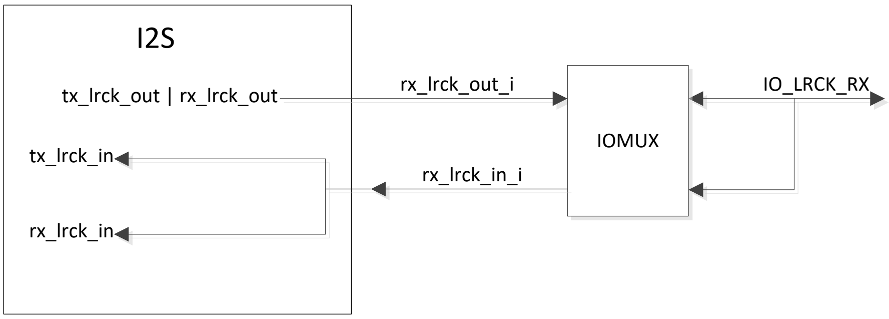


Fig. 19-15 I2S/PCM RX common mode

Chapter 20 Serial Peripheral Interface (SPI)

20.1 Overview

The serial peripheral interface is an APB slave device. A four wire full duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

20.2 Block Diagram

The SPI Controller comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt

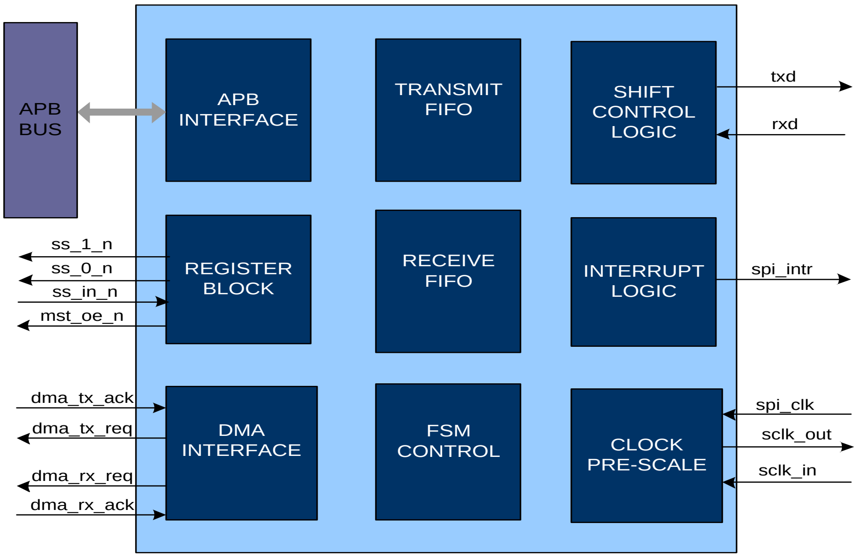


Fig. 20-1 SPI Controller Block diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits and 8 or 16 bits when reading or writing internal FIFO if data frame size(SPI_CTRL0[1:0]) is set to 8 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serial device into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

20.3 Function Description

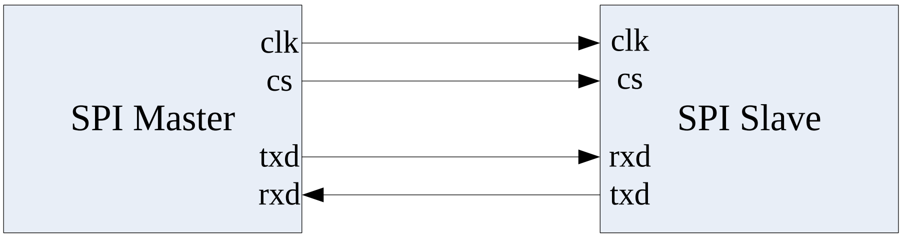


Fig. 20-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.
1). Transmit and Receive
When SPI_CTRLR0 [19:18]== 2'b00, both transmit and receive logic are valid.
2).Transmit Only
When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.
3).Receive Only
When SPI_CTRLR0 [19:18]== 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,
When SPI Controller works as master, the F_{spi_clk}>= 2 × (maximum F_{sclk_out})
When SPI Controller works as slave, the F_{spi_clk}>= 6 × (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.
When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. The following two figures show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

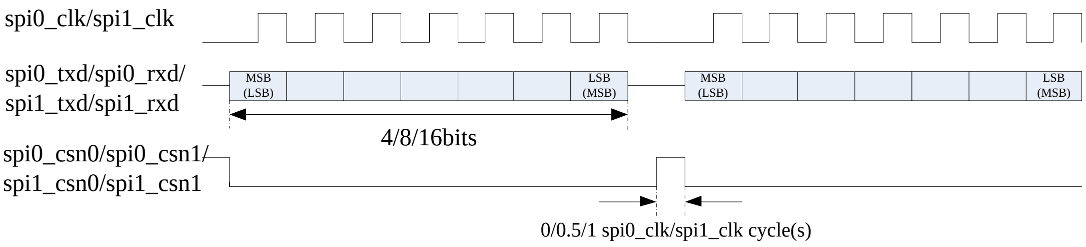


Fig. 20-3 SPI Format (SCPH=0 SCPOL=0)

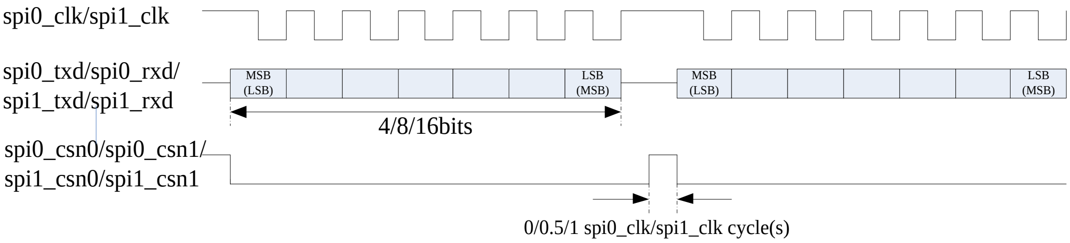


Fig. 20-4 SPI Format (SCPH=0 SCPOL=1)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. The following two figures show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

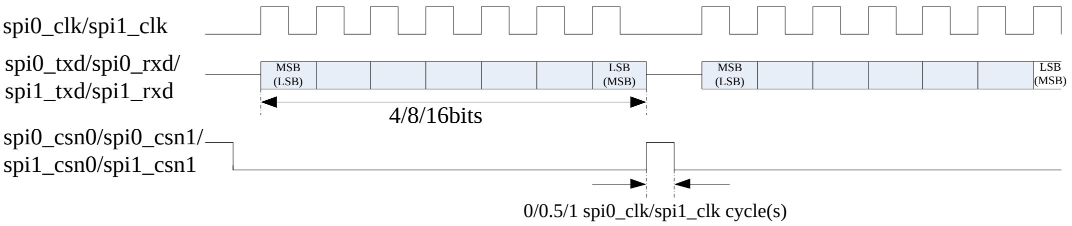


Fig. 20-5 SPI Format (SCPH=1 SCPOL=0)

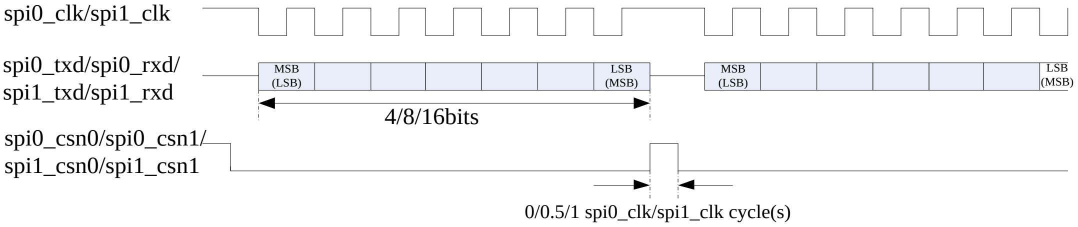


Fig. 20-6 SPI Format (SCPH=1 SCPOL=1)

20.4 Register Description

20.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI_CTRLR0	0x0000	W	0x00000002	Control Register 0
SPI_CTRLR1	0x0004	W	0x00000000	Control Register 1
SPI_ENR	0x0008	W	0x00000000	SPI Enable Register
SPI_SER	0x000c	W	0x00000000	Slave Enable Register
SPI_BAUDR	0x0010	W	0x00000000	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x00000000	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x00000000	Receive FIFO Threshold Level
SPI_TXFLR	0x001c	W	0x00000000	Transmit FIFO Level
SPI_RXFLR	0x0020	W	0x00000000	Receive FIFO Level
SPI_SR	0x0024	W	0x0000000c	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status

Name	Offset	Size	Reset Value	Description
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

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

20.4.2 Detail Register Description

SPI_CTRLR0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	MTM Valid when frame format is set to National Semiconductors Microwire. 1'b0: non-sequential transfer 1'b1: sequential transfer
20	RW	0x0	OPM 1'b0: Master Mode 1'b1: Slave Mode
19:18	RW	0x0	XFM 2'b00: Transmit & Receive 2'b01: Transmit Only 2'b10: Receive Only 2'b11: Reserved
17:16	RW	0x0	FRF 2'b00: Motorola SPI 2'b01: Texas Instruments SSP 2'b10: National Semiconductors Microwire 2'b11: reserved
15:14	RW	0x0	RSD When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 2'b00: do not delay 2'b01: 1 cycle delay 2'b10: 2 cycles delay 2'b11: 3 cycles delay

Bit	Attr	Reset Value	Description
13	RW	0x0	BHT Valid when data frame size is 8bit. 1'b0: apb 16bit write/read, spi 8bit write/read 1'b1: apb 8bit write/read, spi 8bit write/read
12	RW	0x0	FBM 1'b0: first bit is MSB 1'b1: first bit is LSB
11	RW	0x0	EM Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 1'b0: little endian 1'b1: big endian
10	RW	0x0	SSD Valid when the frame format is set to Motorola SPI and SPI used as a master. 1'b0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1'b1: the period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0x0	CSM Valid when the frame format is set to Motorola SPI and SPI used as a master. 2'b00: ss_n keep low after every frame data is transferred. 2'b01: ss_n be high for half sclk_out cycles after every frame data is transferred. 2'b10: ss_n be high for one sclk_out cycle after every frame data is transferred. 2'b11: reserved
7	RW	0x0	SCPOL Valid when the frame format is set to Motorola SPI. 1'b0: inactive state of serial clock is low 1'b1: inactive state of serial clock is high
6	RW	0x0	SCPH Valid when the frame format is set to Motorola SPI. 1'b0: serial clock toggles in middle of first data bit 1'b1: serial clock toggles at start of first data bit

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CFS Selects the length of the control word for the Microwire frame format. 4'b0000~0010: reserved 4'b0011: 4-bit serial data transfer 4'b0100: 5-bit serial data transfer 4'b0101: 6-bit serial data transfer 4'b0110: 7-bit serial data transfer 4'b0111: 8-bit serial data transfer 4'b1000: 9-bit serial data transfer 4'b1001: 10-bit serial data transfer 4'b1010: 11-bit serial data transfer 4'b1011: 12-bit serial data transfer 4'b1100: 13-bit serial data transfer 4'b1101: 14-bit serial data transfer 4'b1110: 15-bit serial data transfer 4'b1111: 16-bit serial data transfer
1:0	RW	0x2	DFS Selects the data frame length. 2'b00: 4bit data 2'b01: 8bit data 2'b10: 16bit data 2'b11: reserved

SPI_CTRLR1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	NDM When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ENR Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	SER This register is valid only when SPI is configured as a master device.

SPI_BAUDR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	BAUDR SPI Clock Divider. This register is valid only when the SPI is configured as a master device. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	TXFTLR When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RXFTLR When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	TXFLR Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	RXFLR Contains the number of valid data entries in the receive FIFO.

SPI_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFF 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RO	0x1	RFE 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RO	0x1	TFE 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RO	0x0	TFF 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RO	0x0	BSF When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 1'b0: SPI is idle or disabled 1'b1: SPI is actively transferring data

SPI_IPR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	IPR Interrupt Polarity Register 1'b0: Active Interrupt Polarity Level is HIGH 1'b1: Active Interrupt Polarity Level is LOW

SPI_IMR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	RFFIM 1'b0: spi_rxf_intr interrupt is masked 1'b1: spi_rxf_intr interrupt is not masked
3	RW	0x0	RFOIM 1'b0: spi_rxo_intr interrupt is masked 1'b1: spi_rxo_intr interrupt is not masked
2	RW	0x0	RFUIM 1'b0: spi_rxu_intr interrupt is masked 1'b1: spi_rxu_intr interrupt is not masked
1	RW	0x0	TFOIM 1'b0: spi_txo_intr interrupt is masked 1'b1: spi_txo_intr interrupt is not masked
0	RW	0x0	TFEIM 1'b0: spi_txe_intr interrupt is masked 1'b1: spi_txe_intr interrupt is not masked

SPI_ISR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	RFFIS 1'b0: spi_rxf_intr interrupt is not active after masking 1'b1: spi_rxf_intr interrupt is full after masking
3	RO	0x0	RFOIS 1'b0: spi_rxo_intr interrupt is not active after masking 1'b1: spi_rxo_intr interrupt is active after masking
2	RO	0x0	RFUIS 1'b0: spi_rxu_intr interrupt is not active after masking 1'b1: spi_rxu_intr interrupt is active after masking
1	RO	0x0	TFOIS 1'b0: spi_txo_intr interrupt is not active after masking 1'b1: spi_txo_intr interrupt is active after masking
0	RO	0x0	TFEIS 1'b0: spi_txe_intr interrupt is not active after masking 1'b1: spi_txe_intr interrupt is active after masking

SPI_RISR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RO	0x0	RFFRIS 1'b0: spi_rxf_intr interrupt is not active prior to masking 1'b1: spi_rxf_intr interrupt is full prior to masking
3	RO	0x0	RFORIS 1'b0: spi_rxo_intr interrupt is not active prior to masking 1'b1: spi_rxo_intr interrupt is active prior to masking
2	RO	0x0	RFURIS 1'b0: spi_rxu_intr interrupt is not active prior to masking 1'b1: spi_rxu_intr interrupt is active prior to masking
1	RO	0x0	TFORIS 1'b0: spi_txo_intr interrupt is not active prior to masking 1'b1: spi_txo_intr interrupt is active prior to masking
0	RO	0x1	TFERIS 1'b0: spi_txe_intr interrupt is not active prior to masking 1'b1: spi_txe_intr interrupt is active prior to masking

SPI_ICR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	W1C	0x0	CTFOI Write 1 to Clear Transmit FIFO Overflow Interrupt
2	W1C	0x0	CRFOI Write 1 to Clear Receive FIFO Overflow Interrupt
1	W1C	0x0	CRFUI Write 1 to Clear Receive FIFO Underflow Interrupt
0	W1C	0x0	CCI Write 1 to Clear Combined Interrupt

SPI_DMACR

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TDE 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
0	RW	0x0	RDE 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled

SPI_DMATDLR

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	TDL This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	RDL This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	WO	0x0000	TXDR When it is written to, data are moved into the transmit FIFO.

SPI_RXDR

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RXDR When the register is read, data in the receive FIFO is accessed.

20.5 Interface Description

Table 20-1 1SPI interface description

Module Pin	Direction	Pin Name	IOMUX Setting
spi0_clk	I/O	GPIO2_A2/UART0_CTSN/SPI0_CLK/I2C2_SDA	GRF_GPIO2A_IOMUX[5:4]=2’b10
spi0_miso	I	GPIO2_A0/UART0_RX/SPI0_MISO	SPI Slave mode: GRF_GPIO2A_IOMUX[1:0]=2’b10
spi0_mosi	O	GPIO2_A1/UART0_TX/SPI0_MOSI	SPI Master mode: GRF_GPIO2A_IOMUX[3:2]=2’b10
spi0_csn0	I/O	GPIO2_A3/UART0_RTSN/SPI0_CSN0/I2C2_SCL	GRF_GPIO2A_IOMUX[7:6]=2’b10

Module Pin	Direction	Pin Name	IOMUX Setting
spi1_clk	I/O	GPIO3_B3/FLASH_ALE/EMMC_PWREN/SPI1_CLK	GRF_GPIO3B_IOMUX[7:6]=2'b11
spi1_miso	I	GPIO3_B2/FLASH_RDN/SPI1_MISO	SPI Slave mode: GRF_GPIO3B_IOMUX[5:4]=2'b11
spi1_mosi	O	GPIO3_B4/FLASH_RDY/I2C3_SDA_M1/SPI1_MOSI/UART3_RX	SPI Master mode: GRF_GPIO3B_IOMUX[11:8]=4'b0011
spi1_csn0	O	GPIO3_B5/FLASH_CSN0/I2C3_SCL_M1/SPI1_CSN0/UART3_TX	GRF_GPIO3B_IOMUX[15:12]=4'b0011
spi2_clk	I/O	GPIO1_D0/UART1_RX/I2C0_SDA/SPI2_CLK	GRF_GPIO1D_IOMUX[1:0]=2'b11
spi2_miso	I	GPIO1_C6/UART1_CTSN/UART2_RX_M0/SPI2_MISO/JTAG_TCK	SPI Slave mode: GRF_GPIO1C_IOMUX_H[7:4]=4'b0011
spi2_mosi	O	GPIO1_C7/UART1_RTSN/UART2_TX_M0/SPI2_MOSI/JTAG_TMS	SPI Master mode: GRF_GPIO1C_IOMUX_H[11:8]=4'b0011
spi2_csn0	O	GPIO1_D1/UART1_TX/I2C0_SCL/SPI2_CSN0	GRF_GPIO1D_IOMUX[3:2]=2'b11

Notes: I=input, O=output, I/O=input/output, bidirectional. spi_csn1 can only be used in master mode

20.6 Application Notes

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,
When SPI Controller works as master, the Fspi_clk>= 2 × (maximum Fsclk_out)
When SPI Controller works as slave, the Fspi_clk>= 6 × (maximum Fsclk_in)

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.
When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

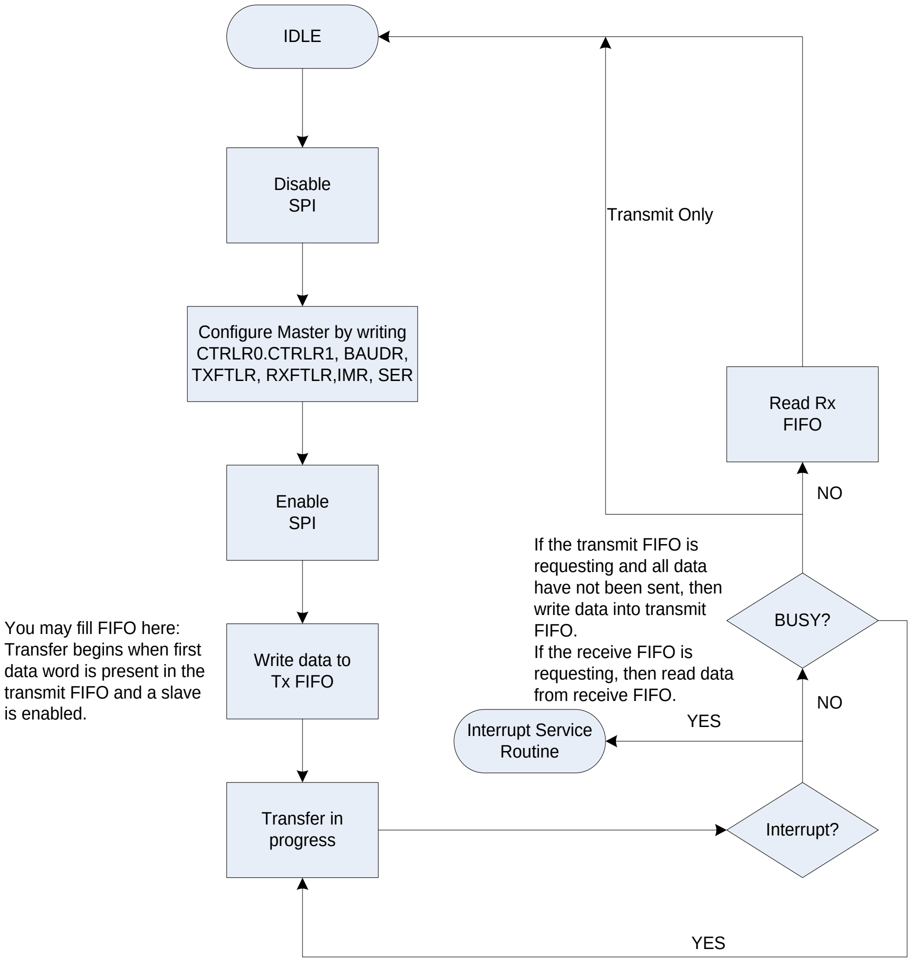

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, BAUDR, TXFTLR, RXFTLR,IMR, SER
Enable SPI
Write data to Tx FIFO
Transfer in progress
You may fill FIFO here: Transfer begins when first data word is present in the transmit FIFO and a slave is enabled.
Transmit Only
Read Rx FIFO
NO
If the transmit 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

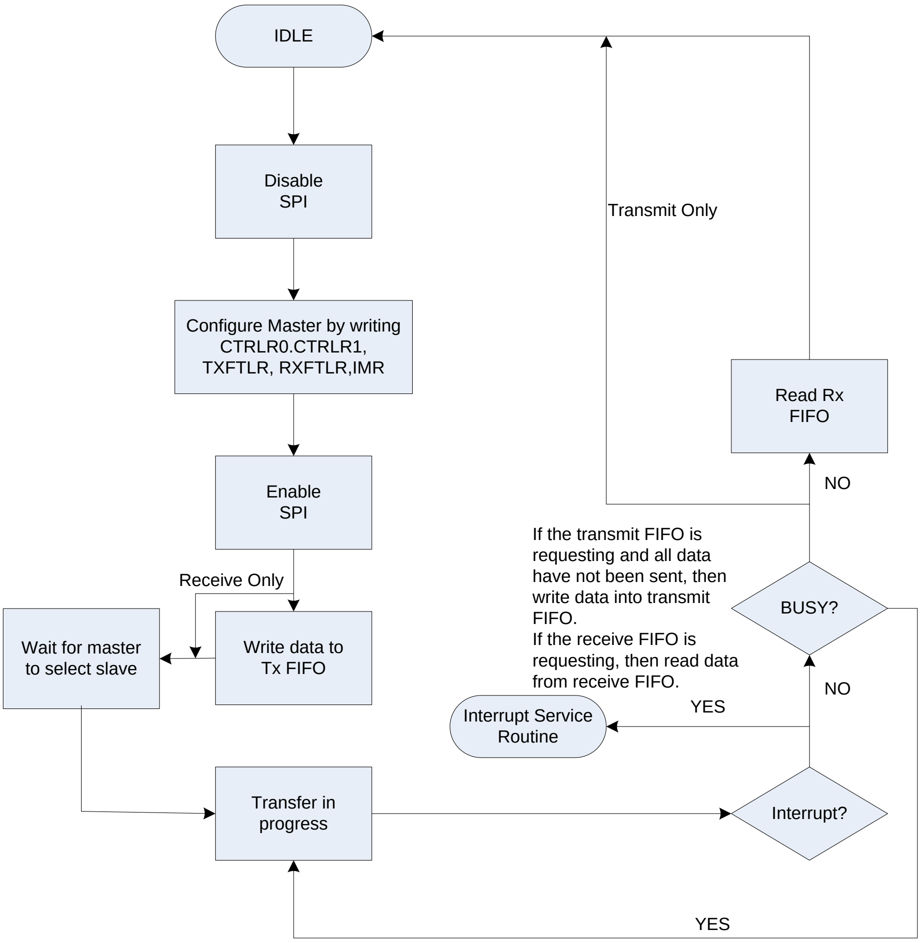

IDLE
Disable SPI
Configure Master by writing CTRLR0.CTRLR1, TXFTLR, RXFTLR,IMR
Enable SPI
Receive Only
Wait for master to select slave
Write data to Tx FIFO
Transfer in progress
Interrupt?
YES
Interrupt Service Routine
NO
BUSY?
NO
Read Rx FIFO
Transmit Only
YES
If the transmit FIFO is requesting and 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 Serial Flash Controller (SFC)

21.1 Overview

The serial flash controller (SFC) is used to control the data transfer between the chip system and the serial NOR/NAND flash device.
The SFC supports the following features:

- Support AHB slave interface to configure register and read/write serial flash
- Support AHB master interface to transfer data from/to SPI flash device
- Support AHB burst with INCR4 x32bits, or INCR x32bits
- Support two independent clock domain: AHB clock and SPI clock
- Support x1, x2, x4 data bits mode
- Support up to 1 chip selections
- Support interrupt output, interrupt maskable

21.2 Block Diagram

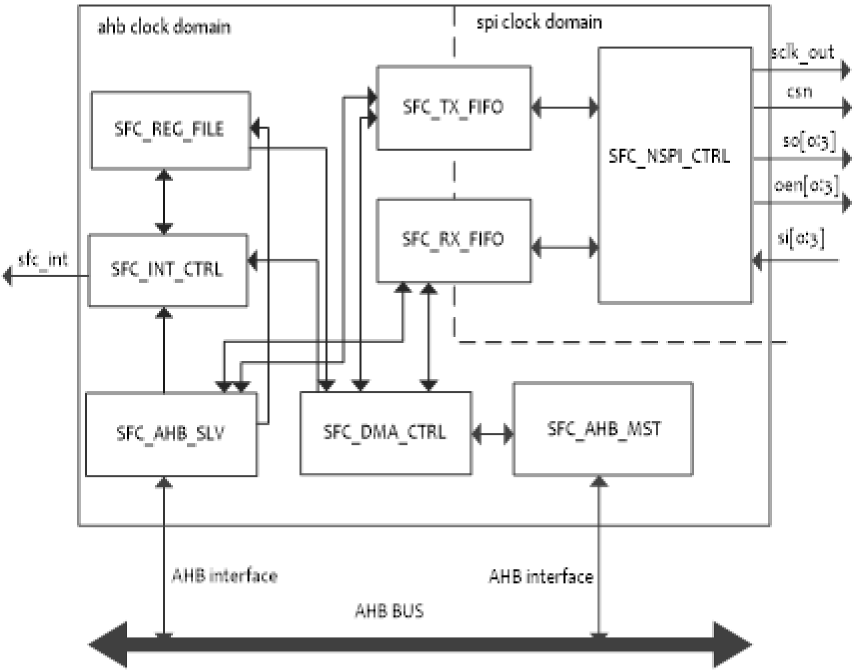


Fig. 21-1 SFC Architecture

21.3 Function Description

21.3.1 SFC AHB slave

The AHB slave is used to configure the register, and also write to/read from the serial NOR/NAND flash device.
The SFC_CTRL register is a global control register, when the controller is in busy state (SFC_SR), SFC_CTRL cannot be set. The field sclk_idle_level_cycles(SFC_CTRL[7:4]) of this register are used to configure the idle level cycles of SFC core clock (sfc_sclk) before reading the first bit of the read command.
Like the following picture shows: the red line of the sclk is the idle cycles, during these cycles, the chip pad is switched to output. When sclk_idle_level_cycles=0, it means there will be not such idle level.

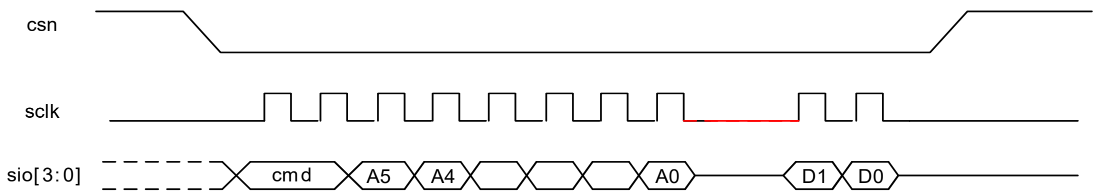


Fig. 21-2 idle cycles

When the field spi mode is set, the transfer waveform will like following, and switch to mode3.

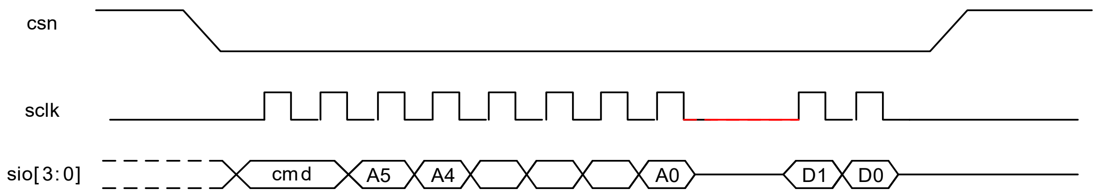


Fig. 21-3 SPI mode

21.4 Register Description

21.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SFC_CTRL	0x0000	W	0x00000000	Control Register
SFC_IMR	0x0004	W	0x00000000	Interrupt Mask
SFC_ICLR	0x0008	W	0x00000000	Interrupt Clear
SFC_FTLR	0x000c	W	0x00000000	FIFO Threshold Level
SFC_RCVR	0x0010	W	0x00000000	SFC Recover
SFC_AX	0x0014	W	0x00000000	SFC AX Value
SFC_ABIT	0x0018	W	0x00000000	Flash Address bits
SFC_ISR	0x001c	W	0x00000000	Interrupt Status
SFC_FSR	0x0020	W	0x00000001	FIFO Status
SFC_SR	0x0024	W	0x00000000	SFC Status
SFC_RISR	0x0028	W	0x00000000	Raw Interrupt Status
SFC_VER	0x002c	W	0x0a340003	Version Register
SFC_QOP	0x0030	W	0x00000000	Quad line operation io level preset
SFC_DMATR	0x0080	W	0x00000000	DMA Trigger
SFC_DMAADDR	0x0084	W	0x00000000	DMA Address
SFC_CMD	0x0100	W	0x00000000	SFC CMD
SFC_ADDR	0x0104	W	0x00000000	Address of data to read or write.
SFC_DATA	0x0108	W	0x00000000	DATA that write to or read from the flash.

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

21.4.2 Detail Register Description

SFC_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	DATB Data line width. 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: reserved
11:10	RW	0x0	ADRB Address bits width. 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: reserved
9:8	RW	0x0	CMDB Command bits width. 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: reserved
7:4	RW	0x0	IDLE_CYCLE 4'd0: idle hold is disable 4'd1: hold the sclk_out in idle for two cycles when switch to shift in ....
3:2	RO	0x0	reserved
1	RW	0x0	SHIFTPHASE 1'b0: shift in the data at posedge sclk_out 1'b1: shift in the data at negedge sclk_out
0	RW	0x0	SPIM SPI MODE Select. 1'b0: mode 0 1'b1: mode 3

SFC_IMR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	DMAM DMA finish interrupt mask 1'b0: dma_intr interrupt is not masked 1'b1: dma_intr interrupt is masked
6	RW	0x0	NSPIM SPI error interrupt mask 1'b0: nspi_intr interrupt is not masked 1'b1: nspi_intr interrupt is masked

Bit	Attr	Reset Value	Description
5	RW	0x0	AHBM AHB error interrupt mask 1'b0: ahb_intr interrupt is not masked 1'b1: ahb_intr interrupt is masked
4	RW	0x0	TRANSM Transfer finish interrupt mask 1'b0: transf_intr interrupt is not masked 1'b1: transf_intr interrupt is masked
3	RW	0x0	TXEM Transmit FIFO empty interrupt 1'b0: txe_intr interrupt is not masked 1'b1: txe_intr interrupt is masked
2	RW	0x0	TXOM Transmit FIFO overflow interrupt mask 1'b0: txo_intr interrupt is not masked 1'b1: txo_intr interrupt is masked
1	RW	0x0	RXUM Receive FIFO underflow interrupt mask 1'b0: rxu_intr interrupt is not masked 1'b1: rxu_intr interrupt is masked
0	RW	0x0	RXFM Receive FIFO full interrupt mask 1'b0: rxf_intr interrupt is not masked 1'b1: rxf_intr interrupt is masked

SFC_ICLR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	DMAC DMA finish Interrupt Clear.
6	W1C	0x0	NSPIC SPI Error Interrupt Clear.
5	W1C	0x0	AHBC AHB Error Interrupt Clear.
4	W1C	0x0	TRANSC Transfer finish Interrupt Clear.
3	W1C	0x0	TXEC Transmit FIFO Empty Interrupt Clear.
2	W1C	0x0	TXOC Transmit FIFO Overflow Interrupt Clear.
1	W1C	0x0	RXUC Receive FIFO Underflow Interrupt Clear.

Bit	Attr	Reset Value	Description
0	W1C	0x0	RXFC Receive FIFO Full Interrupt Clear.

SFC_FTLR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	RXFTLR When the number of receive FIFO entries is bigger than or equal to this value, the receive FIFO full interrupt is triggered.
7:0	RW	0x00	TXFTLR When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SFC_RCVR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	RCVR SFC Recover. Write 1 to recover the SFC State Machine, FIFO state and other logic state.

SFC_AX

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	AX The AX Value when doing the continuous read (enhance mode). That is M7-M0 in “Continuous Read Mode”.

SFC_ABIT

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	ABIT Flash Address bits.

SFC_ISR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RO	0x0	DMAS DMA Finish Interrupt Status. 1'b0: not active 1'b1: active
6	RO	0x0	NSPIS SPI Error Interrupt Status. 1'b0: not active 1'b1: active
5	RO	0x0	AHBS AHB Error Interrupt Status. 1'b0: not active 1'b1: active
4	RO	0x0	TRANSS Transfer finish Interrupt Status. 1'b0: not active 1'b1: active
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status. 1'b0: not active 1'b1: active
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status. 1'b1: active
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status. 1'b0: not active 1'b1: active
0	RW	0x0	RXFS Receive FIFO Full Interrupt Status. 1'b0: not active 1'b1: active

SFC_FSR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20:16	RW	0x00	RXWLVL RX FIFO Water Level. 5'h0: FIFO is empty 5'h1: 1 entry is taken ... 5'h10: 16 entry is taken, FIFO is full
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:8	RO	0x00	TXWLVL TX FIFO Water Level. 5'h0: FIFO is full 5'h1: left 1 entry ... 5'h10: left 16 entry, FIFO is empty
7:4	RO	0x0	reserved
3	RO	0x0	RXFS Receive FIFO Full Status. 1'b0: rx FIFO is not full 1'b1: rx FIFO is full
2	RO	0x0	RXES Receive FIFO Empty Status. 1'b0: rx FIFO is not empty 1'b1: rx FIFO is empty
1	RO	0x0	TXES Transmit FIFO Empty Status. 1'b0: tx FIFO is not empty 1'b1: tx FIFO is empty
0	RO	0x1	TXFS Transmit FIFO Full Status. 1'b0: tx FIFO is not full 1'b1: tx FIFO is full

SFC_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	SR 1'b0: SFC is idle 1'b1: SFC is busy When busy, don't set the control register. When idle, the rx FIFO and tx FIFO are all empty.

SFC_RISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	DMAS DMA Finish Interrupt Status. 1'b0: not active 1'b1: active

Bit	Attr	Reset Value	Description
6	RO	0x0	NSPIS SPI Error Interrupt Status. 1'b0: not active 1'b1: active
5	RO	0x0	AHBS AHB Error Interrupt Status. 1'b0: not active 1'b1: active
4	RO	0x0	TRANSS Transfer finish Interrupt Status. 1'b0: not active 1'b1: active
3	RO	0x0	TXES Transmit FIFO Empty Interrupt Status. 1'b0: not active 1'b1: active
2	RO	0x0	TXOS Transmit FIFO Overflow Interrupt Status. 1'b0: not active 1'b1: active
1	RO	0x0	RXUS Receive FIFO Underflow Interrupt Status. 1'b0: not active 1'b1: active
0	RO	0x0	RXFS Receive FIFO Full Interrupt Status. 1'b0: not active 1'b1: active

SFC_VER

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	Reserved.
15:0	RW	0x3	VER The version id of sfc.

SFC_QOP

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	SO123 The value of SIO1, SIO2 and SIO3 during command and address bits output.

SFC_DMATR
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1C	0x0	DMATR Write 1 to start the dma transfer.

SFC_DMAADDR
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAADDR DMA Address.

SFC_CMD
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	WO	0x0	CS Flash chip select. 2'b00: chip select 0 Others: reserved
29:16	WO	0x0000	TRB Total Data Bytes number that will write to /read from the flash. In DMA mode, this register must be aligned to 2 bytes.
15:14	WO	0x0	ADDRB Address bits number select, if there is not address command to send, set to zero. 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register
13	WO	0x0	CONT Continuous read mode. 1'b0: disable continuous read mode 1'b1: enable continuous read mode
12	WO	0x0	WR Flash Write or Read. 1'b0: read 1'b1: write
11:8	WO	0x0	DUMM Dummy Bits Number.
7:0	WO	0x00	CMD Flash Command.

SFC_ADDR
Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ADDR Flash's address.

SFC_DATA

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DATA Flash's Data. The LSB of this data will be sent first.

21.5 Interface Description

Table 21-11SFC interface description

Module Pin	Direction	Pin Name	IOMUX Setting
sfc_clk	O	GPIO3_A4/FLASH_D4/EMMC_D4/SFC_CLK	GRF_GPIO3A_IOMUX_SEL[9:8]=2’b11
sfc_csn0	O	GPIO3_A5/FLASH_D5/EMMC_D5/SFC_CSN0	GRF_GPIO3A_IOMUX_SEL[11:10]=2’b11
sfc_sio0	I/O	GPIO3_A0/FLASH_D0/EMMC_D0/SFC_SIO0	GRF_GPIO3A_IOMUX_SEL[1:0]=2’b11
sfc_sio1	I/O	GPIO3_A1/FLASH_D1/EMMC_D1/SFC_SIO1	GRF_GPIO3A_IOMUX_SEL[3:2]=2’b11
sfc_sio2	I/O	GPIO3_A2/FLASH_D2/EMMC_D2/SFC_WP_SIO2	GRF_GPIO3A_IOMUX_SEL[5:4]=2’b11
sfc_sio3	I/O	GPIO3_A3/FLASH_D3/EMMC_D3/SFC_HOLD_SIO3	GRF_GPIO3A_IOMUX_SEL[7:6]=2’b11

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

21.6 Application Notes

21.6.1 AHB Slave write flash flow (non-DMA mode)

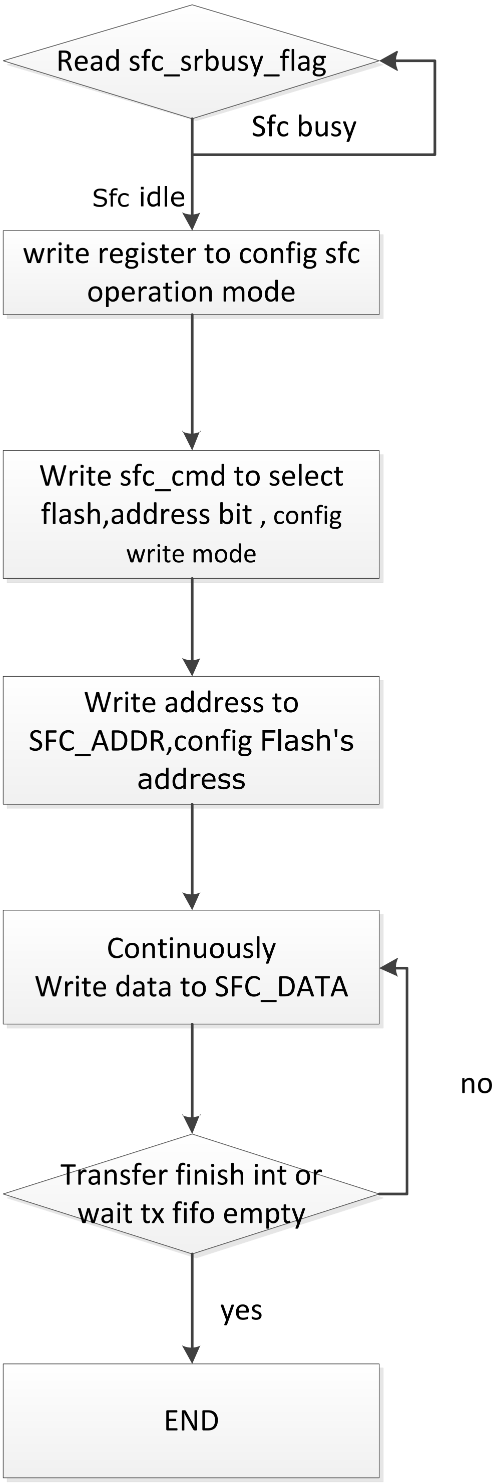

Read sfc_srbusy_flag
Sfc busy
Sfc idle
write register to config sfc operation mode
Write sfc_cmd to select flash,address bit , config write mode
Write address to SFC_ADDR,config Flash's address
Continuously Write data to SFC_DATA
no
Transfer finish int or wait tx fifo empty
yes
END

Fig. 21-4 slave mode write

All the AHB bus write data to SFC_CMD, SFC_ADDR and SFC_DATA will be marked with different header and then pushed into transmit FIFO by writing order.

21.6.2 AHB Slave read flash flow (non-DMA mode)

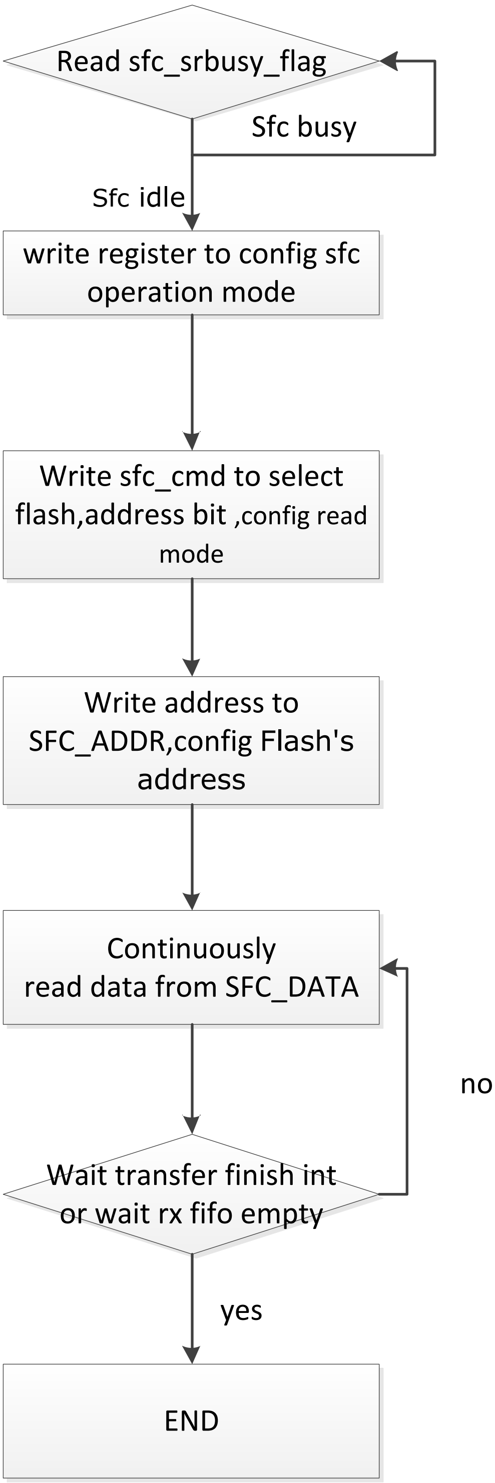

Read sfc_srbusy_flag
Sfc busy
Sfc idle
write register to config sfc operation mode
Write sfc_cmd to select flash,address bit ,config read mode
Write address to SFC_ADDR,config Flash's address
Continuously read data from SFC_DATA
no
Wait transfer finish int or wait rx fifo empty
yes
END

Fig. 21-5 slave mode read

21.6.3 AHB DMA transfer flow (DMA mode)

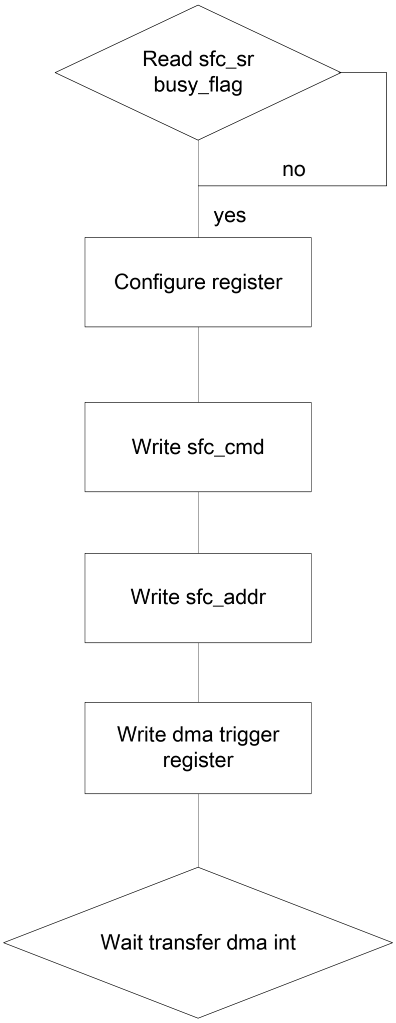


Fig. 21-6 master mode flow

The total transfer bytes is decided by TRB register in SFC_CMD and must be aligned to 2 bytes.

21.6.4 SPI Mode and Shift Phase

The register SPIM in SFC_CTRL will decide the default value of sclk_out. When SPIM=0, the default value is 0. When SPIM=1, the default value is 1. The register SHIFTPAHSE in SFC_CTRL will decide when to sample the SIO data. If SHIFTPAHSE=0, it will sample the data at the posedge of sclk_out. If SHIFTPHASE=1, it will sample the data at the negedge of sclk_out.

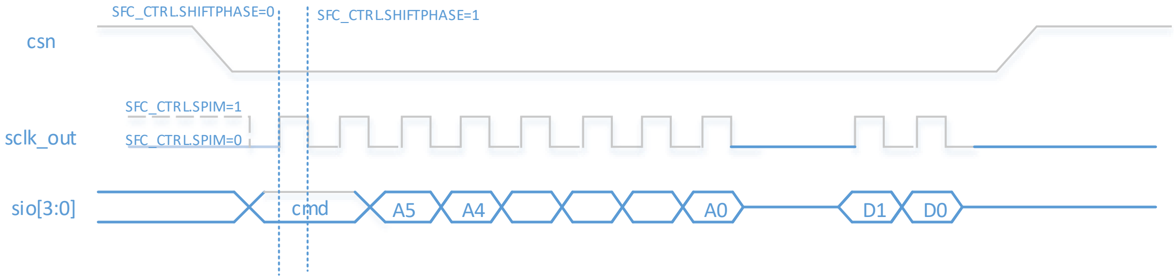


Fig. 21-7 SPI mode

21.6.5 Other Notes

The SFC core clock(SFC_sclk)need to be kept under 100MHZ and the SFC interface clock(sfc_clk) is a half of the SFC_sclk. It’s better to soft reset the SFC before data transfer.

Chapter 22 GPIO

22.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is an APB slave device.GPIO controls the output data and direction of external I/O pads. It also can read back thedata on external pads using memory-mapped registers.
GPIO supports the following features:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode

22.2 Block Diagram

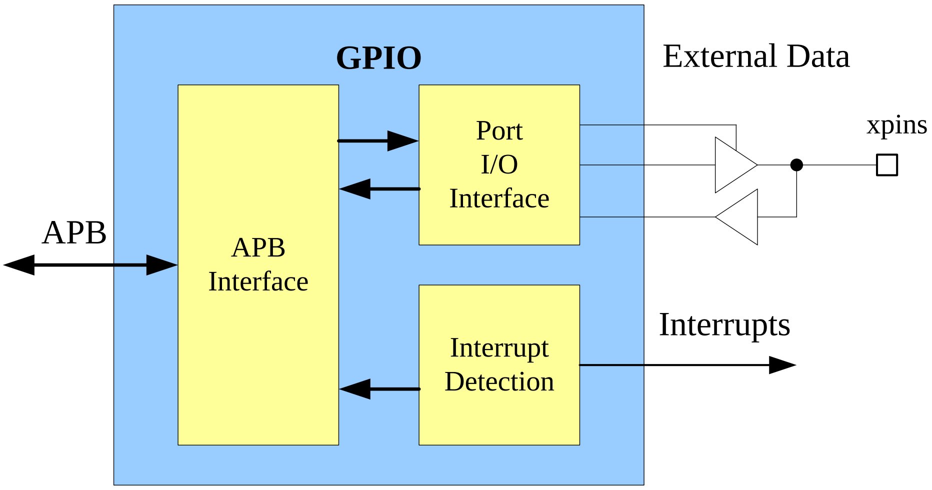


Fig. 22-1GPIO block diagram

Block descriptions:

APB Interface
The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

Port I/O Interface
External data Interface to or from I/O pads.

Interrupt Detection
Interrupt interface to or from interrupt controller.

22.3 Function Description

22.3.1 Operation

Control Mode (software)
Under software control, the data and direction control for the signal aresourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR). The direction of the external I/O pad is controlled by a write to the Porta datadirection register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mapped onto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls thedirection of an external I/O pad.
The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad.External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register(GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only,meaning that it cannot be written from the APB

software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.

An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge
- Both the rising edge and the falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost.However, no new interrupts are generated.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

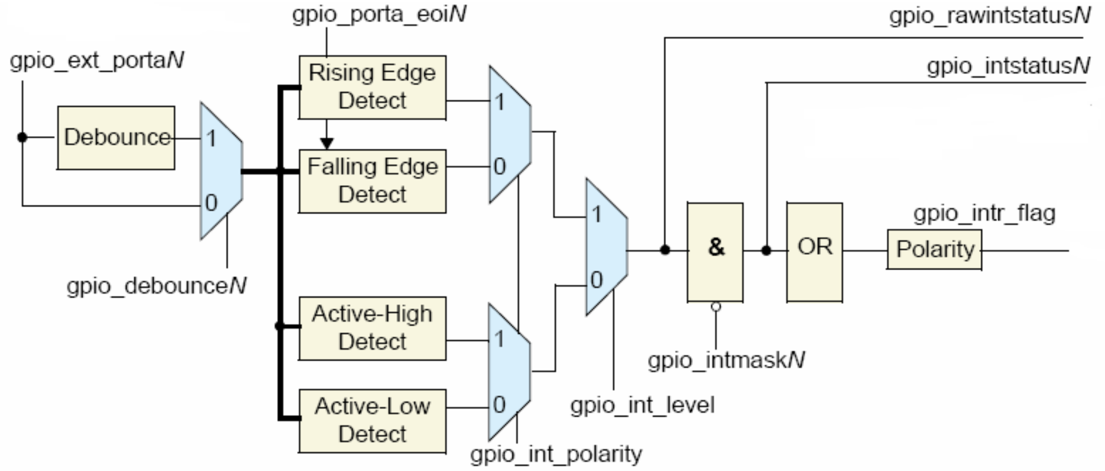


Fig. 22-2 GPIO Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When input interrupt signals are debounced using a debounce clock (pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one risingedge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization topclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional andunder software control (GPIO_LS_SYNC).

22.3.2 Programming

Programming Considerations

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitiveinterrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A inorder to prevent spurious glitches on the interrupt lines to the interrupt controller.
- Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on alevel-sensitive interrupt.

GPIOs' hierarchy in the chip

GPIO0/GPIO1/GPIO2/GPIO3/GPIO4 are all in PD_LOGIC subsystem.

22.4 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 5 GPIOs (GPIO0 ~ GPIO4), and each of them has same register group. Therefore, 5 GPIOs' register groups have 5 different base addresses.

22.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003c	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004c	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register
GPIO_INT_BOTHEDGE	0x0068	W	0x00000000	Interrupt both edge type

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

22.4.2 Detail Register Description

GPIO_SWPORTA_DR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_dr Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register

GPIO_SWPORTA_DDR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_swporta_ddr Values written to this register independently control the direction of the corresponding data bit in Port A. 1'b0: Input (default) 1'b1: Output

GPIO_INTEN

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_en Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 1'b0: Configure Port A bit as normal GPIO signal (default) 1'b1: Configure Port A bit as interrupt

GPIO_INTMASK

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_int_mask Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 1'b0: Interrupt bits are unmasked (default) 1'b1: Mask interrupt

GPIO_INTTYPE_LEVEL

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	gpio_inttype_level Controls the type of interrupt that can occur on Port A. 1'b0: Level-sensitive (default) 1'b1: Edge-sensitive

GPIO_INT_POLARITY

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_int_polarity Controls the polarity of edge or level sensitivity that can occur on input of Port A. 1'b0: Active-low (default) 1'b1: Active-high

GPIO_INT_STATUS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_status Interrupt status of Port A

GPIO_INT_RAWSTATUS

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_debounce Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 1'b0: No debounce (default) 1'b1: Enable debounce

GPIO_PORTA_EOI

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	gpio_porta_eoi Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 1'b0: No interrupt clear (default) 1'b1: Clear interrupt

GPIO_EXT_PORTA

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_ext_porta When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A

GPIO_LS_SYNC

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	gpio_ls_sync Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 1'b0: No synchronization to pclk_intr (default) 1'b1: Synchronize to pclk_intr

GPIO_INT_BOTHEDGE

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	interrupt_both_edge_type Controls the edge type of interrupt that can occur on Port A. Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port A. The values programmed in the registers gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity registers

22.5 Interface Description

Table 22-1 GPIO interface description

Module Pin	Dir	Pin Name	IOMUX Setting
GPIO0 Interface			
gpio0_porta[7:0]	I/O	GPIO0_A[7:0]	GRF_GPIO0A_IOMUX[15:0]=16'h0
gpio0_porta[15:8]	I/O	GPIO0_B[7:0]	GRF_GPIO0B_IOMUX[15:0]=16'h0
gpio0_porta[21:16]	I/O	GPIO0_C[5:0]	GRF_GPIO0C_IOMUX[11:0]=12'h0
GPIO1 Interface			
gpio1_porta[7:0]	I/O	GPIO1_A[7:0]	GRF_GPIO1A_IOMUX[15:0]=16'h0
gpio1_porta[13:8]	I/O	GPIO1_B[5:0]	GRF_GPIO1B_IOMUX_L[11:0]=12'h0
gpio1_porta[14]	I/O	GPIO1_B[6]	GRF_GPIO1B_IOMUX_L[15:12]=4'h0
gpio1_porta[15]	I/O	GPIO1_B[7]	GRF_GPIO1B_IOMUX_H[3:0]=4'h0
gpio1_porta[17:16]	I/O	GPIO1_C[1:0]	GRF_GPIO1C_IOMUX_L[3:0]=4'h0
gpio1_porta[20:18]	I/O	GPIO1_C[4:2]	GRF_GPIO1C_IOMUX_L[15:4]=12'h0
gpio1_porta[23:21]	I/O	GPIO1_C[7:5]	GRF_GPIO1C_IOMUX_H[11:0]=12'h0
gpio1_porta[25:24]	I/O	GPIO1_D[1:0]	GRF_GPIO1D_IOMUX[3:0]=4'h0
GPIO2 Interface			
gpio2_porta[7:0]	I/O	GPIO2_A[7:0]	GRF_GPIO2A_IOMUX[15:0]=16'h0
gpio2_porta[15:8]	I/O	GPIO2_B[7:0]	GRF_GPIO2B_IOMUX[15:0]=16'h0
gpio2_porta[16]	I/O	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b0
GPIO3 Interface			
gpio3_porta[7:0]	I/O	GPIO3_A[7:0]	GRF_GPIO3A_IOMUX[15:0]=16'h0
gpio3_porta[11:8]	I/O	GPIO3_B[3:0]	GRF_GPIO3B_IOMUX[7:0]=8'h0
gpio3_porta[13:12]	I/O	GPIO3_B[5:4]	GRF_GPIO3B_IOMUX[15:8]=8'h0
GPIO4 Interface			
gpio4_porta[7:0]	I/O	GPIO4_A[7:0]	GRF_GPIO4A_IOMUX[15:0]=16'h0
gpio4_porta[15:8]	I/O	GPIO4_B[7:0]	GRF_GPIO4B_IOMUX[15:0]=16'h0
gpio4_porta[16]	I/O	GPIO4_C[0]	GRF_GPIO4C_IOMUX[1:0]=2'b0
gpio4_porta[30:24]	I/O	GPIO4_D[6:0]	GRF_GPIO4D_IOMUX[13:0]=14'h0

Note: Unused Module Pin is tied to zero!

22.6 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 23 MAC Ethernet Interface

23.1 Overview

The MAC Ethernet Controller provides a complete Ethernet interface from processor to a Reduced Media Independent Interface (RMII) compliant Ethernet PHY.
The MAC includes a DMA controller. The DMA controller efficiently moves packet data from microprocessor's RAM, formats the data for an IEEE 802.3-2002 compliant packet and transmits the data to an Ethernet Physical Interface (PHY). It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

23.1.1 Feature

- Supports 10/100-Mbps data transfer rates with the RMII interfaces
- Supports both full-duplex and half-duplex operation
  - Supports CSMA/CD Protocol for half-duplex operation
  - Supports IEEE 802.3x flow control for full-duplex operation
  - Optional forwarding of received pause control frames to the user application in full-duplex operation
  - Back-pressure support for half-duplex operation
  - Automatic transmission of zero-quanta pause frame on de-assertion of flow control input in full-duplex operation
- Preamble and start-of-frame data (SFD) insertion in Transmit, and deletion in Receive paths
- Automatic CRC and pad generation controllable on a per-frame basis
- Options for Automatic Pad/CRC Stripping on receive frames
- Programmable frame length to support Standard Ethernet frames
- Programmable InterFrameGap (40-96 bit times in steps of 8)
- Supports a variety of flexible address filtering modes:
  - 64-bit Hash filter (optional) for multicast and uni-cast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report
- Separate 32-bit status returned for transmission and reception packets
- Supports IEEE 802.1Q VLAN tag detection for reception frames
- MDIO Master interface for PHY device configuration and management
- Support detection of LAN wake-up frames and AMD Magic Packet frames
- Support checksum off-load for received IPv4 and TCP packets encapsulated by the Ethernet frame
- Support checking IPv4 header checksum and TCP, UDP, or ICMP checksum encapsulated in IPv4 or IPv6 datagrams
- Comprehensive status reporting for normal operation and transfers with errors
- Support per-frame Transmit/Receive complete interrupt control
- Supports 4-KB receive FIFO depths on reception.
- Supports 2-KB FIFO depth on transmission
- Automatic generation of PAUSE frame control or backpressure signal to the MAC core based on Receive FIFO-fill (threshold configurable) level
- Handles automatic retransmission of Collision frames for transmission
- Discards frames on late collision, excessive collisions, excessive deferral and underrun conditions
- AXI interface to any CPU or memory
- Software can select the type of AXI burst (fixed and variable length burst) in the AXI Master interface
- Supports internal loopback on the RMII for debugging
- Debug status register that gives status of FSMs in Transmit and Receive data-paths and

FIFO fill-levels.

23.2 Block Diagram

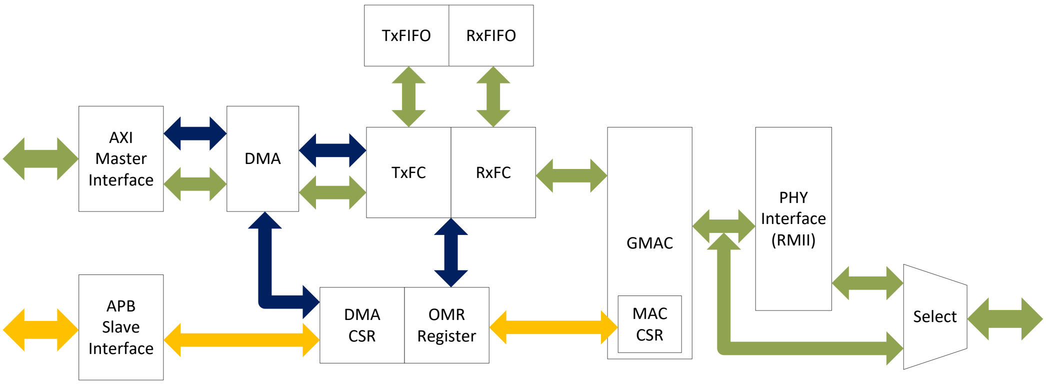


Fig. 23-1 MAC Architecture

The MAC is broken up into multiple separate functional units. These blocks are interconnected in the MAC module. The block diagram shows the general flow of data and control signals between these blocks.

The MAC transfers data to system memory through the AXI master interface. The host CPU uses the APB Slave interface to access the MAC subsystem’s control and status registers (CSRs).

The MAC supports the PHY interfaces of reduced MII (RMII).

The Transmit FIFO (Tx FIFO) buffers data read from system memory by the DMA before transmission by the MAC Core. Similarly, the Receive FIFO (Rx FIFO) stores the Ethernet frames received from the line until they are transferred to system memory by the DMA. These are asynchronous FIFOs, as they also transfer the data between the application clock and the MAC line clocks.

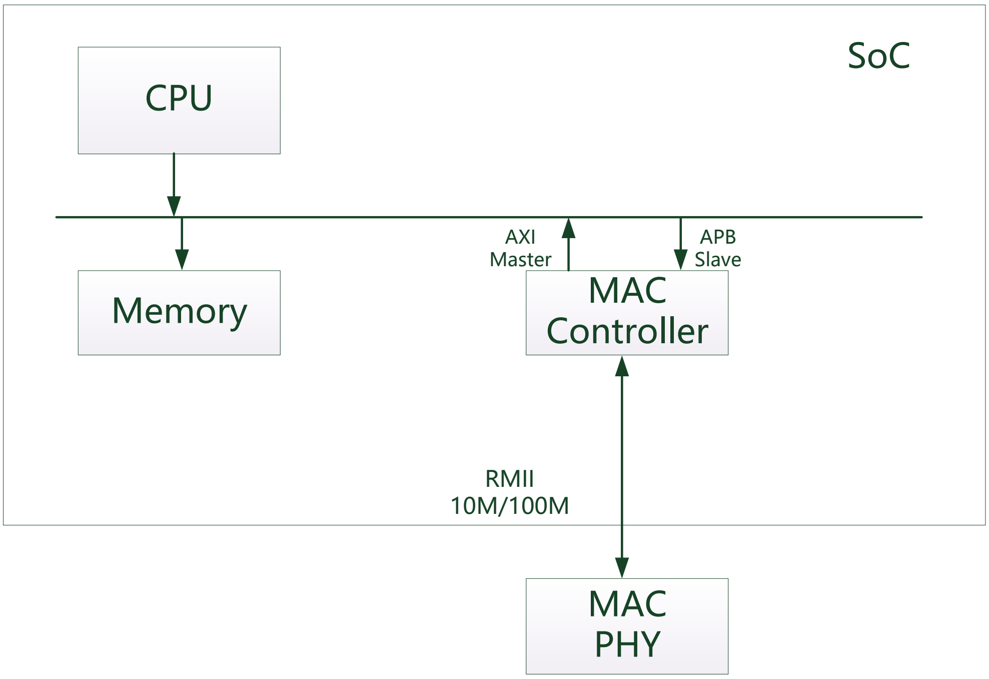


Fig. 23-2 MAC Block Diagram

The MAC controller named MAC2IO:

- MAC2IO Supports 10/100-Mbps data transfer rates with the RMII interfaces

23.3 Function Description

23.3.1 Frame Structure

Data frames transmitted shall have the frame format shown in Fig. 1-3.

<inter-frame><preamble><sfd><data><efd>

Fig. 23-3 MAC Frame Structure

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consists of 7 octets with the following bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble. The bit value is 10101011.
The data in a well formed frame shall consist of N octet's data.

23.3.2 RMII Interface timing diagram

The Reduced Media Independent Interface (RMII) specification reduces the pin count between Ethernet PHYs and Switch ASICs (only in 10/100 mode). According to the IEEE 802.3u standard, an MII contains 16 pins for data and control. In devices incorporating multiple MAC or PHY interfaces (such as switches), the number of pins adds significant cost with increase in port count. The RMII specification addresses this problem by reducing the pin count to 7 for each port - a 62.5% decrease in pin count.
The RMII module is instantiated between the MAC and the PHY. This helps translation of the MAC's MII into the RMII. The RMII block has the following characteristics:

- Supports 10-Mbps and 100-Mbps operating rates. It does not support 1000-Mbps operation.
- Two clock references are sourced externally or CRU, providing independent, 2-bit wide transmit and receive paths.

Transmit Bit Ordering
Each nibble from the MII must be transmitted on the RMII a di-bit at a time with the order of di-bit transmission shown in Fig.1-4. The lower order bits (D1 and D0) are transmitted first followed by higher order bits (D2 and D3).

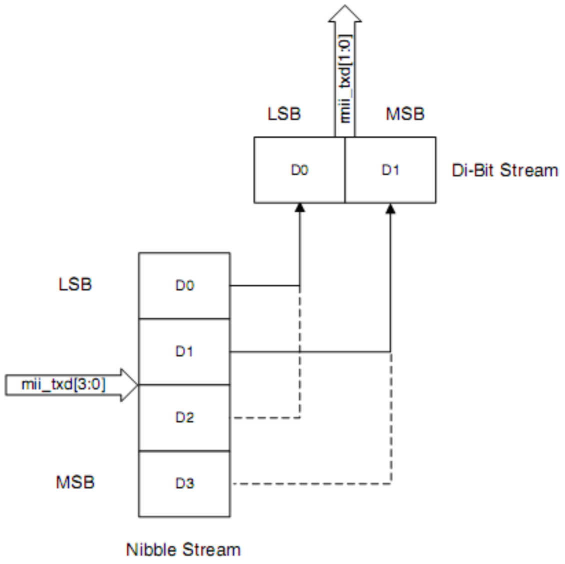


Fig. 23-4 RMII transmission bit ordering

RMII Transmit Timing Diagrams
Fig.1-5 through 1-8 show MII-to-RMII transaction timing. The clk_rmii_i (REF_CLK) frequency is 50MHz in RMII interface. In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on rmii_txd_o[1:0] (TXD[1:0]) shall be valid such that

TXD[1:0] may be sampled every 10th cycle, regard-less of the starting cycle within the group and yield the correct frame data.

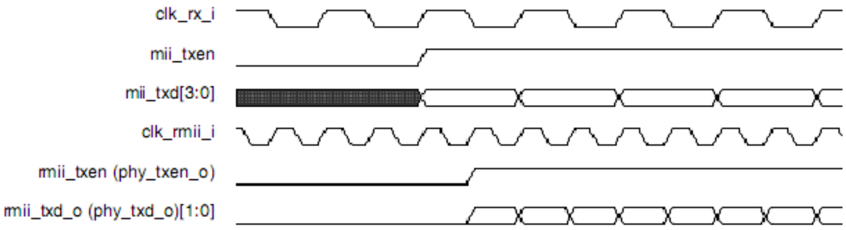


Fig. 23-5 Start of MII and RMII transmission in 100-Mbps mode

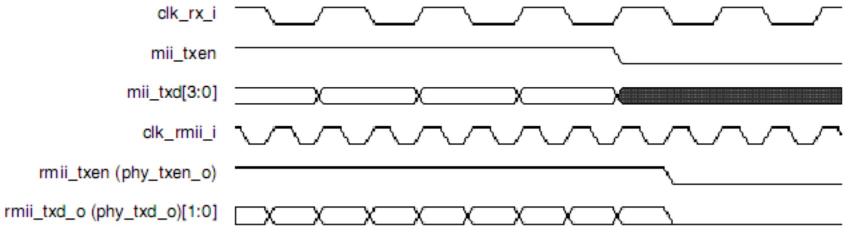


Fig. 23-6 End of MII and RMII Transmission in 100-Mbps Mode

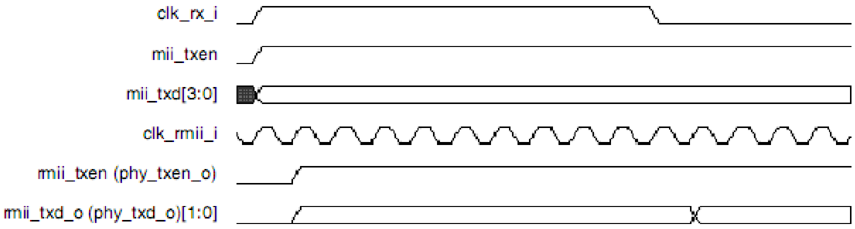


Fig. 23-7 Start of MII and RMII Transmission in 10-Mbps Mode

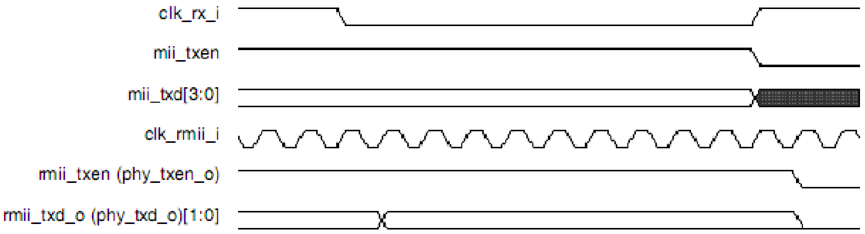


Fig. 23-8 End of MII and RMII Transmission in 10-Mbps Mode

Receive Bit Ordering

Each nibble is transmitted to the MII from the di-bit received from the RMII in the nibble transmission order shown in Fig.1-9. The lower order bits (D0 and D1) are received first, followed by the higher order bits (D2 and D3).

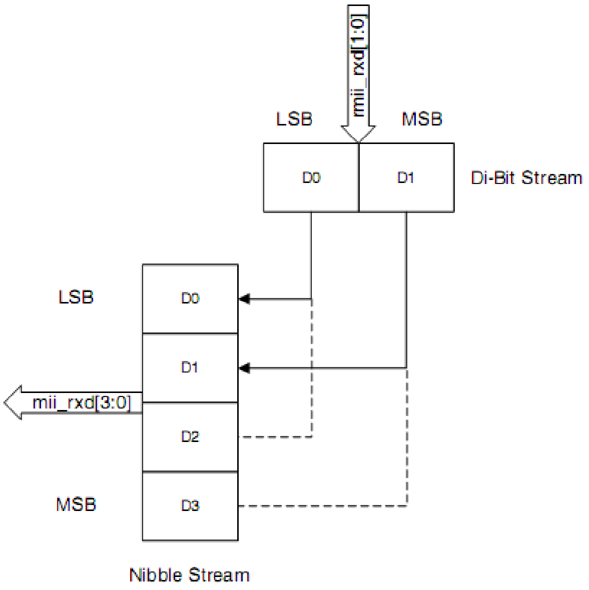


Fig. 23-9 RMII receive bit ordering

23.3.3 Management Interface

The MAC management interface provides a simple, two-wire, serial interface to connect the MAC and a managed PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information across the MII bus: MDIO and MDC.

The MAC initiates the management write/read operation. The clock gmii_mdc_o(MDC) is a divided clock from the application clock pclk_MAC. The divide factor depends on the clock range setting in the GMII address register. Clock range is set as follows:

Selection	**pclk_MAC**	**MDC Clock**
0000	60-100 MHz	pclk_MAC/42
0001	100-150 MHz	pclk_MAC/62
0010	20-35 MHz	pclk_MAC/16
0011	35-60 MHz	pclk_MAC/26
0100	150-250 MHz	pclk_MAC/102
0101	250-300 MHz	pclk_MAC/124
0110, 0111	Reserved	

The MDC is the derivative of the application clock pclk_MAC. The management operation is performed through the gmii_mdi_i, gmii_mdo_o and gmii_mdo_o_e signals. A three-state buffer is implemented in the PAD.

The frame structure on the MDIO line is shown below.

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE

Fig. 23-10 MDIO frame structure

IDLE: The mdio line is three-state; there is no clock on gmii_mdc_o

PREAMBLE: 32 continuous bits of value 1

START: Start-of-frame is 2’b01

OPCODE: 2’b10 for read and 2’b01 for write

PHY ADDR: 5-bit address select for one of 32 PHYs

REG ADDR: Register address in the selected PHY

TA: Turnaround is 2’bZ0 for read and 2’b10 for Write

DATA: Any 16-bit value. In a write operation, the MAC drives mdio; in a read operation, PHY drives it.

23.3.4 Power Management Block

Power management (PMT) supports the reception of network (remote) wake-up frames and Magic Packet frames. PMT does not perform the clock gate function, but generates interrupts for wake-up frames and Magic Packets received by the MAC. The PMT block sits on the receiver path of the MAC and is enabled with remote wake-up frame enable and Magic Packet enable. These enables are in the PMT control and status register and are programmed by the application.

When the power down mode is enabled in the PMT, then all received frames are dropped by the core and they are not forwarded to the application. The core comes out of the power down mode only when either a Magic Packet or a Remote Wake-up frame is received and the corresponding detection is enabled.

Remote Wake-Up Frame Detection

When the MAC is in sleep mode and the remote wake-up bit is enabled in register MAC_PMT_CTRL_STA (0x002C), normal operation is resumed after receiving a remote wake-up frame. The application writes all eight wake-up filter registers, by performing a sequential write to address (0028). The application enables remote wake-up by writing a 1 to bit 2 of the register MAC_PMT_CTRL_STA.

PMT supports four programmable filters that allow support of different receive frame patterns. If the incoming frame passes the address filtering of Filter Command, and if Filter CRC-16 matches the incoming examined pattern, then the wake-up frame is received. Filter_offset (minimum value 12, which refers to the 13th byte of the frame) determines the offset from which the frame is to be examined. Filter Byte Mask determines which bytes of the frame must be examined. The thirty-first bit of Byte Mask must be set to zero.

The remote wake-up CRC block determines the CRC value that is compared with Filter CRC-16. The wake-up frame is checked only for length error, FCS error, dribble bit error, GMII error, collision, and to ensure that it is not a runt frame. Even if the wake-up frame is more than 512 bytes long, if the frame has a valid CRC value, it is considered valid. Wake-up frame detection is updated in the register MAC_PMT_CTRL_STA for every remote Wake-up frame received. A PMT interrupt to the application triggers a read to the MAC_PMT_CTRL_STA register to determine reception of a wake-up frame.

Magic Packet Detection

The Magic Packet frame is based on a method that uses Advanced Micro Device’s Magic Packet technology to power up the sleeping device on the network. The MAC receives a specific packet of information, called a Magic Packet, addressed to the node on the network. Only Magic Packets that are addressed to the device or a broadcast address will be checked to determine whether they meet the wake-up requirements. Magic Packets that pass the address filtering (unicast or broadcast) will be checked to determine whether they meet the remote Wake-on-LAN data format of 6 bytes of all ones followed by a MAC Address appearing 16 times.

The application enables Magic Packet wake-up by writing a 1 to Bit 1 of the register MAC_PMT_CTRL_STA. The PMT block constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Each frame received is checked for a 48’hFF_FF_FF_FF_FF_FF pattern following the destination and source address field. The PMT block then checks the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 repetitions of the address, the 48’hFF_FF_FF_FF_FF_FF pattern is scanned for again in the incoming frame. The 16 repetitions can be anywhere in the frame, but must be preceded by the synchronization stream (48’hFF_FF_FF_FF_FF_FF). The device will also accept a multicast frame, as long as the 16 duplications of the MAC address are detected.

If the MAC address of a node is 48'h00_11_22_33_44_55, then the MAC scans for the data sequence:

Destination Address Source Address ………………………………. FF FFFFFFFFFF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC
Magic Packet detection is updated in the PMT Control and Status register for Magic Packet received. A PMT interrupt to the Application triggers a read to the PMT CSR to determine whether a Magic Packet frame has been received.

23.3.5 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. These include a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the Application through the MAC Control Interface (MCI). Non-32-bit accesses are allowed as long as the address is word-aligned.

The organization of these registers is shown in Register Description. The MMCs are accessed using transactions, in the same way the CSR address space is accessed. The Register Description in this chapter describe the various counters and list the address for each of the statistics counters. This address will be used for Read/Write accesses to the desired transmit/receive counter.

The MMC module gathers statistics on encapsulated IPv4, IPv6, TCP, UDP, or ICMP payloads in received Ethernet frames.

23.4 Register Description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MAC_MAC_CONF	0x0000	W	0x00000000	MAC Configuration Register This is the operation mode register for the MAC
MAC_MAC_FRM_FILT	0x0004	W	0x00000000	MAC Frame Filter Contains the frame filtering controls
MAC_HASH_TAB_HI	0x0008	W	0x00000000	Hash Table High Register Contains the higher 32 bits of the Multicast Hash table. This register is present only when the Hash filter function is selected in core Consultant
MAC_HASH_TAB_LO	0x000c	W	0x00000000	Hash Table Low Register Contains the lower 32 bits of the Multicast Hash table. This register is present only when the Hash filter function is selected in core Consultant

Name	Offset	Size	Reset Value	Description
MAC_GMII_ADDR	0x0010	W	0x00000000	GMII Address Register Controls the management cycles to an external PHY
MAC_GMII_DATA	0x0014	W	0x00000000	GMII Data Register Contains the data to be written to or read from the PHY register
MAC_FLOW_CTRL	0x0018	W	0x00000000	Flow Control Register Controls the generation of control frames
MAC_VLAN_TAG	0x001c	W	0x00000000	VLAN Tag Register Identifies IEEE 802.1Q VLAN type frames
MAC_DEBUG	0x0024	W	0x00000000	Debug register This debug register gives the status of all the main modules of the transmit and receive data-paths and the FIFOs. An all-zero status indicates that the MAC core is in idle state (and FIFOs are empty) and no activity is going on in the data-paths
MAC_PMT_CTRL_STA	0x002c	W	0x00000000	PMT Control and Status Register PMT Control and Status
MAC_INT_STATUS	0x0038	W	0x00000000	Interrupt Status Register Contains the interrupt status
MAC_INT_MASK	0x003c	W	0x00000000	Interrupt Mask Register Contains the masks for generating the interrupts
MAC_MAC_ADDR0_HI	0x0040	W	0x0000ffff	MAC Address0 High Register Contains the higher 16 bits of the first MAC address
MAC_MAC_ADDR0_LO	0x0044	W	0xffffffff	MAC Address0 Low Register Contains the lower 32 bits of the first MAC address
MAC_AN_CTRL	0x00c0	W	0x00000000	AN Control Register Enables and/or restarts auto-negotiation. It also enables PCS loopback
MAC_AN_STATUS	0x00c4	W	0x00000008	AN Status Register Indicates the link and auto-negotiation status

Name	Offset	Size	Reset Value	Description
MAC_AN_ADV	0x00c8	W	0x000001e0	Auto Negotiation Advertisement Register This register is configured before auto-negotiation begins. It contains the advertised ability of the MAC
MAC_AN_LINK_PART_AB	0x00cc	W	0x00000000	Auto Negotiation Link Partner Ability Register Contains the advertised ability of the link partner. Its value is valid after successful completion of auto-negotiation or when a new base page has been received (indicated in the Auto-Negotiation Expansion Register)
MAC_AN_EXP	0x00d0	W	0x00000000	Auto Negotiation Expansion Register Indicates whether a new base page has been received from the link partner
MAC_INTF_MODE_STA	0x00d8	W	0x00000000	RGMII Status Register Indicates the status signals received from the PHY through the RGMII interface
MAC_MMC_CTRL	0x0100	W	0x00000000	MMC Control Register The MMC Control register establishes the operating mode of the management counters

Name	Offset	Size	Reset Value	Description
MAC_MMC_RX_INTR	0x0104	W	0x00000000	MMC Receive Interrupt Register The MMC Receive Interrupt register maintains the interrupts generated when the receive statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Receive Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit
MAC_MMC_TX_INTR	0x0108	W	0x00000000	MMC Transmit Interrupt Register The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. The MMC Transmit Interrupt register is a 32-bit wide register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits[7:0]) of the respective counter must be read in order to clear the interrupt bit

Name	Offset	Size	Reset Value	Description
MAC_MMC_RX_INT_MSK	0x010c	W	0x00000000	MMC Receive Interrupt Mask Register The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half their maximum value, and when they reach their maximum values
MAC_MMC_TX_INT_MSK	0x0110	W	0x00000000	MMC Transmit Interrupt Mask Register The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when transmit statistic counters reach half their maximum value, and when they reach their maximum values
MAC_MMC_TXOCTETCNT_GB	0x0114	W	0x00000000	MMC TX OCTET Good and Bad Counter
MAC_MMC_TXFRMCNT_GB	0x0118	W	0x00000000	MMC TX OCTET Good and Bad Counter
MAC_MMC_TXUNDFLWERR	0x0148	W	0x00000000	MMC TX Underflow Error
MAC_MMC_TXCARERR	0x0160	W	0x00000000	MMC TX Carrier Error
MAC_MMC_TXOCTETCNT_G	0x0164	W	0x00000000	MMC TX OCTET Good Counter
MAC_MMC_TXFRMCNT_G	0x0168	W	0x00000000	MMC TX Frame Good Counter
MAC_MMC_RXFRMCNT_GB	0x0180	W	0x00000000	MMC RX Frame Good and Bad Counter
MAC_MMC_RXOCTETCNT_GB	0x0184	W	0x00000000	MMC RX OCTET Good and Bad Counter
MAC_MMC_RXOCTETCNT_G	0x0188	W	0x00000000	MMC RX OCTET Good Counter
MAC_MMC_RXMCFRMCNT_G	0x0190	W	0x00000000	MMC RX Multicast Frame Good Counter
MAC_MMC_RXCRCERR	0x0194	W	0x00000000	MMC RX Carrier
MAC_MMC_RXLENERR	0x01c8	W	0x00000000	MMC RX Length Error
MAC_MMC_RXFIFOOVRFLW	0x01d4	W	0x00000000	MMC RX FIFO Overflow

Name	Offset	Size	Reset Value	Description
MAC_MMC_IPC_INT_MSK	0x0200	W	0x00000000	MMC Receive Checksum Offload Interrupt Mask Register The MMC Receive Checksum Offload Interrupt Mask register maintains the masks for the interrupts generated when the receive IPC (Checksum Offload) statistic counters reach half their maximum value , and when they reach their maximum values
MAC_MMC_IPC_INTR	0x0208	W	0x00000000	MMC Receive Checksum Offload Interrupt Register The MMC Receive Checksum Offload Interrupt register maintains the interrupts generated when receive IPC statistic counters reach half their maximum values (0x8000_0000), and when they cross their maximum values (0xFFFF_FFFF). When Counter Stop Rollover is set, then interrupts are set but the counter remains at all-ones. When the MMC IPC counter that caused the interrupt is read, its corresponding interrupt bit is cleared. The counterís least-significant byte lane (bits[7:0]) must be read to clear the interrupt bit
MAC_MMC_RXIPV4GFRM	0x0210	W	0x00000000	MMC RX IPV4 Good Frame
MAC_MMC_RXIPV4HDERRFRM	0x0214	W	0x00000000	MMC RX IPV4 Head Error Frame
MAC_MMC_RXIPV6GFRM	0x0224	W	0x00000000	MMC RX IPV6 Good Frame
MAC_MMC_RXIPV6HDERRFRM	0x0228	W	0x00000000	MMC RX IPV6 Head Error Frame
MAC_MMC_RXUDPERRFRM	0x0234	W	0x00000000	MMC RX UDP Error Frame
MAC_MMC_RXTCPERRFRM	0x023c	W	0x00000000	MMC RX TCP Error Frame
MAC_MMC_RXICMPERRFRM	0x0244	W	0x00000000	MMC RX ICMP Error Frame
MAC_MMC_RXIPV4HDERROCT	0x0254	W	0x00000000	MMC RX OCTET IPV4 Head Error

Name	Offset	Size	Reset Value	Description
MAC_MMC_RXIPV6HDERROCT	0x0268	W	0x00000000	MMC RX OCTET IPV6 Head Error
MAC_MMC_RXUDPERROCT	0x0274	W	0x00000000	MMC RX OCTET UDP Error
MAC_MMC_RXTCPERROCT	0x027c	W	0x00000000	MMC RX OCTET TCP Error
MAC_MMC_RXICMPERROCT	0x0284	W	0x00000000	MMC RX OCTET ICMP Error
MAC_BUS_MODE	0x1000	W	0x00020101	Bus Mode Register
MAC_TX_POLL_DEMAND	0x1004	W	0x00000000	Transmit Poll Demand Register Used by the host to instruct the DMA to poll the Transmit Descriptor List
MAC_RX_POLL_DEMAND	0x1008	W	0x00000000	Receive Poll Demand Register Used by the Host to instruct the DMA to poll the Receive Descriptor list
MAC_RX_DESC_LIST_ADDR	0x100c	W	0x00000000	Receive Descriptor List Address Register Points the DMA to the start of the Receive Descriptor list
MAC_TX_DESC_LIST_ADDR	0x1010	W	0x00000000	Transmit Descriptor List Address Register Points the DMA to the start of the Transmit Descriptor List
MAC_STATUS	0x1014	W	0x00000000	Status Register The Software driver (application) reads this register during interrupt service routine or polling to determine the status of the DMA
MAC_OP_MODE	0x1018	W	0x00000000	Operation Mode Register Establishes the Receive and Transmit operating modes and command
MAC_INT_ENA	0x101c	W	0x00000000	Interrupt Enable Register Enables the interrupts reported by the Status Register
MAC_OVERFLOW_CNT	0x1020	W	0x00000000	Missed Frame and Buffer Overflow Counter Register Contains the counters for discarded frames because no host Receive Descriptor was available, and discarded frames because of Receive FIFO Overflow

Name	Offset	Size	Reset Value	Description
MAC_REC_INT_WDT_TIMER	0x1024	W	0x00000000	Receive Interrupt Watchdog Timer Register Watchdog time-out for Receive Interrupt (RI) from DMA
MAC_AXI_BUS_MODE	0x1028	W	0x00110001	AXI Bus Mode Register Controls AXI Master behavior (mainly controls burst splitting and number of outstanding requests)
MAC_AXI_STATUS	0x102c	W	0x00000000	AXI Status Register Gives the idle status of the AXI master's read/write channels
MAC_CUR_HOST_TX_DESC	0x1048	W	0x00000000	Current Host Transmit Descriptor Register Points to the start of current Transmit Descriptor read by the DMA
MAC_CUR_HOST_RX_DESC	0x104c	W	0x00000000	Current Host Receive Descriptor Register Points to the start of current Receive Descriptor read by the DMA
MAC_CUR_HOST_TX_BUF_ADDR	0x1050	W	0x00000000	Current Host Transmit Buffer Address Register Points to the current Transmit Buffer address read by the DMA
MAC_CUR_HOST_RX_BUF_ADDR	0x1054	W	0x00000000	Current Host Receive Buffer Address Register Points to the current Receive Buffer address read by the DMA

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

23.4.2 Detail Register Description

MAC_MAC_CONF

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	TC Transmit Configuration in RGMII When set, this bit enables the transmission of duplex mode, link speed, and link up/down information to the PHY in the RGMII ports. When this bit is reset, no such information is driven to the PHY

Bit	Attr	Reset Value	Description
23	RW	0x0	WD Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver, and can receive frames of up to 16,384 bytes. When this bit is reset, the MAC allows no more than 2,048 bytes (10,240 if JE is set high) of the frame being received and cuts off any bytes received after that
22	RW	0x0	JD Jabber Disable When this bit is set, the MAC disables the jabber timer on the transmitter, and can transfer frames of up to 16,384 bytes. When this bit is reset, the MAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission
21	RW	0x0	BE Frame Burst Enable When this bit is set, the MAC allows frame bursting during transmission in GMII Half-Duplex mode
20	RO	0x0	reserved
19:17	RW	0x0	IFG Inter-Frame Gap These bits control the minimum IFG between frames during transmission. 3'b000: 96 bit times 3'b001: 88 bit times 3'b010: 80 bit times ... 3'b111: 40 bit times
16	RW	0x0	DCRS Disable Carrier Sense During Transmission When set high, this bit makes the MAC transmitter ignore the (G)MII CRS signal during frame transmission in Half-Duplex mode. This request results in no errors generated due to Loss of Carrier or No Carrier during such transmission. When this bit is low, the MAC transmitter generates such errors due to Carrier Sense and will even abort the transmissions
15	RW	0x0	PS Port Select Selects between GMII and MII: 1'b0: GMII (1000 Mbps) 1'b1: MII (10/100 Mbps)

Bit	Attr	Reset Value	Description
14	RW	0x0	FES Speed Indicates the speed in Fast Ethernet (MII) mode: 1'b0: 10 Mbps 1'b1: 100 Mbps
13	RW	0x0	DO Disable Receive Own When this bit is set, the MAC disables the reception of frames when the gmii_txen_o is asserted in Half-Duplex mode. When this bit is reset, the MAC receives all packets that are given by the PHY while transmitting
12	RW	0x0	LM Loopback Mode When this bit is set, the MAC operates in loopback mode at GMII/MII. The (G)MII Receive clock input (clk_rx_i) is required for the loopback to work properly, as the Transmit clock is not looped-back internally
11	RW	0x0	DM Duplex Mode When this bit is set, the MAC operates in a Full-Duplex mode where it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in Full-Duplex-only configuration
10	RW	0x0	IPC Checksum Offload When this bit is set, the MAC calculates the 16-bit one's complement of the one's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. The MAC core also appends the 16-bit checksum calculated for the IP header datagram payload (bytes after the IPv4 header) and appends it to the Ethernet frame transferred to the application (when Type 2 COE is deselected). When this bit is reset, this function is disabled. When Type 2 COE is selected, this bit, when set, enables IPv4 checksum checking for received frame payloads TCP/UDP/ICMP headers. When this bit is reset, the COE function in the receiver is disabled and the corresponding PCE and IP HCE status bits are always cleared

Bit	Attr	Reset Value	Description
9	RW	0x0	DR Disable Retry When this bit is set, the MAC will attempt only 1 transmission. When a collision occurs on the GMII/MII, the MAC will ignore the current frame transmission and report a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the MAC will attempt retries based on the settings of BL
8	RW	0x0	LUD Link Up/Down Indicates whether the link is up or down during the transmission of configuration in RGMII interface: 1'b0: Link Down 1'b1: Link Up
7	RW	0x0	ACS Automatic Pad/CRC Stripping When this bit is set, the MAC strips the Pad/FCS field on incoming frames only if the length's field value is less than or equal to 1,500 bytes. All received frames with length field greater than or equal to 1,501 bytes are passed to the application without stripping the Pad/FCS field. When this bit is reset, the MAC will pass all incoming frames to the Host unmodified
6:5	RW	0x0	BL Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000 Mbps and 512 bit times for 10/100 Mbps) the MAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only to Half-Duplex mode and is reserved (RO) in Full-Duplex-only configuration. 2'b00: k = min (n, 10) 2'b01: k = min (n, 8) 2'b10: k = min (n, 4) 2'b11: k = min (n, 1), Where n = retransmission attempt. The random integer r takes the value in the range 0 = r < 2^k

Bit	Attr	Reset Value	Description
4	RW	0x0	DC Deferral Check When this bit is set, the deferral check function is enabled in the MAC. The MAC will issue a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status when the transmit state machine is deferred for more than 24,288 bit times in 10/100-Mbps mode. If the Core is configured for 1000 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but is prevented because of an active CRS (carrier sense) signal on the GMII/MII. Defer time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive
3	RW	0x0	TE Transmitter Enable When this bit is set, the transmit state machine of the MAC is enabled for transmission on the GMII/MII. When this bit is reset, the MAC transmit state machine is disabled after the completion of the transmission of the current frame, and will not transmit any further frames
2	RW	0x0	RE Receiver Enable When this bit is set, the receiver state machine of the MAC is enabled for receiving frames from the GMII/MII. When this bit is reset, the MAC receive state machine is disabled after the completion of the reception of the current frame, and will not receive any further frames from the GMII/MII
1:0	RO	0x0	reserved

MAC_MAC_FRM_FILT

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RW	0x0	RA Receive All When this bit is set, the MAC Receiver module passes to the Application all frames received irrespective of whether they pass the address filter. The result of the SA/DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes to the Application only those frames that pass the SA/DA address filter
30:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	HPF Hash or Perfect Filter When set, this bit configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by HMC or HUC bits. When low and if the HUC/HMC bit is set, the frame is passed only if it matches the Hash filter
9	RW	0x0	SAF Source Address Filter Enable The MAC core compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison matches, then the SAMatch bit of RxStatus Word is set high. When this bit is set high and the SA filter fails, the MAC drops the frame. When this bit is reset, then the MAC Core forwards the received frame to the application and with the updated SA Match bit of the RxStatus depending on the SA address comparison
8	RW	0x0	SAIF SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers will be marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers will be marked as failing the SA Address filter
7:6	RW	0x0	PCF Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). Note that the processing of PAUSE control frames depends only on RFE of Register MAC_FLOW_CTRL[2]. 2'b00: MAC filters all control frames from reaching the application. 2'b01: MAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 2'b10: MAC forwards all control frames to application even if they fail the Address Filter. 2'b11: MAC forwards control frames that pass the Address Filter
5	RW	0x0	DBF Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. When this bit is reset, the AFM module passes all received broadcast frames

Bit	Attr	Reset Value	Description
4	RW	0x0	PM Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is '1') are passed. When reset, filtering of multicast frame depends on HMC bit
3	RW	0x0	DAIF DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed
2	RW	0x0	HMC Hash Multicast When set, MAC performs destination address filtering of received multicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers
1	RW	0x0	HUC Hash Unicast When set, MAC performs destination address filtering of unicast frames according to the hash table. When reset, the MAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers
0	RW	0x0	PR Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA/DA Filter Fails status bits of the Receive Status Word will always be cleared when PR is set

MAC_HASH_TAB_HI

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTH Hash Table High This field contains the upper 32 bits of Hash table

MAC_HASH_TAB_LO

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	HTL Hash Table Low This field contains the lower 32 bits of Hash table

MAC_GMII_ADDR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:11	RW	0x00	PA Physical Layer Address This field tells which of the 32 possible PHY devices are being accessed
10:6	RW	0x00	GR GMII Register These bits select the desired GMII register in the selected PHY device

Bit	Attr	Reset Value	Description
5:2	RW	0x0	CR APB Clock Range The APB Clock Range selection determines the frequency of the MDC clock as per the pclk_MAC frequency used in your design. The suggested range of pclk_MAC frequency applicable for each value below (when Bit[5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz - 2.5 MHz. Selection pclk_MAC MDC Clock 0000 60-100 MHz pclk_MAC/42 0001 100-150 MHz pclk_MAC/62 0010 20-35 MHz pclk_MAC/16 0011 35-60 MHz pclk_MAC/26 0100 150-250 MHz pclk_MAC/102 0101 250-300 MHz pclk_MAC/124 0110, 0111 Reserved When bit 5 is set, you can achieve MDC clock of frequency higher than the IEEE 802.3 specified frequency limit of 2.5 MHz and program a clock divider of lower value. For example, when pclk_MAC is of frequency 100 MHz and you program these bits as "1010", then the resultant MDC clock will be of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Please program the values given below only if the interfacing chips supports faster MDC clocks. Selection MDC Clock 1000 pclk_MAC/4 1001 pclk_MAC/6 1010 pclk_MAC/8 1011 pclk_MAC/10 1100 pclk_MAC/12 1101 pclk_MAC/14 1110 pclk_MAC/16 1111 pclk_MAC/18
1	RW	0x0	GW GMII Write When set, this bit tells the PHY that this will be a Write operation using register MAC_GMII_DATA. If this bit is not set, this will be a Read operation, placing the data in register MAC_GMII_DATA

Bit	Attr	Reset Value	Description
0	W1C	0x0	GB GMII Busy This bit should read a logic 0 before writing to Register GMII_ADDR and Register GMII_DATA. This bit must also be set to 0 during a Write to Register GMII_ADDR. During a PHY register access, this bit will be set to 1'b1 by the Application to indicate that a Read or Write access is in progress. Register GMII_DATA (GMII Data) should be kept valid until this bit is cleared by the MAC during a PHY Write operation. The Register GMII_DATA is invalid until this bit is cleared by the MAC during a PHY Read operation. The Register GMII_ADDR (GMII Address) should not be written to until this bit is cleared

MAC_GMII_DATA

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GD GMII Data This contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation

MAC_FLOW_CTRL

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PT Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. If the Pause Time bits is configured to be double-synchronized to the (G)MII clock domain, then consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain
15:8	RO	0x0	reserved
7	RW	0x0	DZPQ Disable Zero-Quanta Pause When set, this bit disables the automatic generation of Zero-Quanta Pause Control frames on the de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i/mti_flowctrl_i). When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled
6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	PLT Pause Low Threshold This field configures the threshold of the PAUSE timer at which the input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of PAUSE Frame. The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot-times), and PLT = 01, then a second PAUSE frame is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot-times after the first PAUSE frame is transmitted. Selection Threshold 00 Pause time minus 4 slot times 01 Pause time minus 28 slot times 10 Pause time minus 144 slot times 11 Pause time minus 256 slot times Slot time is defined as time taken to transmit 512 bits (64 bytes) on the GMII/MII interface
3	RW	0x0	UP Unicast Pause Frame Detect When this bit is set, the MAC will detect the Pause frames with the station's unicast address specified in MAC Address0 High Register and MAC Address0 Low Register, in addition to the detecting Pause frames with the unique multicast address. When this bit is reset, the MAC will detect only a Pause frame with the unique multicast address specified in the 802.3x standard
2	RW	0x0	RFE Receive Flow Control Enable When this bit is set, the MAC will decode the received Pause frame and disable its transmitter for a specified (Pause Time) time. When this bit is reset, the decode function of the Pause frame is disabled
1	RW	0x0	TFE Transmit Flow Control Enable In Full-Duplex mode, when this bit is set, the MAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC will not transmit any Pause frames. In Half-Duplex mode, when this bit is set, the MAC enables the back-pressure operation. When this bit is reset, the backpressure feature is disabled

Bit	Attr	Reset Value	Description
0	RW	0x0	FCB_BPA Flow Control Busy/Backpressure Activate This bit initiates a Pause Control frame in Full-Duplex mode and activates the backpressure function in Half-Duplex mode if TFE bit is set. In Full-Duplex mode, this bit should be read as 1'b0 before writing to the register MAC_FLOW_CTRL. To initiate a pause control frame, the application must set this bit to 1'b1. During a transfer of the control frame, this bit will continue to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the MAC will reset this bit to 1'b0. The register MAC_FLOW_CTRL should not be written to until this bit is cleared. In Half-Duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the MAC Core. During backpressure, when the MAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically OR'ed with the mti_flowctrl_i input signal for the backpressure function

MAC_VLAN_TAG

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	ETV Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier, rather than the complete 16-bit VLAN tag, is used for comparison and filtering. Bits[11:0] of the VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. When this bit is reset, all 16 bits of the received VLAN frame's fifteenth and sixteenth bytes are used for comparison
15:0	RW	0x0000	VL VLAN Tag Identifier for Receive Frames This contains the 802.1Q VLAN tag to identify VLAN frames, and is compared to the fifteenth and sixteenth bytes of the frames being received for VLAN frames. Bits[15:13] are the User Priority, Bit[12] is the Canonical Format Indicator (CFI) and bits[11:0] are the VLAN tag's VLAN Identifier (VID) field. When the ETV bit is set, only the VID (Bits[11:0]) is used for comparison. If VL (VL[11:0] if ETV is set) is all zeros, the MAC does not check the fifteenth and sixteenth bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 to be VLAN frames

MAC_DEBUG

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	TFIFO3 When high, it indicates that the MTL TxStatus FIFO is full and hence the MTL will not be accepting any more frames for transmission
24	RW	0x0	TFIFO2 When high, it indicates that the MTL TxFIFO is not empty and has some data left for transmission
23	RO	0x0	reserved
22	RW	0x0	TFIFO1 When high, it indicates that the MTL TxFIFO Write Controller is active and transferring data to the TxFIFO
21:20	RW	0x0	TFIFOSTA This indicates the state of the TxFIFO read Controller: 2'b00: IDLE state 2'b01: READ state (transferring data to MAC transmitter) 2'b10: Waiting for TxStatus from MAC transmitter 2'b11: Writing the received TxStatus or flushing the TxFIFO
19	RW	0x0	PAUSE When high, it indicates that the MAC transmitter is in PAUSE condition (in full-duplex only) and hence will not schedule any frame for transmission
18:17	RW	0x0	TSAT This indicates the state of the MAC Transmit Frame Controller module: 2'b00: IDLE 2'b01: Waiting for Status of previous frame or IFG/backoff period to be over 2'b10: Generating and transmitting a PAUSE control frame (in full duplex mode) 2'b11: Transferring input frame for transmission
16	RW	0x0	TACT When high, it indicates that the MAC GMII/MII transmit protocol engine is actively transmitting data and not in IDLE state
15:10	RO	0x0	reserved
9:8	RW	0x0	RFIFO This gives the status of the RxFIFO Fill-level: 2'b00: RxFIFO Empty 2'b01: RxFIFO fill-level below flow-control de-activate threshold 2'b10: RxFIFO fill-level above flow-control activate threshold 2'b11: RxFIFO Full
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:5	RW	0x0	RFIFORD It gives the state of the RxFIFO read Controller: 2'b00: IDLE state 2'b01: Reading frame data 2'b10: Reading frame status (or time-stamp) 2'b11: Flushing the frame data and Status
4	RW	0x0	RFIFOWR When high, it indicates that the MTL RxFIFO Write Controller is active and transferring a received frame to the FIFO
3	RO	0x0	reserved
2:1	RW	0x0	ACT When high, it indicates the active state of the small FIFO Read and Write controllers respectively of the MAC receive Frame Controller module
0	RW	0x0	RDB When high, it indicates that the MAC GMII/MII receive protocol engine is actively receiving data and not in IDLE state

MAC_PMT_CTRL_STA

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31	W1C	0x0	WFFRPR Wake-Up Frame Filter Register Pointer Reset When set, resets the Remote Wake-up Frame Filter register pointer to 3'b000. It is automatically cleared after 1 clock cycle
30:10	RO	0x0	reserved
9	RW	0x0	GU Global Unicast When set, enables any unicast packet filtered by the MAC (DAF) address recognition to be a wake-up frame
8:7	RO	0x0	reserved
6	RC	0x0	WFR Wake-Up Frame Received When set, this bit indicates the power management event was generated due to reception of a wake-up frame. This bit is cleared by a read into this register
5	RC	0x0	MPR Magic Packet Received When set, this bit indicates the power management event was generated by the reception of a Magic Packet. This bit is cleared by a read into this register
4:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	WFE Wake-Up Frame Enable When set, enables generation of a power management event due to wake-up frame reception
1	RW	0x0	MPE Magic Packet Enable When set, enables generation of a power management event due to Magic Packet reception
0	R/W SC	0x0	PD Power Down When set, all received frames will be dropped. This bit is cleared automatically when a magic packet or Wake-Up frame is received, and Power-Down mode is disabled. Frames received after this bit is cleared are forwarded to the application. This bit must only be set when either the Magic Packet Enable or Wake-Up Frame Enable bit is set high

MAC_INT_STATUS

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	MRCOIS MMC Receive Checksum Offload Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared
6	RO	0x0	MTIS MMC Transmit Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration
5	RO	0x0	MRIS MMC Receive Interrupt Status This bit is set high whenever an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this interrupt register are cleared. This bit is only valid when the optional MMC module is selected during configuration
4	RO	0x0	MIS MMC Interrupt Status This bit is set high whenever any of bits 7:5 is set high and cleared only when all of these bits are low. This bit is valid only when the optional MMC module is selected during configuration

Bit	Attr	Reset Value	Description
3	RO	0x0	PIS PMT Interrupt Status This bit is set whenever a Magic packet or Wake-on-LAN frame is received in Power-Down mode). This bit is cleared when both bits[6:5] are cleared due to a read operation to the register MAC_PMT_CTRL_STA
2:1	RO	0x0	reserved
0	RO	0x0	RIS RGMII Interrupt Status This bit is set due to any change in value of the Link Status of RGMII interface. This bit is cleared when the user makes a read operation the RGMII Status register

MAC_INT_MASK

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	PIM PMT Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of PMT Interrupt Status bit in Register MAC_INT_STATUS
2:1	RO	0x0	reserved
0	RW	0x0	RIM RGMII Interrupt Mask This bit when set, will disable the assertion of the interrupt signal due to the setting of RGMII Interrupt Status bit in Register MAC_INT_STATUS

MAC_MAC_ADDR0_HI

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0xffff	A47_A32 MAC Address0 [47:32] This field contains the upper 16 bits (47:32) of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames

MAC_MAC_ADDR0_LO

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	A31_A0 MAC Address0 [31:0] This field contains the lower 32 bits of the 6-byte first MAC address. This is used by the MAC for filtering for received frames and for inserting the MAC address in the Transmit Flow Control (PAUSE) Frames

MAC_AN_CTRL

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	ANE Auto-Negotiation Enable When set, will enable the MAC to perform auto-negotiation with the link partner. Clearing this bit will disable auto-negotiation
11:10	RO	0x0	reserved
9	R/W SC	0x0	RAN Restart Auto-Negotiation When set, will cause auto-negotiation to restart if the ANE is set. This bit is self-clearing after auto-negotiation starts. This bit should be cleared for normal operation
8:0	RO	0x0	reserved

MAC_AN_STATUS

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	ANC Auto-Negotiation Complete When set, this bit indicates that the auto-negotiation process is completed. This bit is cleared when auto-negotiation is reinitiated
4	RO	0x0	reserved
3	RO	0x1	ANA Auto-Negotiation Ability This bit is always high, because the MAC supports auto-negotiation
2	R/W SC	0x0	LS Link Status When set, this bit indicates that the link is up. When cleared, this bit indicates that the link is down
1:0	RO	0x0	reserved

MAC_AN_ADV

Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support This bit is tied to low, because the MAC does not support the next page
14	RO	0x0	reserved
13:12	RW	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating to a link partner that a fault or error condition has occurred
11:9	RO	0x0	reserved
8:7	RW	0x3	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the MAC is capable of configuring the PAUSE function as defined in IEEE 802.3x
6	RW	0x1	HD Half-Duplex This bit, when set high, indicates that the MAC supports Half-Duplex. This bit is tied to low (and RO) when the MAC is configured for Full-Duplex-only operation
5	RW	0x1	FD Full-Duplex This bit, when set high, indicates that the MAC supports Full-Duplex
4:0	RO	0x0	reserved

MAC_AN_LINK_PART_AB

Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RO	0x0	NP Next Page Support When set, this bit indicates that more next page information is available. When cleared, this bit indicates that next page exchange is not desired
14	RO	0x0	ACK Acknowledge When set, this bit is used by the auto-negotiation function to indicate that the link partner has successfully received the MAC's base page. When cleared, it indicates that a successful receipt of the base page has not been achieved

Bit	Attr	Reset Value	Description
13:12	RO	0x0	RFE Remote Fault Encoding These 2 bits provide a remote fault encoding, indicating a fault or error condition of the link partner
11:9	RO	0x0	reserved
8:7	RO	0x0	PSE Pause Encoding These 2 bits provide an encoding for the PAUSE bits, indicating that the link partner's capability of configuring the PAUSE function as defined in IEEE 802.3x
6	RO	0x0	HD Half-Duplex When set, this bit indicates that the link partner has the ability to operate in Half-Duplex mode. When cleared, the link partner does not have the ability to operate in Half-Duplex mode
5	RO	0x0	FD Full-Duplex When set, this bit indicates that the link partner has the ability to operate in Full-Duplex mode. When cleared, the link partner does not have the ability to operate in Full-Duplex mode
4:0	RO	0x0	reserved

MAC_AN_EXP

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	NPA Next Page Ability This bit is tied to low, because the MAC does not support next page function
1	RO	0x0	NPR New Page Received When set, this bit indicates that a new page has been received by the MAC. This bit will be cleared when read
0	RO	0x0	reserved

MAC_INTF_MODE_STA

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RO	0x0	LST Link Status Indicates whether the link is up (1'b1) or down (1'b0)

Bit	Attr	Reset Value	Description
2:1	RO	0x0	LSD Link Speed Indicates the current speed of the link: 2'b00: 2.5 MHz 2'b01: 25 MHz 2'b10: 125 MHz
0	RW	0x0	LM Link Mode Indicates the current mode of operation of the link: 1'b0: Half-Duplex mode 1'b1: Full-Duplex mode

MAC_MMC_CTRL

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	FHP Full-Half preset When low and bit4 is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2K Bytes) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16) When high and bit4 is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2K Bytes) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16)
4	R/W SC	0x0	CP Counters Preset When set, all counters will be initialized or preset to almost full or almost half as per Bit5 above. This bit will be cleared automatically after 1 clock cycle. This bit along with bit5 is useful for debugging and testing the assertion of interrupts due to MMC counter becoming half-full or full
3	RW	0x0	MCF MMC Counter Freeze When set, this bit freezes all the MMC counters to their current value. (None of the MMC counters are updated due to any transmitted or received frame until this bit is reset to 0. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.)
2	RW	0x0	ROR Reset on Read When set, the MMC counters will be reset to zero after Read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits[7:0]) is read

Bit	Attr	Reset Value	Description
1	RW	0x0	CSR Counter Stop Rollover When set, counter after reaching maximum value will not roll over to zero
0	R/W SC	0x0	CR Counters Reset When set, all counters will be reset. This bit will be cleared automatically after 1 clock cycle

MAC_MMC_RX_INTR

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 The bit is set when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value
20:19	RO	0x0	reserved
18	RC	0x0	INT18 The bit is set when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value
17:6	RO	0x0	reserved
5	RW	0x0	INT5 The bit is set when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value
4	RC	0x0	INT4 The bit is set when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value
3	RO	0x0	reserved
2	RC	0x0	INT2 The bit is set when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
1	RC	0x0	INT1 The bit is set when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TX_INTR

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RC	0x0	INT21 The bit is set when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value
20	RC	0x0	INT20 The bit is set when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
19	RC	0x0	INT19 The bit is set when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value
18:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value
12:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_RX_INT_MSK

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the rxfifooverflow counter reaches half the maximum value, and also when it reaches the maximum value
20:19	RO	0x0	reserved
18	RW	0x0	INT18 Setting this bit masks the interrupt when the rxlengtherror counter reaches half the maximum value, and also when it reaches the maximum value
17:6	RO	0x0	reserved
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxcrcerror counter reaches half the maximum value, and also when it reaches the maximum value
4	RW	0x0	INT4 Setting this bit masks the interrupt when the rxmulticastframes_g counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
3	RO	0x0	reserved
2	RW	0x0	INT2 Setting this bit masks the interrupt when the rxoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TX_INT_MSK

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RW	0x0	INT21 Setting this bit masks the interrupt when the txframecount_g counter reaches half the maximum value, and also when it reaches the maximum value
20	RW	0x0	INT20 Setting this bit masks the interrupt when the txoctetcount_g counter reaches half the maximum value, and also when it reaches the maximum value
19	RW	0x0	INT19 Setting this bit masks the interrupt when the txcarriererror counter reaches half the maximum value, and also when it reaches the maximum value
18:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the txunderflowerror counter reaches half the maximum value, and also when it reaches the maximum value
12:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the txframecount_gb counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the txoctetcount_gb counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_TXOCTETCNT_GB

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_gb Number of bytes transmitted, exclusive of preamble and retried bytes, in good and bad frames

MAC_MMC_TXFRMCNT_GB

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_gb Number of good and bad frames transmitted, exclusive of retried frames

MAC_MMC_TXUNDFLWERR

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txunderflowerror Number of frames aborted due to frame underflow error

MAC_MMC_TXCARERR

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txcarriererror Number of frames aborted due to carrier sense error (no carrier or loss of carrier)

MAC_MMC_TXOCTETCNT_G

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txoctetcount_g Number of bytes transmitted, exclusive of preamble, in good frames only

MAC_MMC_TXFRMCNT_G

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txframecount_g Number of good frames transmitted

MAC_MMC_RXFRMCNT_GB

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxframecount_gb Number of good and bad frames received

MAC_MMC_RXOCTETCNT_GB

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_gb Number of bytes received, exclusive of preamble, in good and bad frames

MAC_MMC_RXOCTETCNT_G

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxoctetcount_g Number of bytes received, exclusive of preamble, only in good frames

MAC_MMC_RXMCFRMCNT_G

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxmulticastframes_g Number of good multicast frames received

MAC_MMC_RXCRCERR

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxcrcerror Number of frames received with CRC error

MAC_MMC_RXLENERR

Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxlengtherror Number of frames received with length error (Length type field ≠frame size), for all frames with valid length field

MAC_MMC_RXFIFOOVRFLW

Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxfifooverflow Number of missed received frames due to FIFO overflow

MAC_MMC_IPC_INT_MSK

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29	RW	0x0	INT29 Setting this bit masks the interrupt when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
28	RO	0x0	reserved
27	RW	0x0	INT27 Setting this bit masks the interrupt when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
26	RO	0x0	reserved
25	RW	0x0	INT25 Setting this bit masks the interrupt when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
24:23	RO	0x0	reserved
22	RW	0x0	INT22 Setting this bit masks the interrupt when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
21:18	RO	0x0	reserved
17	RW	0x0	INT17 Setting this bit masks the interrupt when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
16:14	RO	0x0	reserved
13	RW	0x0	INT13 Setting this bit masks the interrupt when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
12	RO	0x0	reserved
11	RW	0x0	INT11 Setting this bit masks the interrupt when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
10	RO	0x0	reserved
9	RW	0x0	INT9 Setting this bit masks the interrupt when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
8:7	RO	0x0	reserved
6	RW	0x0	INT6 Setting this bit masks the interrupt when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
5	RW	0x0	INT5 Setting this bit masks the interrupt when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value
4:2	RO	0x0	reserved
1	RW	0x0	INT1 Setting this bit masks the interrupt when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
0	RW	0x0	INT0 Setting this bit masks the interrupt when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_IPC_INTR

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RC	0x0	INT29 The bit is set when the rxicmp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
28	RO	0x0	reserved
27	RC	0x0	INT27 The bit is set when the rxtcp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
26	RO	0x0	reserved
25	RC	0x0	INT25 The bit is set when the rxudp_err_octets counter reaches half the maximum value, and also when it reaches the maximum value
24:23	RO	0x0	reserved
22	RC	0x0	INT22 The bit is set when the rxipv6_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
21:18	RO	0x0	reserved
17	RC	0x0	INT17 The bit is set when the rxipv4_hdrerr_octets counter reaches half the maximum value, and also when it reaches the maximum value
16:14	RO	0x0	reserved
13	RC	0x0	INT13 The bit is set when the rxicmp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value

Bit	Attr	Reset Value	Description
12	RO	0x0	reserved
11	RC	0x0	INT11 The bit is set when the rxtcp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
10	RO	0x0	reserved
9	RC	0x0	INT9 The bit is set when the rxudp_err_frms counter reaches half the maximum value, and also when it reaches the maximum value
8:7	RO	0x0	reserved
6	RC	0x0	INT6 The bit is set when the rxipv6_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
5	RC	0x0	INT5 The bit is set when the rxipv6_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value
4:2	RO	0x0	reserved
1	RC	0x0	INT1 The bit is set when the rxipv4_hdrerr_frms counter reaches half the maximum value, and also when it reaches the maximum value
0	RC	0x0	INT0 The bit is set when the rxipv4_gd_frms counter reaches half the maximum value, and also when it reaches the maximum value

MAC_MMC_RXIPV4GFRM

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_gd_frms Number of good IPv4 datagrams received with the TCP, UDP, or ICMP payload

MAC_MMC_RXIPV4HDERRFRM

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_frms Number of IPv4 datagrams received with header (checksum, length, or version mismatch) errors

MAC_MMC_RXIPV6GFRM

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_gd_frms Number of good IPv6 datagrams received with TCP, UDP, or ICMP payloads

MAC_MMC_RXIPV6HDERRFRM

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_frms Number of IPv6 datagrams received with header errors (length or version mismatch)

MAC_MMC_RXUDPERRFRM

Address: Operational Base + offset (0x0234)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_frms Number of good IP datagrams whose UDP payload has a checksum error

MAC_MMC_RXTCPERRFRM

Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_frms Number of good IP datagrams whose TCP payload has a checksum error

MAC_MMC_RXICMPERRFRM

Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_frms Number of good IP datagrams whose ICMP payload has a checksum error

MAC_MMC_RXIPV4HDERROCT

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv4_hdrerr_octets Number of bytes received in IPv4 datagrams with header errors (checksum, length, version mismatch). The value in the Length field of IPv4 header is used to update this counter

MAC_MMC_RXIPV6HDERROCT

Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxipv6_hdrerr_octets Number of bytes received in IPv6 datagrams with header errors (length, version mismatch). The value in the IPv6 header's Length field is used to update this counter

MAC_MMC_RXUDPERROCT
Address: Operational Base + offset (0x0274)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxudp_err_octets Number of bytes received in a UDP segment that had checksum errors

MAC_MMC_RXTCPERROCT
Address: Operational Base + offset (0x027c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxtcp_err_octets Number of bytes received in a TCP segment with checksum errors

MAC_MMC_RXICMPERROCT
Address: Operational Base + offset (0x0284)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxicmp_err_octets Number of bytes received in an ICMP segment with checksum errors

MAC_BUS_MODE
Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	AAL Address-Aligned Beats When this bit is set high and the FB bit equals 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit equals 0, the first burst (accessing the data buffer's start address) is not aligned, but subsequent bursts are aligned to the address
24	RW	0x0	PBL_Mode 8xPBL Mode When set high, this bit multiplies the PBL value programmed (bits [22:17] and bits [13:8]) eight times. Thus the DMA will transfer data in to a maximum of 8, 16, 32, 64, 128, and 256 beats depending on the PBL value

Bit	Attr	Reset Value	Description
23	RW	0x0	USP Use Separate PBL When set high, it configures the RxDMA to use the value configured in bits [22:17] as PBL while the PBL value in bits [13:8] is applicable to TxDMA operations only. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines
22:17	RW	0x01	RPBL RxDMA PBL These bits indicate the maximum number of beats to be transferred in one RxDMA transaction. This will be the maximum value that is used in a single block Read/Write. The RxDMA will always attempt to burst as specified in RPBL each time it starts a Burst transfer on the host bus. RPBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. These bits are valid and applicable only when USP is set high
16	RW	0x0	FB Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AXI will use SINGLE and INCR burst transfer operations
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x01	PBL Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This will be the maximum value that is used in a single block Read/Write. The DMA will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, 16, and 32. Any other value will result in undefined behavior. When USP is set high, this PBL value is applicable for TxDMA transactions only. The PBL values have the following limitations. The maximum number of beats (PBL) possible is limited by the size of the Tx FIFO and Rx FIFO in the MTL layer and the data bus width on the DMA. The FIFO has a constraint that the maximum beat supported is half the depth of the FIFO, except when specified (as given below). For different data bus widths and FIFO sizes, the valid PBL range (including x8 mode) is provided in the following table. If the PBL is common for both transmit and receive DMA, the minimum Rx FIFO and Tx FIFO depths must be considered. Do not program out-of-range PBL values, because the system may not behave properly. For TxFIFO, valid PBL range in full duplex mode and duplex mode is 128 or less. For RxFIFO, valid PBL range in full duplex mode is all
7	RO	0x0	reserved
6:2	RW	0x00	DSL Descriptor Skip Length This bit specifies the number of dword to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When DSL value equals zero, then the descriptor table is taken as contiguous by the DMA, in Ring mode
1	RO	0x0	reserved
0	R/W SC	0x1	SWR Software Reset When this bit is set, the MAC DMA Controller resets all MAC Subsystem internal registers and logic. It is cleared automatically after the reset operation has completed in all of the core clock domains. Read a 0 value in this bit before re-programming any register of the core. Note: The reset operation is completed only when all the resets in all the active clock domains are de-asserted. Hence it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion

MAC_TX_POLL_DEMAND

Address: Operational Base + offset (0x1004)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TPD Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register MAC_CUR_HOST_TX_DESC. If that descriptor is not available (owned by Host), transmission returns to the Suspend state and DMA Register MAC_STATUS[2] is asserted. If the descriptor is available, transmission resumes

MAC_RX_POLL_DEMAND

Address: Operational Base + offset (0x1008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RPD Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Register MAC_CUR_HOST_RX_DESC. If that descriptor is not available (owned by Host), reception returns to the Suspended state and Register MAC_STATUS[7] is not asserted. If the descriptor is available, the Receive DMA returns to active state

MAC_RX_DESC_LIST_ADDR

Address: Operational Base + offset (0x100c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SRL Start of Receive List This field contains the base address of the First Descriptor in the Receive Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only

MAC_TX_DESC_LIST_ADDR

Address: Operational Base + offset (0x1010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	STL Start of Transmit List This field contains the base address of the First Descriptor in the Transmit Descriptor list. The LSB bits [1/2/3:0] for 32/64/128-bit bus width) will be ignored and taken as all-zero by the DMA internally. Hence these LSB bits are Read Only

MAC_STATUS

Address: Operational Base + offset (0x1014)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	GPI MAC PMT Interrupt This bit indicates an interrupt event in the MAC core's PMT module. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear its source to reset this bit to 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
27	RO	0x0	GMI MAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the MAC core. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
26	RO	0x0	GLI MAC Line interface Interrupt This bit reflects an interrupt event in the MAC Core's PCS or RGMII interface block. The software must read the corresponding registers in the MAC core to get the exact cause of interrupt and clear the source of interrupt to make this bit as 1'b0. The interrupt signal from the MAC subsystem (sbd_intr_o) is high when this bit is high
25:23	RO	0x0	EB Error Bits These bits indicate the type of error that caused a Bus Error (e.g., error response on the AXI interface). Valid only with Fatal Bus Error bit (Register MAC_STATUS[13]) set. This field does not generate an interrupt. Bit 23: 1'b1 Error during data transfer by TxDMA 1'b0 Error during data transfer by RxDMA Bit 24: 1'b1 Error during read transfer 1'b0 Error during write transfer Bit 25: 1'b1 Error during descriptor access 1'b0 Error during data buffer access

Bit	Attr	Reset Value	Description
22:20	RO	0x0	TS Transmit Process State These bits indicate the Transmit DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped; Reset or Stop Transmit Command issued. 3'b001: Running; Fetching Transmit Transfer Descriptor. 3'b010: Running; Waiting for status. 3'b011: Running; Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO). 3'b100: TIME_STAMP write state. 3'b101: Reserved for future use. 3'b110: Suspended; Transmit Descriptor Unavailable or Transmit Buffer Underflow. 3'b111: Running; Closing Transmit Descriptor
19:17	RO	0x0	RS Receive Process State These bits indicate the Receive DMA FSM state. This field does not generate an interrupt. 3'b000: Stopped: Reset or Stop Receive Command issued. 3'b001: Running: Fetching Receive Transfer Descriptor. 3'b010: Reserved for future use. 3'b011: Running: Waiting for receive packet. 3'b100: Suspended: Receive Descriptor Unavailable. 3'b101: Running: Closing Receive Descriptor. 3'b110: TIME_STAMP write state. 3'b111: Running: Transferring the receive packet data from receive buffer to host memory
16	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 MAC_STATUS[0]: Transmit Interrupt Register MAC_STATUS[2]: Transmit Buffer Unavailable Register MAC_STATUS[6]: Receive Interrupt Register MAC_STATUS[14]: Early Receive Interrupt Only unmasked bits affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing a 1 to this bit) each time a corresponding bit that causes NIS to be set is cleared

Bit	Attr	Reset Value	Description
15	W1C	0x0	AIS Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in Register OP_MODE: Register MAC_STATUS[1]: Transmit Process Stopped Register MAC_STATUS[3]: Transmit Jabber Timeout Register MAC_STATUS[4]: Receive FIFO Overflow Register MAC_STATUS[5]: Transmit Underflow Register MAC_STATUS[7]: Receive Buffer Unavailable Register MAC_STATUS[8]: Receive Process Stopped Register MAC_STATUS[9]: Receive Watchdog Timeout Register MAC_STATUS[10]: Early Transmit Interrupt Register MAC_STATUS[13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared
14	W1C	0x0	ERI Early Receive Interrupt This bit indicates that the DMA had filled the first data buffer of the packet. Receive Interrupt Register MAC_STATUS[6] automatically clears this bit
13	W1C	0x0	FBI Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as detailed in [25:23]. When this bit is set, the corresponding DMA engine disables all its bus accesses
12:11	RO	0x0	reserved
10	W1C	0x0	ETI Early Transmit Interrupt This bit indicates that the frame to be transmitted was fully transferred to the MTL Transmit FIFO
9	W1C	0x0	RWT Receive Watchdog Timeout This bit is asserted when a frame with a length greater than 2,048 bytes is received
8	W1C	0x0	RPS Receive Process Stopped This bit is asserted when the Receive Process enters the Stopped state

Bit	Attr	Reset Value	Description
7	W1 C	0x0	RU Receive Buffer Unavailable This bit indicates that the Next Descriptor in the Receive List is owned by the host and cannot be acquired by the DMA. Receive Process is suspended. To resume processing Receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, Receive Process resumes when the next recognized incoming frame is received. Register MAC_STATUS[7] is set only when the previous Receive Descriptor was owned by the DMA
6	W1 C	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	W1 C	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	W1 C	0x0	OVF Receive Overflow This bit indicates that the Receive Buffer had an Overflow during frame reception. If the partial frame is transferred to application, the overflow status is set in RDES0[11]
3	W1 C	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	W1 C	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	W1 C	0x0	TPS Transmit Process Stopped This bit is set when the transmission is stopped

Bit	Attr	Reset Value	Description
0	W1 C	0x0	TI Transmit Interrupt This bit indicates that frame transmission is finished and TDES1[31] is set in the First Descriptor

MAC_OP_MODE

Address: Operational Base + offset (0x1018)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	DT Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the core does not drop frames that only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the MAC but have errors in the encapsulated payload only. When this bit is reset, all error frames are dropped if the FEF bit is reset
25	RW	0x0	RSF Receive Store and Forward When this bit is set, the MTL only reads a frame from the Rx FIFO after the complete frame has been written to it, ignoring RTC bits. When this bit is reset, the Rx FIFO operates in Cut-Through mode, subject to the threshold specified by the RTC bits
24	RW	0x0	DFF Disable Flushing of Received Frames When this bit is set, the RxDMA does not flush any frames due to the unavailability of receive descriptors/buffers as it does normally when this bit is reset
23:22	RO	0x0	reserved
21	RW	0x0	TSF Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in Register MAC_OP_MODE[16:14] are ignored. This bit should be changed only when transmission is stopped

Bit	Attr	Reset Value	Description
20	W1 C	0x0	FTF Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the Tx FIFO is lost/flushed. This bit is cleared internally when the flushing operation is completed fully. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the MAC transmitter will not be flushed. It will be scheduled for transmission and will result in underflow and runt frame transmission. Note: The flush operation completes only after emptying the TxFIFO of its contents and all the pending Transmit Status of the transmitted frames are accepted by the host. In order to complete this flush operation, the PHY transmit clock (clk_tx_i) is required to be active
19:17	RO	0x0	reserved
16:14	RW	0x0	TTC Transmit Threshold Control These three bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL Transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when the TSF bit (Bit 21) is reset. 3'b000: 64 3'b001: 128 3'b010: 192 3'b011: 256 3'b100: 40 3'b101: 32 3'b110: 24 3'b111: 16

Bit	Attr	Reset Value	Description
13	RW	0x0	ST Start/Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Register MAC_TX_DESC_LIST_ADDR, or from the position retained when transmission was stopped previously. If the current descriptor is not owned by the DMA, transmission enters the Suspended state and Transmit Buffer Unavailable (Register MAC_STATUS[2]) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting DMA Register TX_DESC_LIST_ADDR, then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The Next Descriptor position in the Transmit List is saved, and becomes the current position when transmission is restarted. The stop transmission command is effective only the transmission of the current frame is complete or when the transmission is in the Suspended state
12:11	RW	0x0	RFD Threshold for deactivating flow control (in both HD and FD) These bits control the threshold (Fill-level of Rx FIFO) at which the flow-control is de-asserted after activation. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the de-assertion is effective only after flow control is asserted
10:9	RW	0x0	RFA Threshold for activating flow control (in both HD and FD) These bits control the threshold (Fill level of Rx FIFO) at which flow control is activated. 2'b00: Full minus 1 KB 2'b01: Full minus 2 KB 2'b10: Full minus 3 KB 2'b11: Full minus 4 KB Note that the above only applies to Rx FIFOs of 4 KB or more when the EFC bit is set high

Bit	Attr	Reset Value	Description
8	RW	0x0	EFC Enable HW flow control When this bit is set, the flow control signal operation based on fill-level of Rx FIFO is enabled. When reset, the flow control operation is disabled
7	RW	0x0	FEF Forward Error Frames When this bit is reset, the Rx FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, overflow). However, if the frame's start byte (write) pointer is already transferred to the read controller side (in Threshold mode), then the frames are not dropped. When FEF is set, all frames except runt error frames are forwarded to the DMA. But when RxFIFO overflows when a partial frame is written, then such frames are dropped even when FEF is set
6	RW	0x0	FUF Forward Undersized Good Frames When set, the Rx FIFO will forward Undersized frames (frames with no Error and length less than 64 bytes) including pad-bytes and CRC). When reset, the Rx FIFO will drop all frames of less than 64 bytes, unless it is already transferred due to lower value of Receive Threshold (e.g., RTC = 01)
5	RO	0x0	reserved
4:3	RW	0x0	RTC Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are transferred automatically. Note that value of 11 is not applicable if the configured Receive FIFO size is 128 bytes. These bits are valid only when the RSF bit is zero, and are ignored when the RSF bit is set to 1. 2'b00: 64 2'b01: 32 2'b10: 96 2'b11: 128
2	RW	0x0	OSF Operate on Second Frame When this bit is set, this bit instructs the DMA to process a second frame of Transmit data even before status for first frame is obtained

Bit	Attr	Reset Value	Description
1	RW	0x0	SR Start/Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes incoming frames. Descriptor acquisition is attempted from the current position in the list, which is the address set by register MAC_RX_DESC_LIST_ADDR or the position retained when the Receive process was previously stopped. If no descriptor is owned by the DMA, reception is suspended and Receive Buffer Unavailable (Register MAC_STATUS[7]) is set. The Start Receive command is effective only when reception has stopped. If the command was issued before setting register MAC_RX_DESC_LIST_ADDR, DMA behavior is unpredictable. When this bit is cleared, RxDMA operation is stopped after the transfer of the current frame. The next descriptor position in the Receive list is saved and becomes the current position after the Receive process is restarted. The Stop Receive command is effective only when the Receive process is in either the Running (waiting for receive packet) or in the Suspended state
0	RO	0x0	reserved

MAC_INT_ENA

Address: Operational Base + offset (0x101c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NIE Normal Interrupt Summary Enable When this bit is set, a normal interrupt is enabled. When this bit is reset, a normal interrupt is disabled. This bit enables the following bits: Register MAC_STATUS[0]: Transmit Interrupt Register MAC_STATUS[2]: Transmit Buffer Unavailable Register MAC_STATUS[6]: Receive Interrupt Register MAC_STATUS[14]: Early Receive Interrupt

Bit	Attr	Reset Value	Description
15	RW	0x0	AIE Abnormal Interrupt Summary Enable When this bit is set, an Abnormal Interrupt is enabled. When this bit is reset, an Abnormal Interrupt is disabled. This bit enables the following bits Register MAC_STATUS[1]: Transmit Process Stopped Register MAC_STATUS[3]: Transmit Jabber Timeout Register MAC_STATUS[4]: Receive Overflow Register MAC_STATUS[5]: Transmit Underflow Register MAC_STATUS[7]: Receive Buffer Unavailable Register MAC_STATUS[8]: Receive Process Stopped Register MAC_STATUS[9]: Receive Watchdog Timeout Register MAC_STATUS[10]: Early Transmit Interrupt Register MAC_STATUS[13]: Fatal Bus Error
14	RW	0x0	ERE Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Early Receive Interrupt is enabled. When this bit is reset, Early Receive Interrupt is disabled
13	RW	0x0	FBE Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Fatal Bus Error Interrupt is enabled. When this bit is reset, Fatal Bus Error Enable Interrupt is disabled
12:11	RO	0x0	reserved
10	RW	0x0	ETE Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (BIT 15), Early Transmit Interrupt is enabled. When this bit is reset, Early Transmit Interrupt is disabled
9	RW	0x0	RWE Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, Receive Watchdog Timeout Interrupt is disabled
8	RW	0x0	RSE Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Stopped Interrupt is enabled. When this bit is reset, Receive Stopped Interrupt is disabled

Bit	Attr	Reset Value	Description
7	RW	0x0	RUE Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled
6	RW	0x0	RIE Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Receive Interrupt is enabled. When this bit is reset, Receive Interrupt is disabled
5	RW	0x0	UNE Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Underflow Interrupt is enabled. When this bit is reset, Underflow Interrupt is disabled
4	RW	0x0	OVE Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Receive Overflow Interrupt is enabled. When this bit is reset, Overflow Interrupt is disabled
3	RW	0x0	TJE Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, Transmit Jabber Timeout Interrupt is disabled
2	RW	0x0	TUE Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, Transmit Buffer Unavailable Interrupt is disabled
1	RW	0x0	TSE Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (BIT 15), Transmission Stopped Interrupt is enabled. When this bit is reset, Transmission Stopped Interrupt is disabled
0	RW	0x0	TIE Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (BIT 16), Transmit Interrupt is enabled. When this bit is reset, Transmit Interrupt is disabled

MAC_OVERFLOW_CNT

Address: Operational Base + offset (0x1020)

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RC	0x0	FIFO_overflow_bit Overflow bit for FIFO Overflow Counter
27:17	RC	0x000	Frame_miss_number Indicates the number of frames missed by the application This counter is incremented each time the MTL asserts the sideband signal mtl_rxoverflow_o. The counter is cleared when this register is read with mci_be_i[2] at 1'b1
16	RC	0x0	Miss_frame_overflow_bit Overflow bit for Missed Frame Counter
15:0	RC	0x0000	Frame_miss_number_2 Indicates the number of frames missed by the controller due to the Host Receive Buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this register is read with mci_be_i[0] at 1'b1

MAC_REC_INT_WDT_TIMER

Address: Operational Base + offset (0x1024)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	RIWT RI Watchdog Timer count Indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RxDMA completes the transfer of a frame for which the RI status bit is not set due to the setting in the corresponding descriptor RDES1[31]. When the watch-dog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when RI bit is set high due to automatic setting of RI as per RDES1[31] of any received frame

MAC_AXI_BUS_MODE

Address: Operational Base + offset (0x1028)

Bit	Attr	Reset Value	Description
31	RW	0x0	EN_LPI Enable LPI (Low Power Interface) When set to 1, enable the LPI (Low Power Interface) supported by the MAC and accepts the LPI request from the AXI System Clock controller. When set to 0, disables the Low Power Mode and always denies the LPI request from the AXI System Clock controller

Bit	Attr	Reset Value	Description
30	RW	0x0	UNLCK_ON_MGK_RWK Unlock on Magic Packet or Remote Wake Up When set to 1, enables it to request coming out of Low Power mode only when Magic Packet or Remote Wake Up Packet is received. When set to 0, enables it requests to come out of Low Power mode when any frame is received
29:22	RO	0x0	reserved
21:20	RW	0x1	WR_OSR_LMT AXI Maximum Write Out Standing Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT+1
19:18	RO	0x0	reserved
17:16	RW	0x1	RD_OSR_LMT AXI Maximum Read Out Standing Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT+1
15:13	RO	0x0	reserved
12	RO	0x0	AXI_AAL Address-Aligned Beats This bit is read-only bit and reflects the AAL bit Register0 (register MAC_BUS_MODE[25]). When this bit set to 1, it performs address-aligned burst transfers on both read and write channels
11:4	RO	0x0	reserved
3	RW	0x0	BLEN16 AXI Burst Length 16 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 16
2	RW	0x0	BLEN8 AXI Burst Length 8 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 8
1	RW	0x0	BLEN4 AXI Burst Length 4 When this bit is set to 1, or when UNDEF is set to 1, it is allowed to select a burst length of 4

Bit	Attr	Reset Value	Description
0	RO	0x1	UNDEF AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of FB bit in register MAC_BUS_MODE[16]. When this bit is set to 1, it is allowed to perform any burst length equal to or below the maximum allowed burst length as programmed in bits[7:1]; When this bit is set to 0, it is allowed to perform only fixed burst lengths as indicated by BLEN256/128/64/32/16/8/4, or a burst length of 1

MAC_AXI_STATUS

Address: Operational Base + offset (0x102c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	RD_CH_STA When high, it indicates that AXI Master's read channel is active and transferring data
0	RO	0x0	WR_CH_STA When high, it indicates that AXI Master's write channel is active and transferring data

MAC_CUR_HOST_TX_DESC

Address: Operational Base + offset (0x1048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTDAP Host Transmit Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_RX_DESC

Address: Operational Base + offset (0x104c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRDAP Host Receive Descriptor Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_TX_BUF_ADDR

Address: Operational Base + offset (0x1050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HTBAP Host Transmit Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation

MAC_CUR_HOST_RX_BUF_ADDR

Address: Operational Base + offset (0x1054)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HRBAP Host Receive Buffer Address Pointer Cleared on Reset. Pointer updated by DMA during operation

23.5 Interface Description

Table 23-1 RMII Interface Description

Module pin	Direction	Pad name	IOMUX setting
RMII interface			
mac_clk	I/O	GPIO1_B4/LCDC_D8/I2S1_8CH_MCLK_M1/MAC_CLK	GRF_GPIO1B_IOMUX_L[9:8]=2'b11
mac_txen	O	GPIO1_C1/LCDC_D13/I2S1_8CH_SDO0_M1/MAC_TXEN	GRF_GPIO1C_IOMUX_L[3:2]=2'b11
mac_txd1	O	GPIO1_C3/LCDC_D15/I2S1_8CH_SDO2_SDI2_M1/PDM_8CH_SDI2_M1/MAC_TXD1	GRF_GPIO1C_IOMUX_L[6:4]=3'b011
mac_txd0	O	GPIO1_C2/LCDC_D14/I2S1_8CH_SDO1_SDI3_M1/PDM_8CH_SDI3_M1/MAC_TXD0	GRF_GPIO1C_IOMUX_L[10:8]=3'b011
mac_rxdv	I	GPIO1_C0/LCDC_D12/I2S1_8CH_LRCK_RX_M1/MAC_RXDV	GRF_GPIO1C_IOMUX_L[1:0]=2'b11
mac_rxer	I	GPIO1_B7/LCDC_D11/I2S1_8CH_LRCK_TX_M1/MAC_RXER	GRF_GPIO1B_IOMUX_H[2:0]=3'b011
mac_rxd1	I	GPIO1_C5/LCDC_D17/I2S1_8CH_SDI0_M1/PDM_8CH_SDI0_M1/MAC_RXD1	GRF_GPIO1C_IOMUX_H[2:0]=3'b011
mac_rxd0	I	GPIO1_C4/LCDC_D16/I2S1_8CH_SDO3_SDI1_M1/PDM_8CH_SDI1_M1/MAC_RXD0	GRF_GPIO1C_IOMUX_L[14:12]=3'b011
Management interface			
mac_mdio	I/O	GPIO1_B6/LCDC_D10/I2S1_8CH_SCLK_RX_M1/PDM_8CH_CLK_M1/MAC_MDIO	GRF_GPIO1B_IOMUX_L[14:12]=3'b011
mac_mdc	O	GPIO1_B5/LCDC_D9/I2S1_8CH_SCLK_TX_M1/MAC_MDC	GRF_GPIO1B_IOMUX_L[11:10]=2'b11

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

23.6 Application Notes

23.6.1 Descriptors

The DMA in MAC can communicate with Host driver through descriptor lists and data buffers. The DMA transfers data frames received by the core to the Receive Buffer in the Host memory, and Transmit data frames from the Transmit Buffer in the Host memory. Descriptors that reside in the Host memory act as pointers to these buffers.

There are two descriptor lists; one for reception, and one for transmission. The base address of each list is written into DMA Registers RX_DESC_LIST_ADDR and TX_DESC_LIST_ADDR, respectively. A descriptor list is forward linked (either implicitly or explicitly). The last descriptor may point back to the first entry to create a ring structure. Explicit chaining of descriptors is accomplished by setting the second address chained in both Receive and Transmit descriptors (RDES1[24] and TDES1[24]). The descriptor lists resides in the Host physical memory address space. Each descriptor can point to a maximum of two buffers. This enables two buffers to be used, physically addressed, rather than contiguous buffers in memory.

A data buffer resides in the Host physical memory space, and consists of an entire frame or

part of a frame, but cannot exceed a single frame. Buffers contain only data, buffer status is maintained in the descriptor. Data chaining refers to frames that span multiple data buffers. However, a single descriptor cannot span multiple frames. The DMA will skip to the next frame buffer when end-of-frame is detected. Data chaining can be enabled or disabled
The descriptor ring and chain structure is shown in following figure.

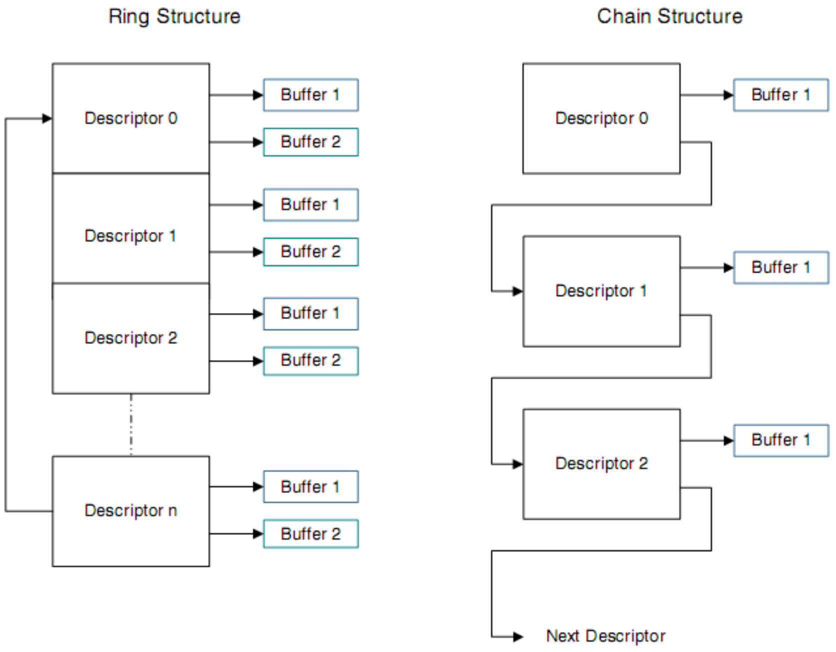


Fig. 23-11 Descriptor Ring and Chain Structure

Each descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes. The descriptor addresses must be aligned to the bus width used (Word/Dword/Lword for 32/64/128-bit buses).

	63 55	47	39	31	23 15 7 0
DES1-DES0	Control Bits [9:0]	Byte Count Buffer2 [10:0]	Byte Count Buffer1[10:0]	OWN	Status [30:0]
DES3-DES2	Buffer2 Address [31:0] / Next Descriptor Address [31:0]			Buffer1 Address[31:0]	

Fig. 23-12 Rx/Tx Descriptors definition

23.6.2 Receive Descriptor

The MAC Subsystem requires at least two descriptors when receiving a frame. The Receive state machine of the DMAalways attempts to acquire an extra descriptor in anticipation of an incoming frame. (The size of the incoming frame is unknown). Before the RxDMA closes a descriptor, it will attempt to acquire the next descriptor even if no frames are received.
In a single descriptor (receive) system, the subsystem will generate a descriptor error if the receive buffer is unable to accommodate the incoming frame and the next descriptor is not owned by the DMA. Thus, the Host is forced to increase either its descriptor pool or the buffer size. Otherwise, the subsystem starts dropping all incoming frames.

Receive Descriptor 0 (RDES0)
RDES0 contains the received frame status, the frame length, and the descriptor ownership information.

Table 23-2 Receive Descriptor 0

Bit	Description
31	OWN: Own Bit

Bit	Description
	When set, this bit indicates that the descriptor is owned by the DMA of the MAC Subsystem. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.
30	AFM: Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the MAC Core.
29:16	FL: Frame Length These bits indicate the byte length of the received frame that was transferred to host memory (including CRC). This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame.
15	ES: Error Summary Indicates the logical OR of the following bits: • RDES0[0]: Payload Checksum Error • RDES0[1]: CRC Error • RDES0[3]: Receive Error • RDES0[4]: Watchdog Timeout • RDES0[6]: Late Collision • RDES0[7]: IPC Checksum • RDES0[11]: Overflow Error • RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
14	DE: Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the Next Descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set
13	SAF: Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the MAC Core.
12	LE: Length Error When set, this bit indicates that the actual length of the frame received and that the Length/ Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset. Length error status is not valid when CRC error is present.
11	OE: Overflow Error When set, this bit indicates that the received frame was damaged due to buffer overflow.
10	VLAN: VLAN Tag When set, this bit indicates that the frame pointed to by this descriptor is a VLAN frame tagged by the MAC Core.
9	FS: First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next Descriptor contains the beginning of the frame.
8	LS: Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
7	IPC Checksum Error/Giant Frame

Bit	Description
	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 23-3 Receive Descriptor 1

Bit	Description
31	Disable Interrupt on Completion When set, this bit will prevent the setting of the RI (CSR5[6]) bit of the MAC_STATUS Register for the received frame that ends in the buffer pointed to by this descriptor. This, in turn, will disable the assertion of the interrupt to Host due to RI for that frame.
30:26	Reserved.
25	RER: Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a Descriptor Ring.
24	RCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next

Bit	Description
	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 23-4 Receive Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. Note that the DMA performs a write operation with the RDES2[2:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[2:0] (corresponding to bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

Receive Descriptor 3 (RDES3)
RDES3 contains the address pointer either to the second data buffer in the descriptor or to the next descriptor.

Table 23-5 Receive Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. If RDES1[24] is set, the buffer (Next Descriptor) address pointer must be bus width-aligned (RDES3[2:0] = 0, corresponding to a bus width of 64. LSBs are ignored internally.) However, when RDES1[24] is reset, there are no limitations on the RDES3 value, except for the following condition: The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. The DMA ignores RDES3[2:0] (corresponding to a bus width of 64) if the address pointer is to a buffer where the middle or last part of the frame is stored.

23.6.3 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 23-6 Transmit Descriptor 0

Bit	Description
31	OWN: Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are empty. The ownership bit of the First Descriptor of the frame should be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.
30:17	Reserved.
16	IHE: IP Header Error When set, this bit indicates that the Checksum Offload engine detected an IP header error and consequently did not modify the transmitted frame for any checksum insertion.
15	ES: Error Summary Indicates the logical OR of the following bits: • TDES0[14]: Jabber Timeout • TDES0[13]: Frame Flush • TDES0[11]: Loss of Carrier • TDES0[10]: No Carrier • TDES0[9]: Late Collision • TDES0[8]: Excessive Collision • TDES0[2]: Excessive Deferral • TDES0[1]: Underflow Error
14	JT: Jabber Timeout When set, this bit indicates the MAC transmitter has experienced a jabber time-out.
13	FF: Frame Flushed When set, this bit indicates that the DMA/MTL flushed the frame due to a SW flush command given by the CPU.
12	PCE: Payload Checksum Error This bit, when set, indicates that the Checksum Offload engine had a failure and did not insert any checksum into the encapsulated TCP, UDP, or ICMP payload. This failure can be either due to insufficient bytes, as indicated by the IP Header's Payload Length field, or the MTL starting to forward the frame to the MAC transmitter in Store-and-Forward mode without the checksum having been calculated yet. This second error condition only occurs when the Transmit FIFO depth is less than the length of the Ethernet frame being transmitted: to avoid deadlock, the MTL starts forwarding the frame when the FIFO is full, even in Store-and-Forward mode.
11	LC: Loss of Carrier When set, this bit indicates that Loss of Carrier occurred during frame transmission. This is valid only for the frames transmitted without collision and when the MAC operates in Half-Duplex Mode.
10	NC: No Carrier When set, this bit indicates that the carrier sense signal form the PHY was not asserted during transmission.
9	LC: Late Collision When set, this bit indicates that frame transmission was aborted due to a collision occurring after the collision window (64 byte times including Preamble in RMII

Bit	Description
	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 MAC Configuration Register is set, this bit is set after the first collision and the transmission of the frame is aborted.
7	VF: VLAN Frame When set, this bit indicates that the transmitted frame was a VLAN-type frame.
6:3	CC: Collision Count This 4-bit counter value indicates the number of collisions occurring before the frame was transmitted. The count is not valid when the Excessive Collisions bit (TDES0[8]) is set.
2	ED: Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288 bit times (155,680 bits times in 1000-Mbps mode) if the Deferral Check (DC) bit is set high in the MAC Control Register.
1	UF: Underflow Error When set, this bit indicates that the MAC aborted the frame because data arrived late from the Host memory. Underflow Error indicates that the DMA encountered an empty Transmit Buffer while transmitting the frame. The transmission process enters the suspended state and sets both Transmit Underflow (Register MAC_STATUS[5]) and Transmit Interrupt (Register MAC_STATUS [0]).
0	DB: Deferred Bit When set, this bit indicates that the MAC defers before transmission because of the presence of carrier. This bit is valid only in Half-Duplex mode.

Transmit Descriptor 1 (TDES1)

TDES1 contains the buffer sizes and other bits which control the descriptor chain/ring and the frame being transferred.

Table 23-7 Transmit Descriptor 1

Bit	Description
31	IC: Interrupt on Completion When set, this bit sets Transmit Interrupt (Register 5[0]) after the present frame has been transmitted.
30	LS: Last Segment When set, this bit indicates that the buffer contains the last segment of the frame.
29	FS: First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
28:27	CIC: Checksum Insertion Control These bits control the insertion of checksums in Ethernet frames that encapsulate TCP, UDP, or ICMP over IPv4 or IPv6 as described below. • 2'b00: Do nothing. Checksum Engine is bypassed • 2'b01: Insert IPv4 header checksum. Use this value to insert IPv4 header checksum when the frame encapsulates an IPv4 datagram. • 2'b10: Insert TCP/UDP/ICMP checksum. The checksum is calculated over the TCP, UDP, or ICMP segment only and the TCP, UDP, or ICMP pseudo-header checksum is assumed to be present in the corresponding input frame's Checksum field. An IPv4 header checksum is also inserted if the encapsulated datagram conforms to IPv4. • 2'b11: Insert a TCP/UDP/ICMP checksum that is fully calculated in this engine. In other words, the TCP, UDP, or ICMP pseudo-header is included in the checksum calculation, and the input frame's corresponding Checksum field has an all-zero value. An IPv4 Header checksum is also inserted if the encapsulated datagram

Bit	Description
	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 MAC does not append the Cyclic Redundancy Check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES1[29]).
25	TER: Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The returns to the base address of the list, creating a descriptor ring.
24	TCH: Second Address Chained When set, this bit indicates that the second address in the descriptor is the Next Descriptor address rather than the second buffer address. When TDES1[24] is set, TBS2 (TDES1[21–11]) are "don't care" values. TDES1[25] takes precedence over TDES1[24].
23	DP: Disable Padding When set, the MAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes and the CRC field is added despite the state of the DC (TDES1[26]) bit. This is valid only when the first segment (TDES1[29]) is set.
22	Reserved.
21:11	TBS2: Transmit Buffer 2 Size These bits indicate the Second Data Buffer in bytes. This field is not valid if TDES1[24] is set.
10:0	TBS1: Transmit Buffer 1 Size These bits indicate the First Data Buffer byte size. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of TCH (Bit 24).

Transmit Descriptor 2 (TDES2)

TDES2 contains the address pointer to the first buffer of the descriptor.

Table 23-8 Transmit Descriptor 2

Bit	Description
31:0	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Transmit Descriptor 3 (TDES3)

TDES3 contains the address pointer either to the second buffer of the descriptor or the next descriptor.

Table 23-9 Transmit Descriptor 3

Bit	Description
31:0	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES1[24]) bit is set, this address contains the pointer to the physical memory where the Next Descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES1[24] is set. (LSBs are ignored internally.)

23.6.4 Programming Guide

DMA Initialization – Descriptors

The following operations must be performed to initialize the DMA.

1. Provide a software reset. This will reset all of the MAC internal registers and logic. (MAC_OP_MODE[0]).
2. Wait for the completion of the reset process (poll MAC_OP_MODE[0], which is only cleared after the reset operation is completed).
3. Program the following fields to initialize the Bus Mode Register by setting values in register MAC_BUS_MODE
   a. Mixed Burst and AAL
   b. Fixed burst or undefined burst
   c. Burst length values and burst mode values.
   d. Descriptor Length (only valid if Ring Mode is used)
   e. Tx and Rx DMA Arbitration scheme
4. Program the AXI Interface options in the register MAC_BUS_MODE
   a. If fixed burst-length is enabled, then select the maximum burst-length possible on the AXI bus (Bits[7:1])
5. A proper descriptor chain for transmit and receive must be created. It should also ensure that the receive descriptors are owned by DMA (bit 31 of descriptor should be set). When OSF mode is used, at least two descriptors are required.
6. Software should create three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.
7. Initialize receive and transmit descriptor list address with the base address of transmit and receive descriptor (register MAC_RX_DESC_LIST_ADDR and MAC_TX_DESC_LIST_ADDR).
8. Program the following fields to initialize the mode of operation by setting values in register MAC_OP_MODE
   a. Receive and Transmit Store And Forward
   b. Receive and Transmit Threshold Control (RTC and TTC)
   c. Hardware Flow Control enable
   d. Flow Control Activation and De-activation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
   e. Error Frame and undersized good frame forwarding enable
   f. OSF Mode
9. Clear the interrupt requests, by writing to those bits of the status register (interrupt bits only) which are set. For example, by writing 1 into bit 16 - normal interrupt summary will clear this bit (register MAC_STATUS).
10. Enable the interrupts by programming the interrupt enable register MAC_INT_ENA.
11. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the control register MAC_OP_MODE.

MAC Initialization

The following MAC Initialization operations can be performed after the DMA initialization sequence. If the MAC Initialization is done before the DMA is set-up, then enable the MAC receiver (last step below) only after the DMA is active. Otherwise, received frames will fill the RxFIFO and overflow.

1. Program the register MAC_GMII_ADDR for controlling the management cycles for external PHY, for example, Physical Layer Address PA (bits 15-11). Also set bit 0 (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of (MAC_GMII_DATA) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in register MAC_GMII_ADDR (bits 15-11).
3. Provide the MAC address registers (MAC_MAC_ADDR0_HI and MAC_MAC_ADDR0_LO).
4. If Hash filtering is enabled in your configuration, program the Hash filter register (MAC_HASH_TAB_HI and MAC_HASH_TAB_LO).
5. Program the following fields to set the appropriate filters for the incoming frames in register MAC_MAC_FRM_FILT
   a. Receive All
   b. Promiscuous mode
   c. Hash or Perfect Filter

d. Unicast, Multicast, broad cast and control frames filter settings etc.
6. Program the following fields for proper flow control in register MAC_FLOW_CTRL.
a. Pause time and other pause frame control bits
b. Receive and Transmit Flow control bits
c. Flow Control Busy/Backpressure Activate
7. Program the Interrupt Mask register bits, as required, and if applicable, for your configuration.
8. Program the appropriate fields in register MAC_MAC_CONF for example, Inter-frame gap while transmission, jabber disable, etc. Based on the Auto-negotiation you can set the Duplex mode (bit 11), port select (bit 15), etc.
9. Set the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register MAC_MAC_CONF.

Normal Receive and Transmit Operation
For normal operation, the following steps can be followed.

- For normal transmit and receive interrupts, read the interrupt status. Then poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
- On completion of the above step, set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
- If the descriptors were not owned by the DMA (or no descriptor is available), the DMA will go into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Tx/Rx poll demand register (MAC_TX_POLL_DEMAND and MAC_RX_POLL_DEMAND).
- The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (MAC_CUR_HOST_TX_DESC and MAC_CUR_HOST_RX_DESC).
- The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (MAC_CUR_HOST_TX_Buf_ADDR and MAC_CUR_HOST_RX_BUF_ADDR).

Stop and Start Operation
When the transmission is required to be paused for some time then the following steps can be followed.
1. Disable the Transmit DMA (if applicable), by clearing ST (bit 13) of the control register MAC_OP_MODE.
2. Wait for any previous frame transmissions to complete. This can be checked by reading the appropriate bits of MAC Debug register.
3. Disable the MAC transmitter and MAC receiver by clearing the bits Transmit enable (TE bit-3) and Receive Enable (RE bit-2) in register MAC_MAC_CONF.
4. Disable the Receive DMA (if applicable), after making sure the data in the RX FIFO is transferred to the system memory (by reading the register MAC_DEBUG).
5. Make sure both the TX FIFO and RX FIFO are empty.
6. To re-start the operation, start the DMAs first, before enabling the MAC Transmitter and Receiver.

23.6.5 Clock Architecture

In RMII mode, reference clock and TX/RX clock can be from CRU or external OSC as following figure.
The mux select is CRU_CLKSEL_CON43[14].

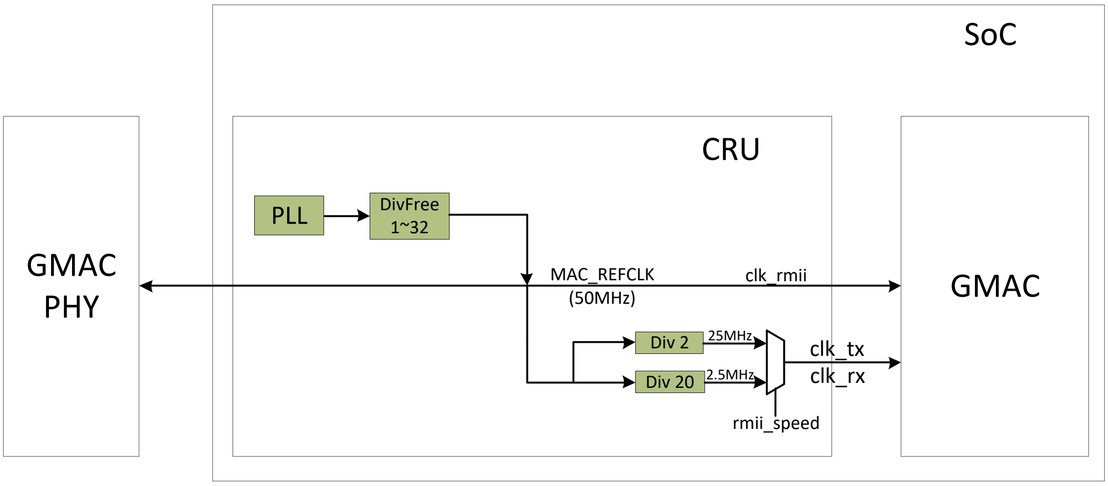


Fig. 23-13 RMII clock architecture when clock source from CRU

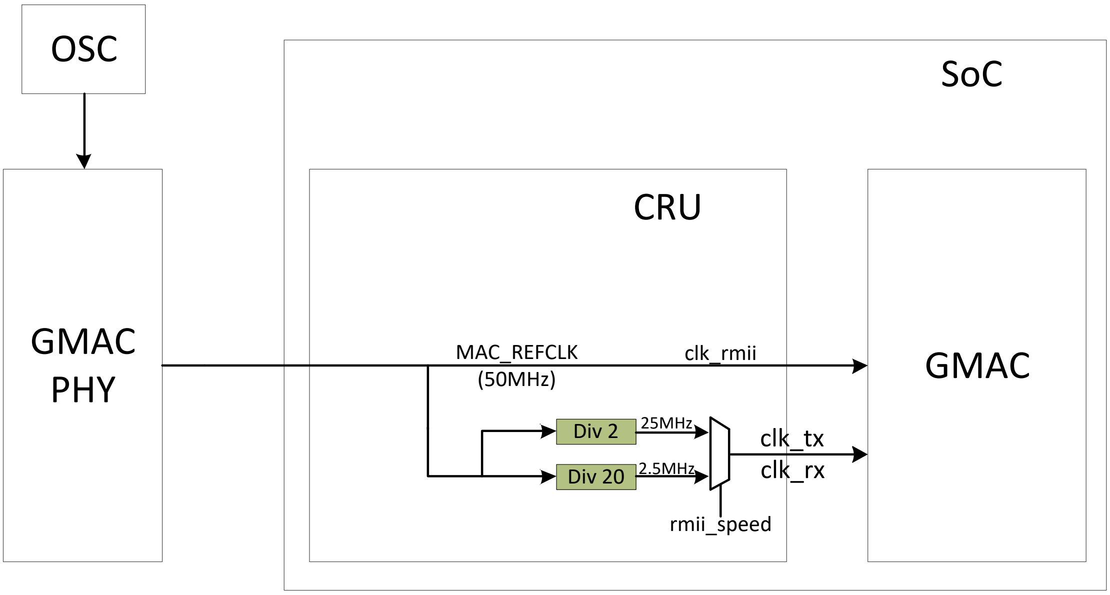
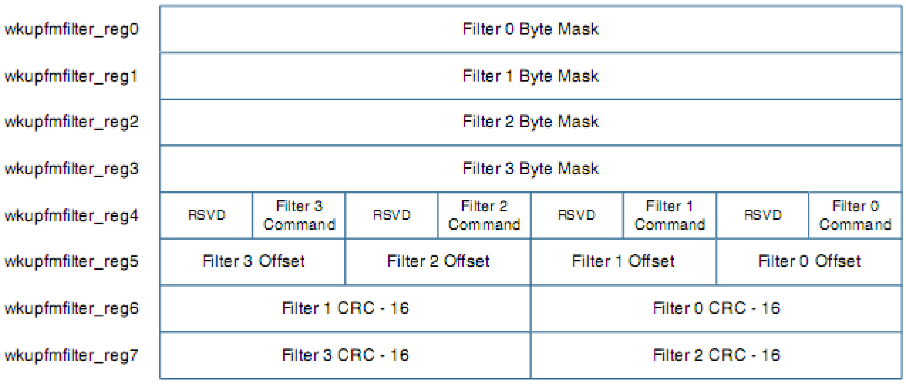


Fig. 23-14 RMII clock architecture when clock source from external OSC

23.6.6 Remote Wake-Up Frame Filter Register

The register wkupfmfilter_reg, address (028H), loads the Wake-up Frame Filter register. To load values in a Wake-up Frame Filter register, the entire register (wkupfmfilter_reg) must be written. The wkupfmfilter_reg register is loaded by sequentially loading the eight register values in address (028) for wkupfmfilter_reg0, wkupfmfilter_reg1, ..., wkupfmfilter_reg7, respectively. Wkupfmfilter_reg is read in the same way.
The internal counter to access the appropriate wkupfmfilter_reg is incremented when lane3 (or lane 0 in big-endian) is accessed by the CPU. This should be kept in mind if you are accessing these registers in byte or half-word mode.



wkupfmfilter_reg0	Filter 0 Byte Mask							
wkupfmfilter_reg1	Filter 1 Byte Mask							
wkupfmfilter_reg2	Filter 2 Byte Mask							
wkupfmfilter_reg3	Filter 3 Byte Mask							
wkupfmfilter_reg4	RSVD	Filter 3 Command	RSVD	Filter 2 Command	RSVD	Filter 1 Command	RSVD	Filter 0 Command
wkupfmfilter_reg5	Filter 3 Offset		Filter 2 Offset		Filter 1 Offset		Filter 0 Offset	
wkupfmfilter_reg6	Filter 1 CRC - 16				Filter 0 CRC - 16			
wkupfmfilter_reg7	Filter 3 CRC - 16				Filter 2 CRC - 16			

Fig. 23-1 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.

23.6.7 System Consideration During Power-Down

MAC neither gates nor stops clocks when Power-Down mode is enabled. Power saving by clock gating must be done outside the core by the CRU. The receive data path must be clocked with clk_rx_i during Power-Down mode, because it is involved in magic packet/wake-on-LAN frame detection. However, the transmit path and the APB path clocks can be gated off during Power-Down mode.

The PMT interrupt is asserted when a valid wake-up frame is received. This interrupt is generated in the clk_rx domain.

The recommended power-down and wake-up sequence is as follows.

1. Disable the Transmit DMA (if applicable) and wait for any previous frame transmissions to complete. These transmissions can be detected when Transmit Interrupt (TI - Register MAC_STATUS[0]) is received.
2. Disable the MAC transmitter and MAC receiver by clearing the appropriate bits in the MAC Configuration register.
3. Wait until the Receive DMA empties all the frames from the Rx FIFO (a software timer may be required).
4. Enable Power-Down mode by appropriately configuring the PMT registers.
5. Enable the MAC Receiver and enter Power-Down mode.
6. Gate the APB and transmit clock inputs to the core (and other relevant clocks in the system) to reduce power and enter Sleep mode.
7. On receiving a valid wake-up frame, the MAC asserts the PMT interrupt signal and exits Power-Down mode.
8. On receiving the interrupt, the system must enable the APB and transmit clock inputs to

the core.
9. Read the register MAC_PMT_CTRL_STA to clear the interrupt, then enable the other modules in the system and resume normal operation.

23.6.8 GRF Register Summary

MAC2IO	
GRF Register	**Register Description**
GRF_MAC_CON0[0]	MACspeed 1'b1: 100-Mbps 1'b0: 10-Mbps
GRF_MAC_CON0[1]	MAC transmit flow control When set high, instructs the MAC to transmit PAUSE Control frames in Full-duplex mode. In Half-duplex mode, the MAC enables the Back-pressure function until this signal is made low again
GRF_MAC_CON0[4:2]	PHY interface select 3'b001: RGMII(useless) 3'b100: RMII All others: Reserved
CRU_CLKSEL_CON43[14]	rmii_extclk_sel 1’b1:from CRU 1’b0:from IO
CRU_CLKSEL_CON43[15]	rmii_clk_sel 1’b1:100M 1’b0:10M

Chapter 24 WatchDog

24.1 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 .The WDT would generate interrupt or reset signal when it's counter reaches zero, then a reset controller would reset the system. there are a Non-secure WDT(WDT_NS) and a Secure WDT(WDT_S);
WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
  - Generate a system reset
  - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period
- Support two WTD, one is used for non-secure application, the other is used for secure application

24.2 Block Diagram

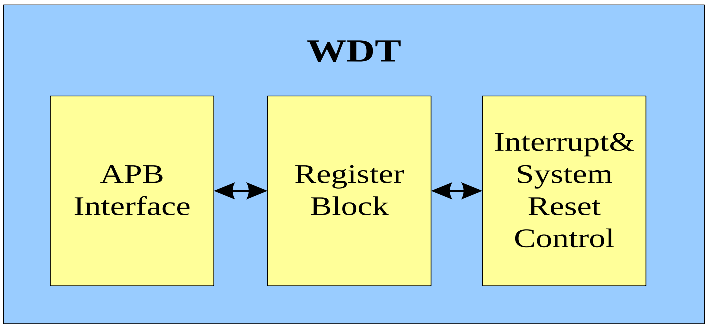


Fig. 24-1 WDT block diagram

Block Descriptions:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- Register Block

A register block that read coherence for the current count register.

- Interrupt & system reset control

An interrupt/system reset generation block is comprised of a decrementing counter and control logic.

24.3 Function Description

24.3.1 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.

24.4 Register Description

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

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x0000ffff	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

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

24.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. 000: 2 pclk cycles 001: 4 pclk cycles 010: 8 pclk cycles 011: 16 pclk cycles 100: 32 pclk cycles 101: 64 pclk cycles 110: 128 pclk cycles 111: 256 pclk cycles

Bit	Attr	Reset Value	Description
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 0: Generate a system reset. 1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset
0	RW	0x0	wdt_en WDT enable: 0: WDT disabled; 1: WDT enabled

WDT_TORR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 0000: 0x0000ffff 0001: 0x0001ffff 0010: 0x0003ffff 0011: 0x0007ffff 0100: 0x000fffff 0101: 0x001fffff 0110: 0x003fffff 0111: 0x007fffff 1000: 0x00ffffff 1001: 0x01ffffff 1010: 0x03ffffff 1011: 0x07ffffff 1100: 0x0fffffff 1101: 0x1fffffff 1110: 0x3fffffff 1111: 0x7fffffff

WDT_CCVR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt Current counter value. This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	W1C	0x00	cnt_restart Counter restart. This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero

WDT_STAT
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1: Interrupt is active regardless of polarity; 0: Interrupt is inactive

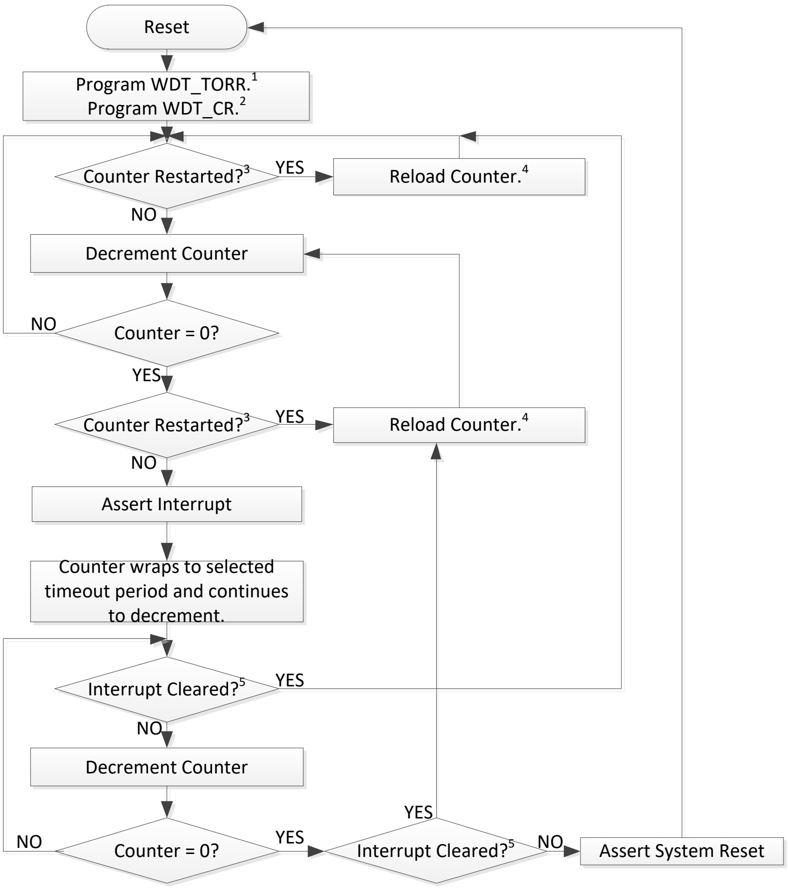
WDT_EOI
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter

24.5 Application Notes

24.5.1 Programming sequence

The following figure show the operation flow chart (Response mode=1).



1. Select required timeout period.
2. Set reset pulse length, response mode, and enable WDT.
3. Write 0x76 to WDT_CRR.
4. Starts back to selected timeout period.
5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 24-2 WDT Operation Flow

Chapter 25 SAR-ADC

25.1 Overview

The ADC is a 6-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference. It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 1MSPS with 13MHz A/D converter clock.

25.2 Block Diagram

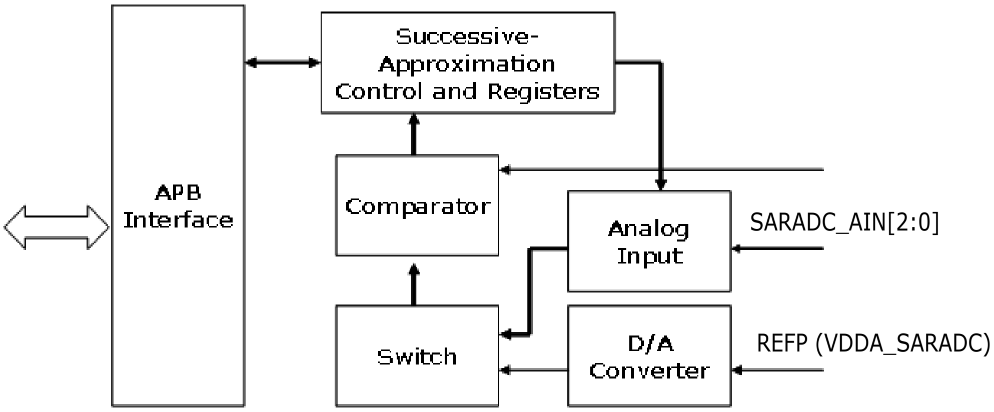


Fig. 25-1 RK3308 SAR-ADC block diagram

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.

25.3 Function Description

25.3.1 APB Interface

In RK3308, 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.

25.4 Register description

25.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	The data after A/D conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D converter.

Name	Offset	Size	Reset Value	Description
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	Delay between power up and start command.

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

25.4.2 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 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

Bit	Attr	Reset Value	Description
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]) 3'b011: Input source 3 (SARADC_AIN[3]) 3'b100: Input source 4 (SARADC_AIN[4]) 3'b101: Input source 5 (SARADC_AIN[5]) Others: Reserved

SARADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up

25.5 Timing Diagram

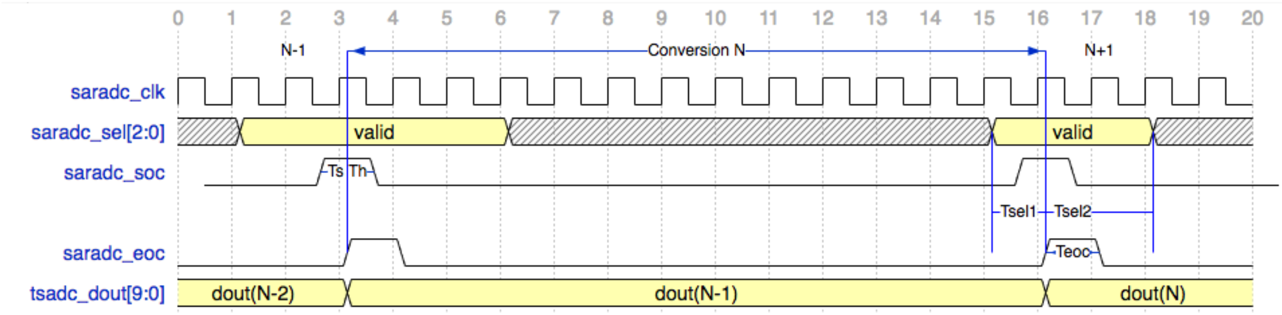


Fig. 25-2 SAR-ADC timing diagram in single-sample conversion mode

The following table shows the detailed value for timing parameters in the above diagram.

Table 25-1 RK3308 SAR-ADC timing parameters list

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Condition						
Analog Supply	AVDD		1.62	1.8	1.98	V
Digital Supply	VDD		0.81	0.9	0.99	V
Junction Temperature	Tj		-40		125	℃
Saradc Performance						
Resolution				10		bit
Effective Number of Bit	ENOB			9		bit
Differential Nonlinearity	DNL		-1		1	LSB
Intergral Nonlinearity	INL		-2		2	LSB
Input Voltage Range	Vin		0		1	AVDD

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Input Capacitance	Cin			10		pF
Sampling Rate	fs				1	MS/s
Spurious Free Dynamic Range	SFDR	fs=1MS/s fout = 1.17KHz		61		dB
Signal to Noise and Harmonic Ratio	SNDR			56		dB
Timing Characteristic						
Clock Frequency	fCLK				13	MHz
Clock Period	tCLK		75			ns
Clock Duty Cycle			45		55	%
Conversion Time			13			tCLK
Setup Time of soc signal	Ts			0.5		tCLK
Hold Time of soc signal	Th			0.5		tCLK
Time Interval between Transition of sel[2:0] and Rising Edge of 1st clock	Tsel1		1			tCLK
Rising Edge of 1st clock and Time Interval between Transition of sel[2:0]	Tsel2		2			tCLK
High Level Time of eoc signal	Teoc		1			tCLK
Power Consumption						
Analog Supply Current	I_{AVDD}	fs=1MS/s		450		uA
		Power Down		1		uA
Digital Supply Current	I_{VDD}	fs=1MS/s		50		uA
		Power Down		1		uA

25.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down adc converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed
- Read the conversion result from SARADC_DATA[9:0]
- Note: The A/D converter was designed to operate at maximum 1MHZ.

Chapter 26 Temperature-Sensor ADC(TS-ADC)

26.1 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 over the defined high temperature value in a period of time, an interrupt is generated to the processor down-measures taken; if the temperature over the defined SHUT temperature value, the resulting TSHUT gave CRU module, let it reset the entire chip, or via GPIO give PMIC.

TS-ADC Controller supports the following features:

- Support User-Defined Mode and Automatic Mode
- In User-Defined Mode, start_of_conversion can be controlled completely by software, and also can be generated by hardware.
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt can be configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support to 2 channel TS-ADC, the temperature criteria 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
- -40~125°C temperature range and 5°C temperature resolution
- 12-bit SARADC up to 50KS/s sampling rate

26.2 Block Diagram

TS-ADC controller comprises with:

- APB Interface
- TS-ADC control logic

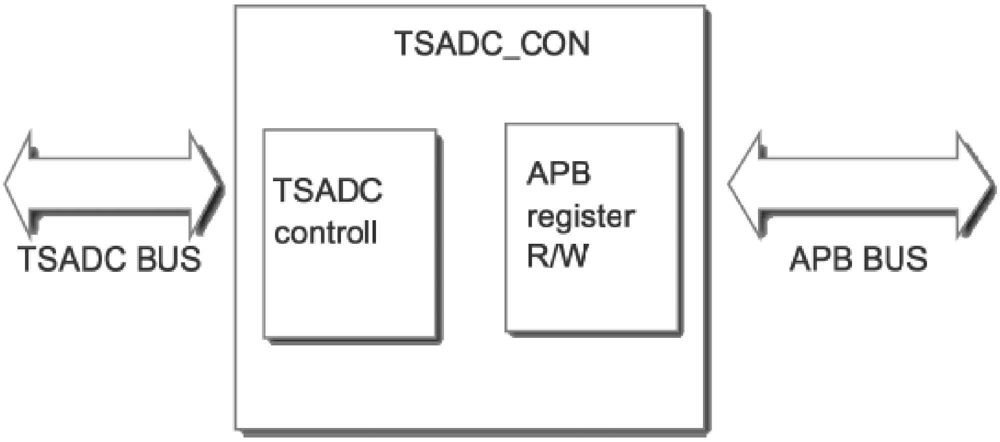


Fig. 26-1 TS-ADC Controller Block Diagram

26.3 Function Description

26.3.1 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.

26.3.2 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.

26.4 Register description

26.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TSADC_USER_CON	0x0000	W	0x00000200	The control register of A/D converter.
TSADC_AUTO_CON	0x0004	W	0x00000000	TSADC auto mode control register
TSADC_INT_EN	0x0008	W	0x00000000	TSADC interrupts enable control
TSADC_INT_PD	0x000c	W	0x00000000	TSADC interrupts status
TSADC_DATA0	0x0020	W	0x00000000	This register contains the data from CH0 after A/D conversion.
TSADC_DATA1	0x0024	W	0x00000000	This register contains the data from CH1 after A/D conversion.
TSADC_COMP0_INT	0x0030	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMP1_INT	0x0034	W	0x00000000	TSADC high temperature level for source 1
TSADC_COMP0_SHUT	0x0040	W	0x00000000	TSADC shut temperature level for source 0
TSADC_COMP1_SHUT	0x0044	W	0x00000000	TSADC shut temperature level for source 1
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	High temperature debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	Shut temperature debounce
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	TSADC auto access period
TSADC_AUTO_PERIOD_HT	0x006c	W	0x00010000	TSADC auto access period when temperature is high
TSADC_COMP0_LOW_INT	0x0080	W	0x00000000	TSADC low temperature level for source 0
TSADC_COMP1_LOW_INT	0x0084	W	0x00000000	TSADC low temperature level for source 1

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

26.4.2 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) 1'b0: ADC stop 1'b1: Conversion in progress
11:6	RW	0x08	inter_pd_soc interleave between power down and start of conversion
5	RW	0x0	start When software write 1 to this bit , start_of_conversion will be assert. This bit will be cleared after TSADC access finishing. When TSADC_USER_CON[4] = 1'b1 take effect.
4	RW	0x0	start_mode start mode. 1'b0: tsadc controller will asert start_of_conversion after "inter_pd_soc" cycles. 1'b1: the start_of_conversion will be controlled by TSADC_USER_CON[5].
3	RW	0x0	adc_power_ctrl ADC power down control bit 1'b0: ADC power down 1'b1: ADC power up and reset
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]) Others: Reserved

TSADC_AUTO_CON
Address: Operational Base + offset (0x0004)
TSADC auto mode control register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RW	0x0	last_tshut_2cru last_tshut_2cru for cru first/second reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
24	RW	0x0	last_tshut_2gpio last_tshut_2gpio for hardware reset TSHUT status. This bit will set to 1 when tshut is valid, and only be cleared when application write 1 to it. This bit will not be cleared by system reset.
23:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17	RO	0x0	sample_dly_sel 1'b0: AUTO_PERIOD is used 1'b1: AUTO_PERIOD_HT is used
16	RO	0x0	auto_status 1'b0: auto mode stop 1'b1: auto mode in progress
15:14	RO	0x0	reserved
13	RW	0x0	src1_lt_en 1'b0: do not care low temperature of source 1 1'b1: enable the low temperature monitor of source 1
12	RW	0x0	src0_lt_en 1'b0: do not care low temperature of source 0 1'b1: enable the low temperature monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_prolarity 1'b0: low active 1'b1: high active
7:6	RO	0x0	reserved
5	RW	0x0	src1_en 1'b0: do not care the temperature of source 1 1'b1: if the temperature of source 1 is too high, TSHUT will be valid
4	RW	0x0	src0_en 1'b0: do not care the temperature of source 0 1'b1: if the temperature of source 0 is too high, TSHUT will be valid
3:2	RO	0x0	reserved
1	RW	0x0	tsadc_q_sel temperature coefficient 1'b0:use tsadc_q as output 1'b1:use(4096 - tsadc_q) as output
0	RW	0x0	auto_en 1'b0: TSADC controller works at user-define mode 1'b1: TSADC controller works at auto mode

TSADC_INT_EN
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_en eoc_Interrupt enable. eoc_interrupt enable in user defined mode 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13	RW	0x0	lt_inten_src1 low temperature interrupt enable for src1 1'b0: disable 1'b1: enable
12	RW	0x0	lt_inten_src0 low temperature interrupt enable for src0 1'b0: disable 1'b1: enable
11:10	RO	0x0	reserved
9	RW	0x0	tshut_2cru_en_src1 1'b0: TSHUT output to cru disabled. TSHUT output will always keep low 1'b1: TSHUT output works
8	RW	0x0	tshut_2cru_en_src0 1'b0: TSHUT output to cru disabled. TSHUT output will always keep low 1'b1: TSHUT output works
7:6	RO	0x0	reserved
5	RW	0x0	tshut_2gpio_en_src1 1'b0: TSHUT output to gpio disabled. TSHUT output will always keep low 1'b1: TSHUT output works
4	RW	0x0	tshut_2gpio_en_src0 1'b0: TSHUT output to gpio disabled. TSHUT output will always keep low 1'b1: TSHUT output works
3:2	RO	0x0	reserved
1	RW	0x0	ht_inten_src1 high temperature interrupt enable for src1 1'b0: disable 1'b1: enable
0	RW	0x0	ht_inten_src0 high temperature interrupt enable for src0 1'b0: disable 1'b1: enable

TSADC_INT_PD

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved

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

TSADC_DATA0
Address: Operational Base + offset (0x0020)
This register contains the data after A/D Conversion.

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

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

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

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the channel 1 last conversion (DOUT[11:0]).

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

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

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

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

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

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

TSADC_COMP1_SHUT

Address: Operational Base + offset (0x0044)
TSADC high temperature level for source 1

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

TSADC_HIGHT_INT_DEBOUNCE

Address: Operational Base + offset (0x0060)
High temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_INT for "debounce" times.

TSADC_HIGHT_TSHUT_DEBOUNCE

Address: Operational Base + offset (0x0064)
Shut temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_SHUT for "debounce" times.

TSADC_AUTO_PERIOD

Address: Operational Base + offset (0x0068)
TSADC auto access period

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period When auto mode is enabled, this register controls the interleave between every two accessing of TSADC.

TSADC_AUTO_PERIOD_HT

Address: Operational Base + offset (0x006c)
TSADC auto access period when temperature is high

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.

TSADC_COMP0_LOW_INT
Address: Operational Base + offset (0x0080)
TSADC low temperature level for source 0

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

TSADC_COMP1_LOW_INT
Address: Operational Base + offset (0x0084)
TSADC low temperature level for source 1

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

26.5 Application Notes

26.5.1 Channel Select

The system has two Temperature Sensors, channel 0 is for logic power domain and channel 1 is for core power domain.

26.5.2 Single-sample conversion

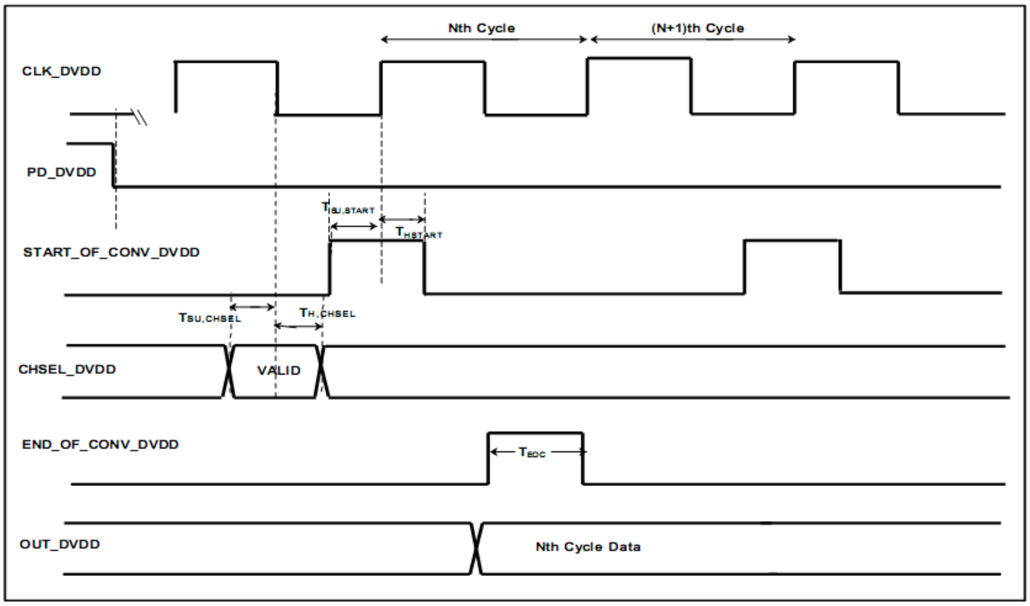


Fig. 26-2 the start flow to enable the sensor and adc

26.5.3 Temperature-to-code mapping

Table 26-1 Temperature Code Mapping

temp(℃)	Code tsadc_q_sel=0	Code tsadc_q_sel=1
-40	3800	296
-35	3792	304
-30	3783	313
-25	3774	322
-20	3765	331
-15	3756	340
-10	3747	349
-5	3737	359
0	3728	368
5	3718	378
10	3708	388
15	3698	398
20	3688	408
25	3678	418
30	3667	429
35	3656	440
40	3645	451
45	3634	462
50	3623	473
55	3611	485
60	3600	496
65	3588	508
70	3575	521
75	3563	533
80	3550	546
85	3537	559
90	3524	572
95	3510	586
100	3496	600
105	3482	614
110	3467	629
115	3452	644
120	3437	659
125	3421	675

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.
RK3308 tsadc_q_sel must be configured as 1'b1.

26.5.4 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 two channel temperature from TSADC_DATAn (n=0,1).

26.5.5 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 interval between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.
- Set TSADC_COMPn_INT(n=0,1), configure the high temperature level, if TSADC output is larger than the value, means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0,1), configure the super high temperature level, if TSADC output is larger 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 27 Timer

27.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. There are 2 groups of timer (TIMER_6CH_0 & TIMER_6CH_1) and each has 6 counting up timers. The group of TIMER_6CH_1 is dedicated to secure mode, its clock and reset configuration registers are in SGRF unit.

Timer supports the following features:

- Two operation modes: free-running and user-defined count.
- TIMER_6CH_1's timer5 defined as A35 Generic Timer.

27.2 Block Diagram

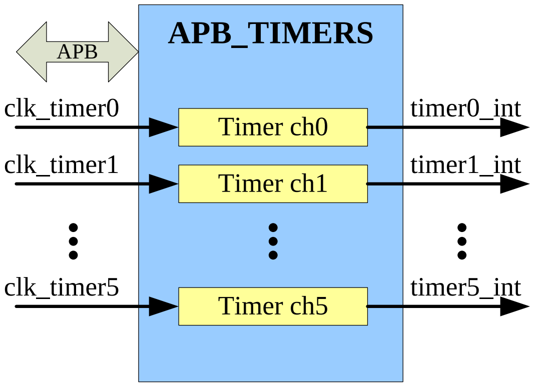


Fig. 27-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (include six programmable timer channels) that in the bus subsystem.

27.3 Function Description

27.3.1 Timer clock

Timers are in the pd_peri_wrapper subsystem. The timer clock is 24MHz OSC.

27.3.2 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).

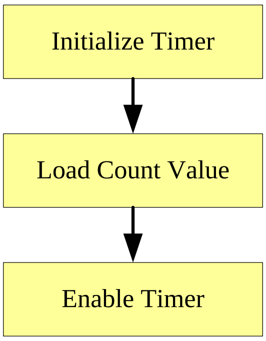


Fig. 27-2 Timer Usage Flow

27.3.3 Loading a timer count value

The initial value for each timer is zero. The count register will count up to the value loaded in the register TIMERn_LOAD_COUNT1 and TIMERn_ LOAD_COUNT0. Two events can cause a timer to load zero:

- Timer is enabled after reset or disabled.
- Timer counts up to the value stored in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0, when timer is configured into free-running mode.

27.3.4 Timer mode selection

- User-defined count mode – Timer loads zero (for incremental timers) as initial value. When the timer counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0, it will not automatically reload the counting register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- Free-running mode – Timer loads zero (for incremental timers) as initial value. Timer will automatically reload the counting register, when timer counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0.

27.4 Register Description

27.4.1 Internal Address Mapping

The table below shows the base address for each timer internal.

Table 27-1 Timer Internal Address Mapping Table

Name	Timern Base Address Offset
Timer0	0x00
Timer1	0x20
Timer2	0x40
Timer3	0x60
Timer4	0x80
Timer5	0xA0

27.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_LOAD_COUNT0	0x0000	W	0x00000000	Timern Load Count Register 0. Low 32 bits Value to be loaded into Timer n.
TIMER_TIMERn_LOAD_COUNT1	0x0004	W	0x00000000	Timern Load Count Register 1. High 32 bits Value to be loaded into Timer n.

Name	Offset	Size	Reset Value	Description
TIMER_TIMERn_CURRENT_VALUE0	0x0008	W	0x00000000	Timern Current Value Register 0. Low 32 bits of Current Value of Timer n.
TIMER_TIMERn_CURRENT_VALUE1	0x000c	W	0x00000000	Timern Current Value Register 1. High 32 bits of Current Value of Timer n.
TIMER_TIMERn_CONTROLREG	0x0010	W	0x00000000	Timern Control Register
TIMER_TIMERn_INTSTATUS	0x0018	W	0x00000000	Timern Interrupt Status Register

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

27.4.3 Detail Register Description

TIMER_TIMERn_LOAD_COUNT0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_0 Low 32 bits Value to be loaded into Timer n. This is the value from which counting commences

TIMER_TIMERn_LOAD_COUNT1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load_count_1 High 32 bits Value to be loaded into Timer n. This is the value from which counting commences.

TIMER_TIMERn_CURRENT_VALUE0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value0 Low 32 bits of Current Value of Timer n.

TIMER_TIMERn_CURRENT_VALUE1

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	timern_current_value1 High 32 bits of Current Value of Timer n.

TIMER_TIMERn_CONTROLREG

Address: Operational Base + offset (0x0010)

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

Bit	Attr	Reset Value	Description
2	RW	0x0	timer_int_mask Timer interrupt mask 1'b0: mask 1'b1: not mask
1	RW	0x0	timer_mode Timer mode 1'b0: free-running mode 1'b1: user-defined count mode
0	RW	0x0	timer_en Timer enable 1'b0: disable 1'b1: enable

TIMER_TIMERn_INTSTATUS

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	timern_int This register contains the interrupt status for Timer n. Write 1 to this register will clear the interrupt.

27.5 Application Notes

In the chip, the timer_clk is from 24MHz OSC, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG (0≤n≤5)), the timer en output signal is de-asserted, and timer_clk will stop. When user enables the timer, the timer_en signal is asserted and timer_clk will start running.

The application is only allowed to re-config registers when timer_en is low.

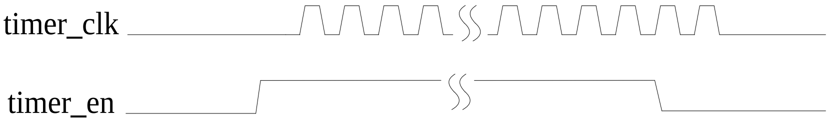


Fig. 27-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 28 SPDIF Transmitter

28.1 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.

It supports following features:

- Support one internal 32-bit wide and 32-location deep sample data buffer
- Support two 16-bit audio data store together in one 32-bit wide location
- Support AHB bus interface
- Support biphase format stereo audio data output
- Support DMA handshake interface and configurable DMA water level
- Support sample data buffer empty and block terminate interrupt
- Support combine interrupt output
- Support 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 48, 44.1, 32kHz sample rate
- Support 16, 20, 24 bits audio data transfer

28.2 Block Diagram

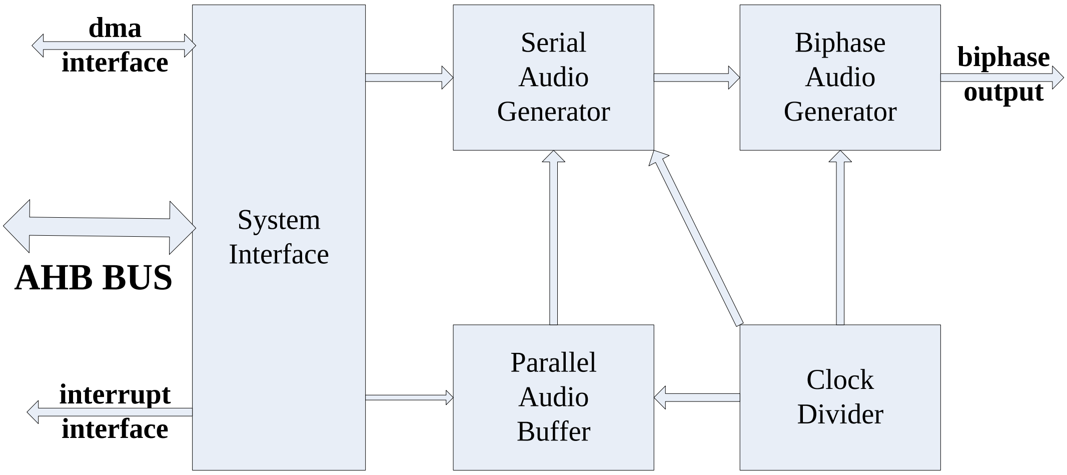


Fig. 28-1 SPDIF transmitter Block Diagram

The SPDIF is composed by:

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshaking interface.

Clock Divider

The clock divider implements clock generation function. It divides the the source clock MCLK to generate the working clock used for the digital audio data transformation and transmission.

Parallel Audio Buffer
The parallel audio buffer stores the audio data to be transmitted. The size of the FIFO is 32bits x 32.

Serial Audio Converter
The serial audio converter converts the parallel audio data from the parallel audio buffer to the 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.

28.3 Function description

28.3.1 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.

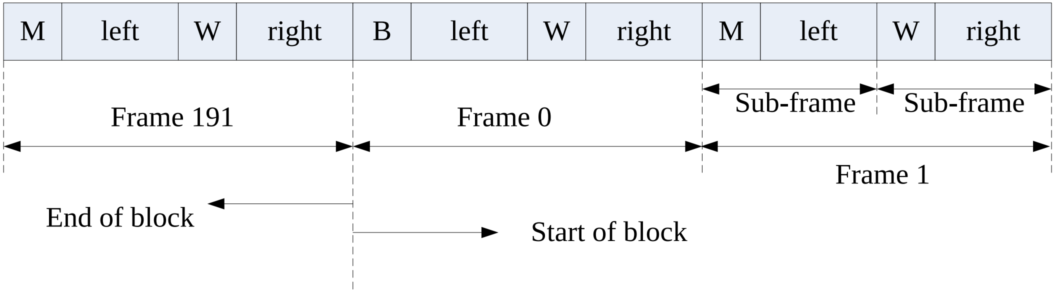


Fig. 28-2 SPDIF Frame Format

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).

28.3.2 Sub-frame Format

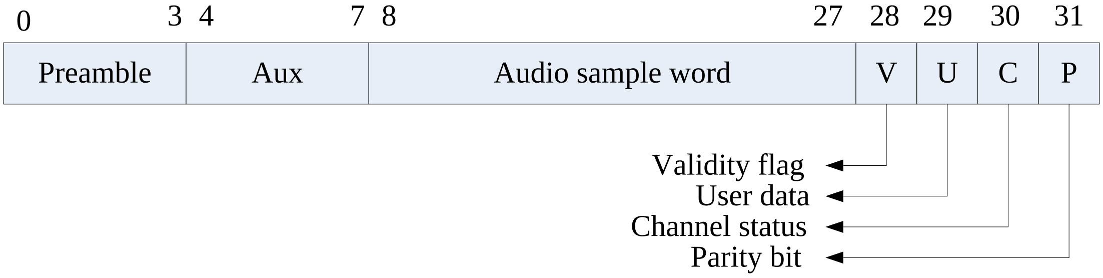


Fig. 28-3 SPDIF Sub-frame Format

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.

28.3.3 Channel Coding

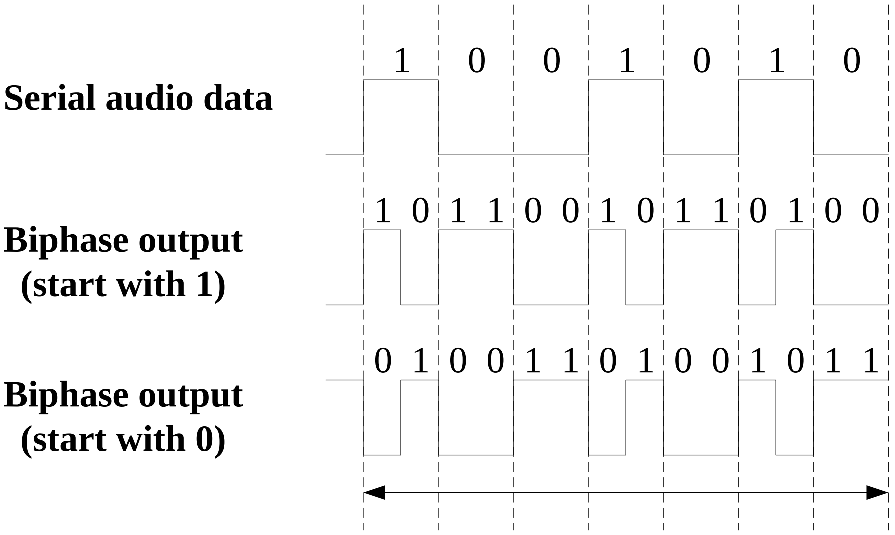


Fig. 28-4 SPDIF Channel Coding

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'.

28.3.4 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.

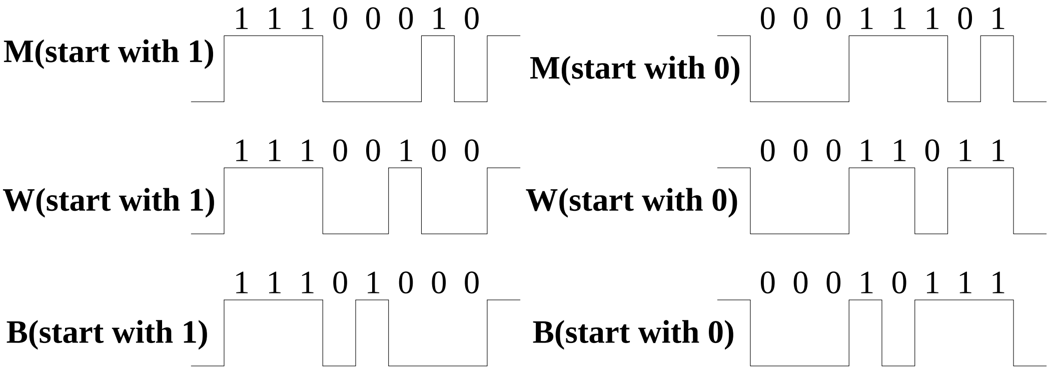


Fig. 28-5 SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

28.4 Register Description

28.4.1 Registers 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	0x00d0	W	0x00000000	Channel Burst Info Register
SPDIF_REPETTION	0x0104	W	0x00000000	Channel Repetition Register
SPDIF_BURTSINFO_SHD	0x0108	W	0x00000000	Shadow Channel Burst Info Register
SPDIF_REPETTION_SHD	0x010c	W	0x00000000	Shadow Channel Repetition Register
SPDIF_USRDR_SHDn	0x0190	W	0x00000000	Shadow User Data Register n

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

28.4.2 Detail Register Description

SPDIF_CFGR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MCD Fmclk/Fsdo This parameter can be calculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	PCMTYPE 1'b0: linear PCM 1'b1: non-linear PCM
7	WO	0x0	CLR Write 1 to clear MCLK domain logic. Read return zero.
6	RW	0x0	CSE 1'b0: disable 1'b1: enable The bit should be set to 1 when the channel conveys non-linear PCM
5	RW	0x0	UDE 1'b0: disable 1'b1: enable
4	RW	0x0	VFE 1'b0: disable 1'b1: enable
3	RW	0x0	ADJ 1'b0: Right justified 1'b1: Left justified
2	RW	0x0	HWT 1'b0: disable 1'b1: enable It is valid only when the valid data width is 16bit.
1:0	RW	0x0	VDW 2'b00: 16bit 2'b01: 20bit 2'b10: 24bit 2'b11: reserved The valid data width is 16bit only for non-linear PCM

SPDIF_SDBLR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	SDBLR Contains the number of valid data entries in the sample data buffer.

SPDIF_DMACR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	TDE 1'b0: Transmit DMA disabled 1'b1: Transmit DMA enabled
4:0	RW	0x00	TDL This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the Sample Date Buffer is equal to or below this field value

SPDIF_INTCR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	W1C	0x0	UDTIC Write '1' to clear the user data interrupt.
16	W1C	0x0	BTTIC Write 1 to clear the interrupt.
15:10	RO	0x0	reserved
9:5	RW	0x00	SDBT Sample Date Buffer Threshold for empty interrupt
4	RW	0x0	SDBEIE 1'b0: disable 1'b1: enable
3	RW	0x0	BTTIE 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. 1'b0: disable 1'b1: enable
2	RW	0x0	UDTIE 1'b0: disable 1'b1: 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)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	SDBEIS 1'b0: inactive 1'b1: active
3	RW	0x0	BTTIS 1'b0: inactive 1'b1: active
2	RW	0x0	UDTIS 1'b0: inactive 1'b1: active
1:0	RO	0x0	reserved

SPDIF_XFER

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	XFER Transfer Start Register

SPDIF_SMPDR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SMPDR Sample Data Register

SPDIF_VLDFRn

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	VLDFR_SUB_1 Validity Flag Register 0
15:0	RW	0x0000	VLDFR_SUB_0 Validity Flag For Subframe 0

SPDIF_USRDRn

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	USR_SUB_1 User Data Bit for Subframe 1
15:0	RW	0x0000	USR_SUB_0 User Data Bit for Subframe 0

SPDIF_CHNSRn

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CHNSR_SUB_1 Channel Status Bit for Subframe 1
15:0	RW	0x0000	CHNSR_SUB_0 Channel Status Bit for Subframe 0

SPDIF_BURTSINFO

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PD Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RW	0x0	BSNUM This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RW	0x00	DATAINFO This field gives the data-type-dependent info
7	RW	0x0	ERRFLAG 1'b0: indicates a valid burst-payload 1'b1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RW	0x00	DATATYPE 7'b0000000: null data 7'b0000001: AC-3 data 7'b0000011: Pause data 7'b0000100: MPEG-1 layer 1 data 7'b0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 7'b0000110: MPEG-2 data with extension 7'b0000111: MPEG-2 AAC 7'b0001000: MPEG-2, layer-1 low sampling frequency 7'b0001001: MPEG-2, layer-2 low sampling frequency 7'b0001010: MPEG-2, layer-3 low sampling frequency 7'b0001011: DTS type I 7'b0001100: DTS type II 7'b0001101: DTS type III 7'b0001110: ATRAC 7'b0001111: ATRAC 2/3 7'b0010000: ATRAC-X 7'b0010001: DTS type IV 7'b0010010: WMA professional type I 7'b0110010: WMA professional type II 7'b1010010: WMA professional type III 7'b1110010: WMA professional type IV 7'b0010011: MPEG-2 AAC low sampling frequency 7'b0110011: MPEG-2 AAC low sampling frequency 7'b1010011: MPEG-2 AAC low sampling frequency 7'b1110011: MPEG-2 AAC low sampling frequency 7'b0010100: MPEG-4 AAC 7'b0110100: MPEG-4 AAC 7'b1010100: MPEG-4 AAC 7'b1110100: MPEG-4 AAC 7'b0010101: Enhanced AC-3 7'b0010110: MAT others: reserved

SPDIF_REPETTION

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	REPETTION This define the repetition period when the channel conveys non-linear PCM

SPDIF_BURTSINFO_SHD

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	PD Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RO	0x0	BSNUM This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RO	0x00	DATAINFO This field gives the data-type-dependent info
7	RO	0x0	ERRFLAG 1'b0: indicates a valid burst-payload 1'b1: indicates that the burst-payload may contain errors
6:0	RO	0x00	DATATYPE 7'b0000000: null data 7'b0000001: AC-3 data 7'b0000011: Pause data 7'b0000100: MPEG-1 layer 1 data 7'b0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 7'b0000110: MPEG-2 data with extension 7'b0000111: MPEG-2 AAC 7'b0001000: MPEG-2, layer-1 low sampling frequency 7'b0001001: MPEG-2, layer-2 low sampling frequency 7'b0001010: MPEG-2, layer-3 low sampling frequency 7'b0001011: DTS type I 7'b0001100: DTS type II 7'b0001101: DTS type III 7'b0001110: ATRAC 7'b0001111: ATRAC 2/3 7'b0010000: ATRAC-X 7'b0010001: DTS type IV 7'b0010010: WMA professional type I 7'b0110010: WMA professional type II 7'b1010010: WMA professional type III 7'b1110010: WMA professional type IV 7'b0010011: MPEG-2 AAC low sampling frequency 7'b0110011: MPEG-2 AAC low sampling frequency 7'b1010011: MPEG-2 AAC low sampling frequency 7'b1110011: MPEG-2 AAC low sampling frequency 7'b0010100: MPEG-4 AAC 7'b0110100: MPEG-4 AAC 7'b1010100: MPEG-4 AAC 7'b1110100: MPEG-4 AAC 7'b0010101: Enhanced AC-3 7'b0010110: MAT others: reserved

SPDIF_REPETTION_SHD

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	REPETTION This register provides the repetition of the bitstream when channel conveys non-linear PCM. In the design, it defines the length between 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)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	USR_SUB_1 User Data Bit for Subframe 1
15:0	RO	0x0000	USR_SUB_0 User Data Bit for Subframe 0

28.5 Interface Description

Table 28-1 spdif transmitter interface

Module Pin	Dir	PIN Name	IOMUX Setting
spdif_tx	O	GPIO0_C1/SPDIF_TX	GRF_GPIO0C_IOMUX[3:2]=2’b01

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

28.6 Application Notes

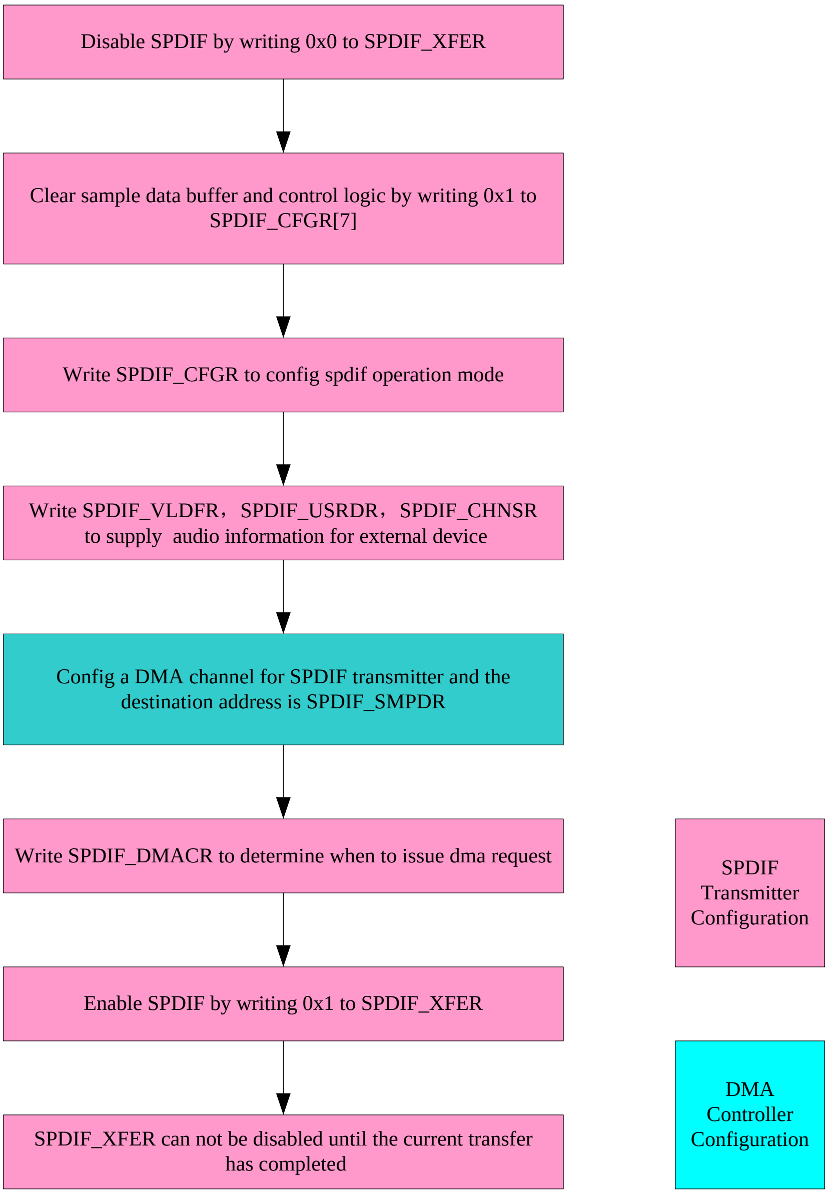

Disable SPDIF by writing 0x0 to SPDIF_XFER
Clear sample data buffer and control logic by writing 0x1 to SPDIF_CFGR[7]
Write SPDIF_CFGR to config spdif operation mode
Write SPDIF_VLDFR，SPDIF_USRDR，SPDIF_CHNSR to supply audio information for external device
Config a DMA channel for SPDIF transmitter and 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. 28-6 SPDIF transmitter operation flow chart

The above figure shows the operation flow of SPDIF operation. Note that the configuration register can be written only when the transfer is stopped.

Chapter 29 SPDIF Receiver

29.1 Overview

The SPDIF receiver 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 receive 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 receive signals sampled at 32 kHz or 44.1 kHz.

When used in a consumer application, the interface is primarily intended to receive stereophonic programmes, with a resolution of up to 24 bits per sample.

When used for other purposes, the interface is primarily intended to receive audio data coded other than as linear PCM coded audio samples. Provision is also made to allow the interface to receive data related to computer software or signals coded using non-linear PCM. The format specification for these applications is not part of this standard.

It supports following features:

- Support AHB Bus Interface
- Support one internal 30-bit wide and 32-location deep FIFO for receiving audio data
- Support combined interrupt output
- Support DMA handshaking interface and configurable DMA water level
- Support liner PCM(IEC60958) and non-liner PCM(IEC61937)
- Support 16~24 bits audio sample length for liner PCM application
- Support 16 bits audio sample length for non-liner PCM application
- Support up to 192kHz sample rate with the corresponding reference clock equal to 192KHz*64*2*10, that is 245.76MHz
- Support the frequency of reference clock is at least 10 times the frequency of the biphase encoding clock, but not more than 256 times
- Support recovering clock and audio data from input bitstream

29.2 Block Diagram

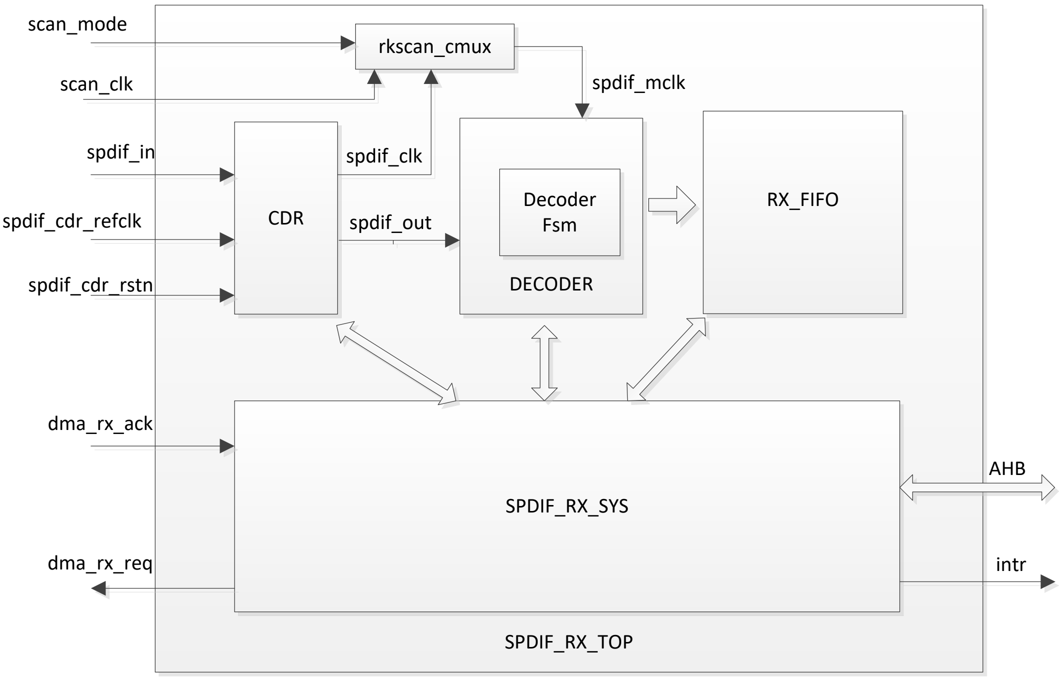


Fig. 29-1 SPDIF receiver Block Diagram

The SPDIF receiver is composed by:

SPDIF_RX_SYS

The module implements the AHB slave operation. It contains not only control registers of SPDIF receiver inside but also interrupt and DMA handshaking interface.

CDR

The CDR is short for clock and data recovery. Clock and data can be recovered from the input bitstream by using a high frequency reference clock.

DECODER

The clock recovered from the CDR is used as the working clock to decode the input bitstream such as extracting audio data, the corresponding preamble code, V/U/C/P flags and other audio information.

RX_FIFO

The RX FIFO stores the audio data extracted from input bitstream. The size of the FIFO is 30bitsx32.

29.3 Function description

29.3.1 Frame Format

SPDIF receiver follows the same protocol(IEC60958/IEC61937) as SPDIF transmitter. A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of receiving of frames corresponds exactly to the source sampling frequency.

In the 2-channel operation mode, the samples received from both channels are arranged 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.

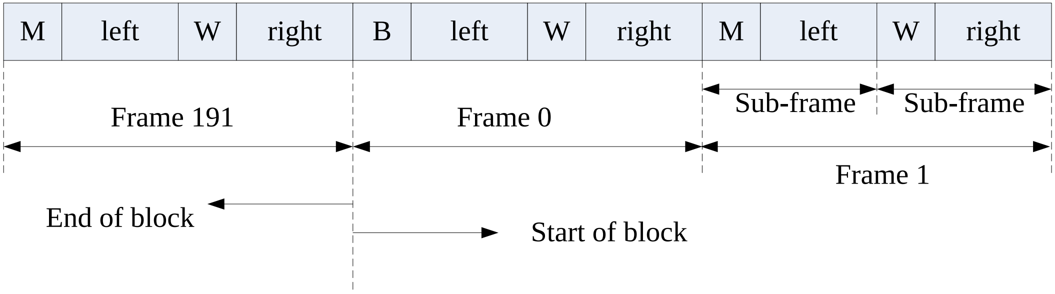


Fig. 29-2 SPDIF Frame Format

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).

29.3.2 Sub-frame Format

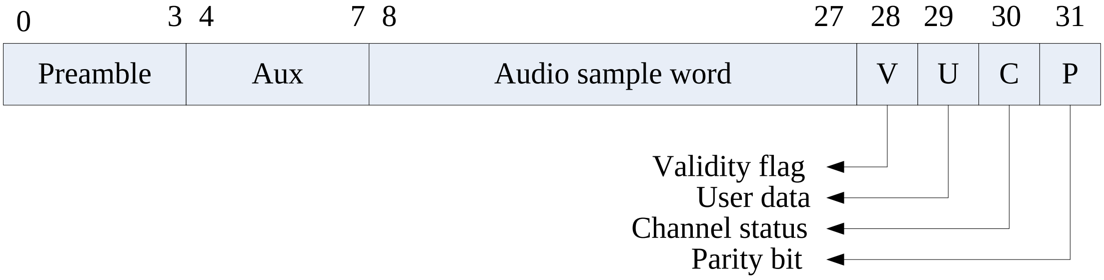


Fig. 29-3 SPDIF Sub-frame Format

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.

29.3.3 Channel Coding

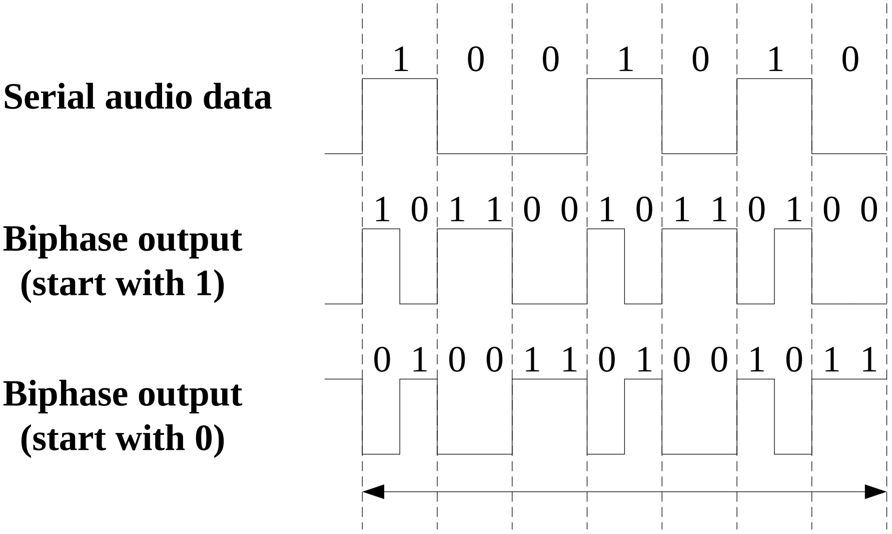


Fig. 29-4 SPDIF Channel Coding

To minimize the direct current component on the transmission/receiving 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'.

29.3.4 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.

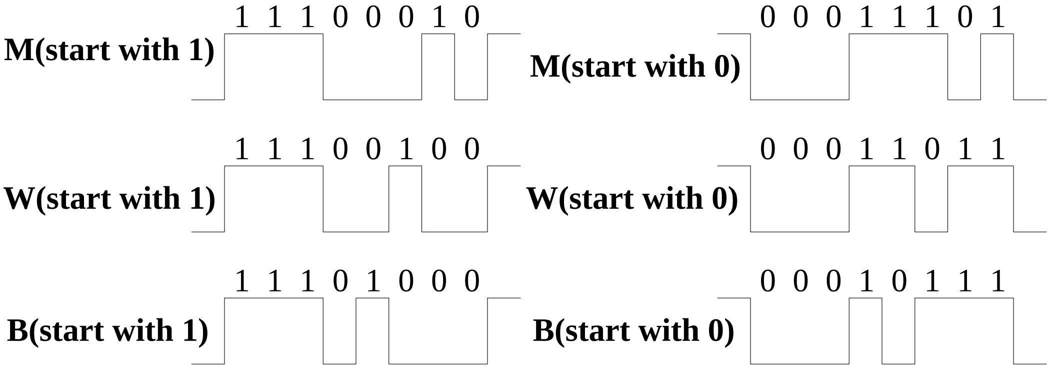


Fig. 29-5 SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

29.4 Register Description

29.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPDIF_RX_VERSION	0x0000	W	0x00000100	Version Register
SPDIF_RX_CFGR	0x0004	W	0x00000000	Transfer Configuration Register
SPDIF_RX_CLR	0x0008	W	0x00000000	Clear Register
SPDIF_RX_CDR	0x000c	W	0x000000a0	Clock And Data Recovery Register
SPDIF_RX_CDRST	0x0010	W	0x000000ff	Clock And Data Recovery Status Register
SPDIF_RX_DMACR	0x0014	W	0x00000000	DMA Control Register
SPDIF_RX_FIFOCTRL	0x0018	W	0x00000000	FIFO Control Register
SPDIF_RX_INTEN	0x001c	W	0x00000000	Interrupt Enable Register
SPDIF_RX_INTMASK	0x0020	W	0x00000000	Interrupt Mask Register
SPDIF_RX_INTSR	0x0024	W	0x00000000	Interrupt Status Register
SPDIF_RX_INTCLR	0x0028	W	0x00000000	Interrupt Clear Register
SPDIF_RX_SMPDR	0x002c	W	0x00000000	Sample Data Register
SPDIF_RX_USRDRn	0x0030	W	0x00000000	User Data Register n(0~11)
SPDIF_RX_CHNSRn	0x0060	W	0x00000000	Channel Status Register n(0~11)
SPDIF_RX_BURTSINFO	0x0100	W	0x00000000	Channel Burst Info Register

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

29.4.2 Detail Register Description

SPDIF_RX_VERSION

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000100	VERSION Version Of SPDIF_RX

SPDIF_RX_CFGR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	TWAD 1'b0: Only audio data transmitted to FIFO. 1'b1: Block start, channel number and VUCP transmitted with audio data to FIFO.
0	RW	0x0	EN 1'b0: disable SPDIF_RX 1'b1: enable SPDIF_RX

SPDIF_RX_CLR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	RXSC This is a software self-cleared bit. Write 1 to clear all receive logic.

SPDIF_RX_CDR
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x0	CS 2'b00: Idle state. CDR is idle and waits for rising edge of spdif input bitstream 2'b01: Detect state. 2'b10: Measurement state. 2'b11: Synchronization state.
8:2	RO	0x0	reserved
1	RW	0x0	AVGSEL Select if the min or average number of cycles from the calculation of spdif_in pulse width. 1'b0: min number of cycles 1'b1: average number of cycles, that is (max+min)/4.
0	RW	0x0	BYPASS Write 1 to enable CDR. Please ensure that this bit is set to 1 before starting receiving data.

SPDIF_RX_CDRST
Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	NOSTRTHR The threshold counter determines the latency between input bitstream idle and exit synchronous status. Make sure that this value is large than 0x3ff.
15:8	RO	0x00	MAXCNT The value indicates the number of reference clock cycles to oversample the maximum pulse width of bitstream.
7:0	RO	0xff	MINCNT The value indicates the number of reference clock cycles to oversample the minimum pulse width of bitstream.

SPDIF_RX_DMACR
Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	RDE 1'b0: Receive DMA disabled 1'b1: Receive DMA enabled
4:0	RW	0x00	RDL This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the Sample Date Buffer is equal to or above this field value + 1.

SPDIF_RX_FIFOCTRL

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	RO	0x00	RFL Contains the number of valid data entries in the receive FIFO.
7:5	RO	0x0	reserved
4:0	RW	0x00	RFT When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.

SPDIF_RX_INTEN

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	UBCIE 1'b0: disable 1'b1: enable
9	RW	0x0	ESYNCIE 1'b0: disable 1'b1: enable Entering synchronous state will generate an interrupt if enabled.
8	RW	0x0	BTEIE 1'b0: disable 1'b1: enable When enabled, an interrupt will be asserted when the block receive is finished if the channel conveys linear PCM or when the repetition period is reached if the channel conveys non-linear PCM.

Bit	Attr	Reset Value	Description
7	RW	0x0	NSYNCIE 1'b0: disable 1'b1: enable A state change from synchronization state to idle state will generate an interrupt if enabled.
6	RW	0x0	BMDEIE 1'b0: disable 1'b1: enable If enabled, an interrupt will be generated if a bi-phase mark decoding error has occurred.
5	RW	0x0	RXOIE 1'b0: disable 1'b1: enable
4	RW	0x0	RXFIE 1'b0: disable 1'b1: enable
3	RW	0x0	NPSPIE 1'b0: disable 1'b1: enable
2	RW	0x0	NVLDIE 1'b0: disable 1'b1: enable
1	RW	0x0	CSCIE 1'b0: disable 1'b1: enable
0	RW	0x0	PEIE 1'b0: disable 1'b1: enable If enabled, an interrupt will be generated if a parity error has occurred.

SPDIF_RX_INTMASK

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	UBCIMSK 1'b0: Don't mask 1'b1: Mask
9	RW	0x0	ESYNCIMSK 1'b0: Don't mask 1'b1: Mask
8	RW	0x0	BTEIMSK 1'b0: Don't mask 1'b1: Mask

Bit	Attr	Reset Value	Description
7	RW	0x0	NSYNCIMSK 1'b0: Don't mask 1'b1: Mask
6	RW	0x0	BMDEIMSK 1'b0: Don't mask 1'b1: Mask
5	RW	0x0	RXOIMSK 1'b0: Don't mask 1'b1: Mask
4	RW	0x0	RXFIMSK 1'b0: Don't mask 1'b1: Mask
3	RW	0x0	NPSPIMSK 1'b0: Don't mask 1'b1: Mask
2	RW	0x0	NVLDIMSK 1'b0: Don't mask 1'b1: Mask
1	RW	0x0	CSCIMSK 1'b0: Don't mask 1'b1: Mask
0	RW	0x0	PEIMSK 1'b0: Don't mask 1'b1: Mask

SPDIF_RX_INTSR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	UBCISR 1'b0: inactive 1'b1: active
9	RW	0x0	ESYNCISR 1'b0: inactive 1'b1: active
8	RW	0x0	BTEISR 1'b0: inactive 1'b1: active
7	RW	0x0	NSYNCISR 1'b0: inactive 1'b1: active
6	RW	0x0	BMDEISR 1'b0: inactive 1'b1: active

Bit	Attr	Reset Value	Description
5	RO	0x0	RXOIS 1'b0: inactive 1'b1: active
4	RO	0x0	RXFIS 1'b0: inactive 1'b1: active
3	RW	0x0	NPSPIS 1'b0: inactive 1'b1: active
2	RO	0x0	NVLDIS 1'b0: inactive 1'b1: active
1	RO	0x0	CSCIS 1'b0: inactive 1'b1: active
0	RO	0x0	PEIS 1'b0: inactive 1'b1: active

SPDIF_RX_INTCLR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10	RW	0x0	UBCICLR Write 1 to clear user bit change interrupt.
9	RW	0x0	ESYNCICLR Write 1 to clear Entering Synchronous State interrupt.
8	RW	0x0	BTEICLR Write 1 to clear Block Transfer/Repetition Period End interrupt.
7	RW	0x0	NSYNCICLR Write 1 to clear CDR Not Synchronization interrupt.
6	RW	0x0	BMDEICLR Write 1 to clear Bi-phase Mark Decoding interrupt.
5	RW	0x0	RXOICLR Write 1 to clear receive overrun interrupt.
4	RW	0x0	RXFICLR Write 1 to clear receive full interrupt.
3	RW	0x0	NPSPICLR Write 1 to clear receive full interrupt.
2	RW	0x0	NVLDICLR Write 1 to clear receive full interrupt.
1	RW	0x0	CSCICLR Write 1 to clear channel status change interrupt.

Bit	Attr	Reset Value	Description
0	RW	0x0	PEICLR Write 1 to clear parity error interrupt.

SPDIF_RX_SMPDR
Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	SMPDR Sample Data Register

SPDIF_RX_USRDRn
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	USR_SUB_1 User Data Bit For Subframe 1
15:0	RO	0x0000	USR_SUB_0 User Data Bit For Subframe 0

SPDIF_RX_CHNSRn
Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CHNSR_SUB_1 Channel Status Bit For Subframe 1
15:0	RO	0x0000	CHNSR_SUB_0 Channel Status Bit For Subframe 0

SPDIF_RX_BURTSINFO
Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PD Preamble Pd for non-linear pcm, indicating the length of burst payload in unit of bytes or bits.
15:13	RW	0x0	BSNUM This field indicates the bitstream number. Usually the bitstream number is 0.
12:8	RO	0x00	DATAINFO This field gives the data-type-dependent info
7	RO	0x0	ERRFLAG 1'b0: indicates a valid burst-payload 1'b1: indicates that the burst-payload may contain errors

Bit	Attr	Reset Value	Description
6:0	RO	0x00	DATATYPE 7'b0000000: null data 7'b0000001: AC-3 data 7'b0000011: Pause data 7'b0000100: MPEG-1 layer 1 data 7'b0000101: MPEG-1 layer 2 or 3 data or MPEG-2 without extension 7'b0000110: MPEG-2 data with extension 7'b0000111: MPEG-2 AAC 7'b0001000: MPEG-2, layer-1 low sampling frequency 7'b0001001: MPEG-2, layer-2 low sampling frequency 7'b0001010: MPEG-2, layer-3 low sampling frequency 7'b0001011: DTS type I 7'b0001100: DTS type II 7'b0001101: DTS type III 7'b0001110: ATRAC 7'b0001111: ATRAC 2/3 7'b0010000: ATRAC-X 7'b0010001: DTS type IV 7'b0010010: WMA professional type I 7'b0110010: WMA professional type II 7'b1010010: WMA professional type III 7'b1110010: WMA professional type IV 7'b0010011: MPEG-2 AAC low sampling frequency 7'b0110011: MPEG-2 AAC low sampling frequency 7'b1010011: MPEG-2 AAC low sampling frequency 7'b1110011: MPEG-2 AAC low sampling frequency 7'b0010100: MPEG-4 AAC 7'b0110100: MPEG-4 AAC 7'b1010100: MPEG-4 AAC 7'b1110100: MPEG-4 AAC 7'b0010101: Enhanced AC-3 7'b0010110: MAT others: reserved

29.5 Interface Description

Table 29-1 SPDIF receiver interface

Module Pin	Dir	PIN Name	IOMUX Setting
spdif_rx_sdi	I	GPIO0_C2/SPDIF_RX	GRF_GPIO0C_IOMUX[5:4]=2’b01

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

The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter by configuring GRF_SOC_CON1[14]. If GRF_SOC_CON1[14] is set to 1, then the SPDIF receiver input signal will be transferred to the SPDIF transmitter.

29.6 Application Notes

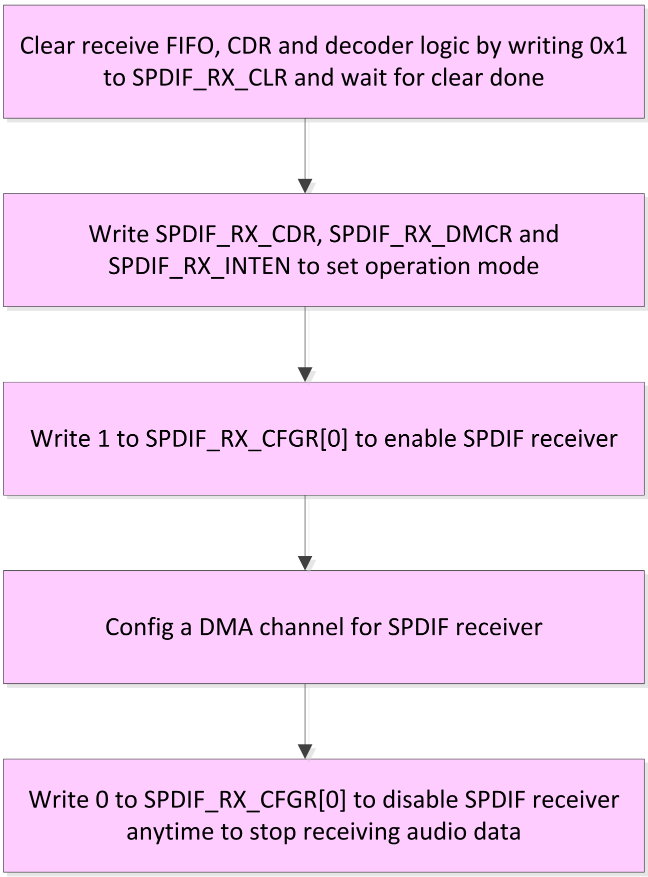

Clear receive FIFO, CDR and decoder logic by writing 0x1 to SPDIF_RX_CLR and wait for clear done
Write SPDIF_RX_CDR, SPDIF_RX_DMCR and SPDIF_RX_INTEN to set operation mode
Write 1 to SPDIF_RX_CFGR[0] to enable SPDIF receiver
Config a DMA channel for SPDIF receiver
Write 0 to SPDIF_RX_CFGR[0] to disable SPDIF receiver anytime to stop receiving audio data

Fig. 29-6 SPDIF receiver operation flow chart

The above figure shows the operation flow of SPDIF receiver operation.

Chapter 30 General Register Files (GRF)

30.1 Overview

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

- GRF, used for general non-secure system
- USBPHY_GRF, used for USB PHY configuration
- DETECT_GRF, used for USB and SDMMC connection and disconnect detect
- CORE_GRF, used for CORE PVTM and Performance Monitor

30.2 Block Diagram

N/A

30.3 Function Description

The function of general register file are：

- IOMUX control
- Control the property of GPIO in power-down mode
- Used for common system control
- Used to record the system state

30.4 Register Description

30.4.1 Internal Address Mapping

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

Table 30-1 GRF Address Mapping Table

Name	Address Base
GRF	0xFF000000
USBPHY_GRF	0xFF008000
DETECT_GRF	0xFF00B000
CORE_GRF	0xFF00C000

30.4.2 GRF Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0A_IOMUX	0x0000	W	0x00000040	GPIO0A iomux function select
GRF_GPIO0B_IOMUX	0x0008	W	0x00000000	GPIO0B iomux function select
GRF_GPIO0C_IOMUX	0x0010	W	0x00000000	GPIO0C iomux function select
GRF_GPIO1A_IOMUX	0x0020	W	0x00000000	GPIO1A iomux function select
GRF_GPIO1B_IOMUX_L	0x0028	W	0x00000000	GPIO1B iomux function select low bits
GRF_GPIO1B_IOMUX_H	0x002c	W	0x00000000	GPIO1B iomux function select high bits
GRF_GPIO1C_IOMUX_L	0x0030	W	0x00000000	GPIO1C iomux function select low bits
GRF_GPIO1C_IOMUX_H	0x0034	W	0x00000440	GPIO1C iomux function select high bits
GRF_GPIO1D_IOMUX	0x0038	W	0x00000000	GPIO1D iomux function select

Name	Offset	Size	Reset Value	Description
GRF_GPIO2A_IOMUX	0x0040	W	0x00000000	GPIO2A iomux function select
GRF_GPIO2B_IOMUX	0x0048	W	0x00000000	GPIO2B iomux function select
GRF_GPIO2C_IOMUX	0x0050	W	0x00000000	GPIO2C iomux function select
GRF_GPIO3A_IOMUX	0x0060	W	0x00000000	GPIO3A iomux function select
GRF_GPIO3B_IOMUX	0x0068	W	0x00000000	GPIO3B iomux function select
GRF_GPIO4A_IOMUX	0x0080	W	0x00000000	GPIO4A iomux function select
GRF_GPIO4B_IOMUX	0x0088	W	0x00000000	GPIO4B iomux function select
GRF_GPIO4C_IOMUX	0x0090	W	0x00000000	GPIO4C iomux function select
GRF_GPIO4D_IOMUX	0x0098	W	0x00000000	GPIO4D iomux function select
GRF_GPIO0A_P	0x00a0	W	0x0000a96a	GPIO0A driver disabled state control
GRF_GPIO0B_P	0x00a4	W	0x0000a96a	GPIO0B driver disabled state control
GRF_GPIO0C_P	0x00a8	W	0x00000a2a	GPIO0C driver disabled state control
GRF_GPIO1A_P	0x00b0	W	0x0000aaaa	GPIO1A driver disabled state control
GRF_GPIO1B_P	0x00b4	W	0x0000aaaa	GPIO1B driver disabled state control
GRF_GPIO1C_P	0x00b8	W	0x00005aaa	GPIO1C driver disabled state control
GRF_GPIO1D_P	0x00bc	W	0x00000005	GPIO1D driver disabled state control
GRF_GPIO2A_P	0x00c0	W	0x0000aa55	GPIO2A driver disabled state control
GRF_GPIO2B_P	0x00c4	W	0x0000aaaa	GPIO2B driver disabled state control
GRF_GPIO2C_P	0x00c8	W	0x00000002	GPIO2C driver disabled state control
GRF_GPIO3A_P	0x00d0	W	0x00005555	GPIO3A driver disabled state control
GRF_GPIO3B_P	0x00d4	W	0x00000599	GPIO3B driver disabled state control
GRF_GPIO4A_P	0x00e0	W	0x00005955	GPIO4A driver disabled state control
GRF_GPIO4B_P	0x00e4	W	0x0000aaa5	GPIO4B driver disabled state control
GRF_GPIO4C_P	0x00e8	W	0x00000002	GPIO4C driver disabled state control
GRF_GPIO4D_P	0x00ec	W	0x00002955	GPIO4D driver disabled state control
GRF_GPIO0A_E	0x0100	W	0x00000000	GPIO0A output drive strength selection

Name	Offset	Size	Reset Value	Description
GRF_GPIO0B_E	0x0104	W	0x00000000	GPIO0B output drive strength selection
GRF_GPIO0C_E	0x0108	W	0x00000000	GPIO0C output drive strength selection
GRF_GPIO1A_E	0x0110	W	0x00000000	GPIO1A output drive strength selection
GRF_GPIO1B_E	0x0114	W	0x00000000	GPIO1B output drive strength selection
GRF_GPIO1C_E	0x0118	W	0x00000000	GPIO1C output drive strength selection
GRF_GPIO1D_E	0x011c	W	0x00000000	GPIO1D output drive strength selection
GRF_GPIO2A_E	0x0120	W	0x00000000	GPIO2A output drive strength selection
GRF_GPIO2B_E	0x0124	W	0x00000000	GPIO2B output drive strength selection
GRF_GPIO2C_E	0x0128	W	0x00000000	GPIO2C output drive strength selection
GRF_GPIO3A_E	0x0130	W	0x0000aaaa	GPIO3A output drive strength selection
GRF_GPIO3B_E	0x0134	W	0x00000aaa	GPIO3B output drive strength selection
GRF_GPIO4A_E	0x0140	W	0x00000000	GPIO4A output drive strength selection
GRF_GPIO4B_E	0x0144	W	0x00000000	GPIO4B output drive strength selection
GRF_GPIO4C_E	0x0148	W	0x00000000	GPIO4C output drive strength selection
GRF_GPIO4D_E	0x014c	W	0x00002aaa	GPIO4D output drive strength selection
GRF_GPIO0A_SR	0x0150	W	0x00000000	GPIO0A slew rate control
GRF_GPIO0B_SR	0x0154	W	0x00000000	GPIO0B slew rate control
GRF_GPIO0C_SR	0x0158	W	0x00000000	GPIO0C slew rate control
GRF_GPIO1A_SR	0x0160	W	0x00000000	GPIO1A slew rate control
GRF_GPIO1B_SR	0x0164	W	0x00000000	GPIO1B slew rate control
GRF_GPIO1C_SR	0x0168	W	0x00000000	GPIO1C slew rate control
GRF_GPIO1D_SR	0x016c	W	0x00000000	GPIO1D slew rate control
GRF_GPIO2A_SR	0x0170	W	0x00000000	GPIO2A slew rate control
GRF_GPIO2B_SR	0x0174	W	0x00000000	GPIO2B slew rate control
GRF_GPIO2C_SR	0x0178	W	0x00000000	GPIO2C slew rate control
GRF_GPIO3A_SR	0x0180	W	0x00000000	GPIO3A slew rate control
GRF_GPIO3B_SR	0x0184	W	0x00000000	GPIO3B slew rate control
GRF_GPIO4A_SR	0x0190	W	0x00000000	GPIO4A slew rate control

Name	Offset	Size	Reset Value	Description
GRF_GPIO4B_SR	0x0194	W	0x00000000	GPIO4B slew rate control
GRF_GPIO4C_SR	0x0198	W	0x00000000	GPIO4C slew rate control
GRF_GPIO4D_SR	0x019c	W	0x00000000	GPIO4D slew rate control
GRF_GPIO0A_SMT	0x01a0	W	0x00000000	GPIO0A schmitt trigger select
GRF_GPIO0B_SMT	0x01a4	W	0x00000000	GPIO0B schmitt trigger select
GRF_GPIO0C_SMT	0x01a8	W	0x00000008	GPIO0C schmitt trigger select
GRF_GPIO1A_SMT	0x01b0	W	0x00000000	GPIO1A schmitt trigger select
GRF_GPIO1B_SMT	0x01b4	W	0x00000000	GPIO1B schmitt trigger select
GRF_GPIO1C_SMT	0x01b8	W	0x00000000	GPIO1C schmitt trigger select
GRF_GPIO1D_SMT	0x01bc	W	0x00000000	GPIO1D schmitt trigger select
GRF_GPIO2A_SMT	0x01c0	W	0x00000000	GPIO2A schmitt trigger select
GRF_GPIO2B_SMT	0x01c4	W	0x00000000	GPIO2B schmitt trigger select
GRF_GPIO2C_SMT	0x01c8	W	0x00000000	GPIO2C schmitt trigger select
GRF_GPIO3A_SMT	0x01d0	W	0x00000000	GPIO3A schmitt trigger select
GRF_GPIO3B_SMT	0x01d4	W	0x00000000	GPIO3B schmitt trigger select
GRF_GPIO4A_SMT	0x01e0	W	0x00000000	GPIO4A schmitt trigger select
GRF_GPIO4B_SMT	0x01e4	W	0x00000000	GPIO4B schmitt trigger select
GRF_GPIO4C_SMT	0x01e8	W	0x00000000	GPIO4C schmitt trigger select
GRF_GPIO4D_SMT	0x01ec	W	0x00000000	GPIO4D schmitt trigger select
GRF_SOC_CON0	0x0300	W	0x00000010	SOC control register 0
GRF_SOC_CON1	0x0304	W	0x000004e4	SOC control register 1
GRF_SOC_CON2	0x0308	W	0x00000000	SOC control register 2
GRF_SOC_CON3	0x030c	W	0x00007c00	SOC control register 3
GRF_SOC_CON4	0x0310	W	0x00001000	SOC control register 4
GRF_SOC_CON5	0x0314	W	0x00000000	SOC control register 5
GRF_SOC_CON10	0x0328	W	0x000001ff	SOC control register 10
GRF_SOC_CON11	0x032c	W	0x00000208	SOC control register 11
GRF_SOC_STATUS0	0x0380	W	0x0000000f	SOC status 0
GRF_CPU_CON0	0x0400	W	0x00000000	CPU control register 0
GRF_CPU_CON1	0x0404	W	0x00000000	CPU control register 1
GRF_CPU_CON2	0x0408	W	0x00000021	CPU control register 2
GRF_CPU_STATUS0	0x0420	W	0x00000000	CPU status 0
GRF_CPU_STATUS1	0x0424	W	0x00000000	CPU status 1
GRF_PVTM_CON0	0x0440	W	0x00000000	PVTM control 0
GRF_PVTM_CON1	0x0444	W	0x00000000	PVTM control 1
GRF_PVTM_STATUS0	0x0448	W	0x00000000	PVTM status 0
GRF_PVTM_STATUS1	0x044c	W	0x00000000	PVTM status 1
GRF_TSADC_TESTBIT_L	0x0460	W	0x00000000	TSADC test low bits register
GRF_TSADC_TESTBIT_H	0x0464	W	0x00000000	TSADC test high bits register
GRF_USB2_HOST0_CON0	0x0480	W	0x00000820	USB 2.0 host control 0
GRF_USB2_HOST0_CON1	0x0484	W	0x000004bc	USB 2.0 host control 1
GRF_USB2_OTG_CON0	0x0488	W	0x00000000	USB 2.0 OTG control 0

Name	Offset	Size	Reset Value	Description
GRF_USB2_HOST0_STATUS0	0x048c	W	0x00000000	USB 2.0 host 0 status 0
GRF_MAC_CON0	0x04a0	W	0x00000000	MAC control 0
GRF_UPCTL_CON0	0x04a4	W	0x00000001	Universal DDR protocol controller control 0
GRF_UPCTL_STATUS0	0x04a8	W	0x00000000	Universal DDR protocol controller status 0
GRF_OS_REG0	0x0500	W	0x00000000	OS register 0
GRF_OS_REG1	0x0504	W	0x00000000	OS register 1
GRF_OS_REG2	0x0508	W	0x00000000	OS register 2
GRF_OS_REG3	0x050c	W	0x00000000	OS register 3
GRF_OS_REG4	0x0510	W	0x00000000	OS register 4
GRF_OS_REG5	0x0514	W	0x00000000	OS register 5
GRF_OS_REG6	0x0518	W	0x00000000	OS register 6
GRF_OS_REG7	0x051c	W	0x00000000	OS register 7
GRF_OS_REG8	0x0520	W	0x00000000	OS register 8
GRF_OS_REG9	0x0524	W	0x00000000	OS register 9
GRF_OS_REG10	0x0528	W	0x00000000	OS register 10
GRF_OS_REG11	0x052c	W	0x00000000	OS register 11
GRF_SOC_CON12	0x0600	W	0x00000000	SOC control register 12
GRF_CHIP_ID	0x0800	W	0x00000cea	Chip ID register

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

30.4.3 GRF Detail Register Description

GRF_GPIO0A_IOMUX

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio0a7_sel 2'b00: GPIO0_A7 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio0a6_sel 2'b00: GPIO0_A6 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved
11:10	RW	0x0	gpio0a5_sel 2'b00: GPIO0_A5 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved
9:8	RW	0x0	gpio0a4_sel 2'b00: GPIO0_A4 2'b01: TEST_CLKOUT 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x1	gpio0a3_sel 2'b00: GPIO0_A3 2'b01: SDMMC_DET 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio0a2_sel 2'b00: GPIO0_A2 2'b01: SDIO_PWREN 2'b10: Reserved 2'b11: Reserved
3:2	RW	0x0	gpio0a1_sel 2'b00: GPIO0_A1 2'b01: SDIO_WRPT 2'b10: Reserved 2'b11: Reserved
1:0	RW	0x0	gpio0a0_sel 2'b00: GPIO0_A0 2'b01: SDIO_INTN 2'b10: Reserved 2'b11: Reserved

GRF_GPIO0B_IOMUX
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio0b7_sel 2'b00: GPIO0_B7 2'b01: PWM2 2'b10: I2C3_SDA_M0 2'b11: Reserved
13:12	RW	0x0	gpio0b6_sel 2'b00: GPIO0_B6 2'b01: PWM1 2'b10: Reserved 2'b11: Reserved
11:10	RW	0x0	gpio0b5_sel 2'b00: GPIO0_B5 2'b01: PWM0 2'b10: Reserved 2'b11: Reserved
9:8	RW	0x0	gpio0b4_sel 2'b00: GPIO0_B4 2'b01: I2C1_SCL 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x0	gpio0b3_sel 2'b00: GPIO0_B3 2'b01: I2C1_SDA 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio0b2_sel 2'b00: GPIO0_B2 2'b01: TSADC_SHUT 2'b10: Reserved 2'b11: Reserved
3:2	RW	0x0	gpio0b1_sel 2'b00: GPIO0_B1 2'b01: PMIC_SLEEP 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio0b0_sel 2'b00: GPIO0_B0 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved

GRF_GPIO0C_IOMUX

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:10	RW	0x0	gpio0c5_sel 2'b00: GPIO0_C5/OTG_DRVBUS 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved When GPIO0_C5 is configured as GPIO function mode, it can be controlled by SOC_CON0[6], when set otg_drvvbus_out_ctrl HIGH, the gpio out data is derived from otg_drvvbus.
9:8	RW	0x0	gpio0c4_sel 2'b00: GPIO0_C4 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x0	gpio0c3_sel 2'b00: GPIO0_C3 2'b01: RTC_CLK 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio0c2_sel 2'b00: GPIO0_C2 2'b01: SPDIF_RX 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio0c1_sel 2'b00: GPIO0_C1 2'b01: SPDIF_TX 2'b10: Reserved 2'b11: Reserved
1:0	RW	0x0	gpio0c0_sel 2'b00: GPIO0_C0 2'b01: PWM3 2'b10: I2C3_SCL_M0 2'b11: Reserved

GRF_GPIO1A_IOMUX

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio1a7_sel 2'b00: GPIO1_A7 2'b01: LCDC_D3 2'b10: I2S1_8CH_SDO0_M0 (8CH) 2'b11: Reserved
13:12	RW	0x0	gpio1a6_sel 2'b00: GPIO1_A6 2'b01: LCDC_D2 2'b10: I2S1_8CH_LRCK_RX_M0 (8CH) 2'b11: Reserved
11:10	RW	0x0	gpio1a5_sel 2'b00: GPIO1_A5 2'b01: LCDC_D1 2'b10: I2S1_8CH_LRCK_TX_M0 (8CH) 2'b11: Reserved
9:8	RW	0x0	gpio1a4_sel 2'b00: GPIO1_A4 2'b01: LCDC_D0 2'b10: I2S1_8CH_SCLK_RX_M0 (8CH) 2'b11: PDM_8CH_CLK_M0

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio1a3_sel 2'b00: GPIO1_A3 2'b01: LCDC_DEN 2'b10: I2S1_8CH_SCLK_TX_M0 (8CH) 2'b11: Reserved
5:4	RW	0x0	gpio1a2_sel 2'b00: GPIO1_A2 2'b01: LCDC_VSYNC 2'b10: I2S1_8CH_MCLK_M0 (8CH) 2'b11: Reserved
3:2	RW	0x0	gpio1a1_sel 2'b00: GPIO1_A1 2'b01: LCDC_HSYNC 2'b10: Reserved 2'b11: Reserved
1:0	RW	0x0	gpio1a0_sel 2'b00: GPIO1_A0 2'b01: LCDC_DCLK 2'b10: Reserved 2'b11: Reserved

GRF_GPIO1B_IOMUX_L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RW	0x0	gpio1b6_sel 4'h0: GPIO1_B6 4'h1: LCDC_D10 4'h2: I2S1_8CH_SCLK_RX_M1 (8CH) 4'h3: MAC_MDIO 4'h4: PDM_8CH_CLK_M1 Others: Reserved
11:10	RW	0x0	gpio1b5_sel 2'b00: GPIO1_B5 2'b01: LCDC_D9 2'b10: I2S1_8CH_SCLK_TX_M1 (8CH) 2'b11: MAC_MDC

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio1b4_sel 2'b00: GPIO1_B4 2'b01: LCDC_D8 2'b10: I2S1_8CH_MCLK_M1 (8CH) 2'b11: MAC_CLK
7:6	RW	0x0	gpio1b3_sel 2'b00: GPIO1_B3 2'b01: LCDC_D7 2'b10: I2S1_8CH_SDI0_M0 (8CH) 2'b11: PDM_8CH_SDI0_M0
5:4	RW	0x0	gpio1b2_sel 2'b00: GPIO1_B2 2'b01: LCDC_D6 2'b10: I2S1_8CH_SDO3_SDI1_M0 (8CH) 2'b11: PDM_8CH_SDI1_M0
3:2	RW	0x0	gpio1b1_sel 2'b00: GPIO1_B1 2'b01: LCDC_D5 2'b10: I2S1_8CH_SDO2_SDI2_M0 (8CH) 2'b11: PDM_8CH_SDI2_M0
1:0	RW	0x0	gpio1b0_sel 2'b00: GPIO1_B0 2'b01: LCDC_D4 2'b10: I2S1_8CH_SDO1_SDI3_M0 (8CH) 2'b11: PDM_8CH_SDI3_M0

GRF_GPIO1B_IOMUX_H

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio1b7_sel 2'b00: GPIO1_B7 2'b01: LCDC_D11 2'b10: I2S1_8CH_LRCK_TX_M1 (8CH) 2'b11: MAC_RXER

GRF_GPIO1C_IOMUX_L
Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RW	0x0	gpio1c4_sel 4'h0: GPIO1_C4 4'h1: LCDC_D16 4'h2: I2S1_8CH_SDO3_SDI1_M1 (8CH) 4'h3: MAC_RXD0 4'h4: PDM_8CH_SDI1_M1 Others: Reserved
11:8	RW	0x0	gpio1c3_sel 4'h0: GPIO1_C3 4'h1: LCDC_DA15 4'h2: I2S1_8CH_SDO2_SDI2_M1 (8CH) 4'h3: MAC_TXD1 4'h4: PDM_8CH_SDI2_M1 Others: Reserved
7:4	RW	0x0	gpio1c2_sel 4'h0: GPIO1_C2 4'h1: LCDC_D14 4'h2: I2S1_8CH_SDO1_SDI3_M1 (8CH) 4'h3: MAC_TXD0 4'h4: PDM_8CH_SDI3_M1 Others: Reserved
3:2	RW	0x0	gpio1c1_sel 2'b00: GPIO1_C1 2'b01: LCDC_D13 2'b10: I2S1_8CH_SDO0_M1 (8CH) 2'b11: MAC_TXEN
1:0	RW	0x0	gpio1c0_sel 2'b00: GPIO1_C0 2'b01: LCDC_D12 2'b10: I2S1_8CH_LRCK_RX_M1 (8CH) 2'b11: MAC_RXDV

GRF_GPIO1C_IOMUX_H
Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:8	RW	0x4	gpio1c7_sel 4'h0: GPIO1_C7 4'h1: UART1_RTSN 4'h2: UART2_TX_M0 4'h3: SPI2_MOSI 4'h4: JTAG_TMS Others: Reserved
7:4	RW	0x4	gpio1c6_sel 4'h0: GPIO1_C6 4'h1: UART1_CTSN 4'h2: UART2_RX_M0 4'h3: SPI2_MISO 4'h4: JTAG_TCK Others: Reserved
3:0	RW	0x0	gpio1c5_sel 4'h0: GPIO1_C5 4'h1: LCDC_D17 4'h2: I2S1_8CH_SDI0_M1 (8CH) 4'h3: MAC_RXD1 4'h4: PDM_8CH_SDI0_M1 Others: Reserved

GRF_GPIO1D_IOMUX

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:2	RW	0x0	gpio1d1_sel 2'b00: GPIO1_D1 2'b01: UART1_TX 2'b10: I2C0_SCL 2'b11: SPI2_CSN0
1:0	RW	0x0	gpio1d0_sel 2'b00: GPIO1_D0 2'b01: UART1_RX 2'b10: I2C0_SDA 2'b11: SPI2_CLK

GRF_GPIO2A_IOMUX

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio2a7_sel 2'b00: GPIO2_A7 2'b01: I2S0_8CH_LRCK_TX (8CH) 2'b10: Reserved 2'b11: Reserved
13:12	RW	0x0	gpio2a6_sel 2'b00: GPIO2_A6 2'b01: I2S0_8CH_SCLK_RX (8CH) 2'b10: PDM_8CH_CLK_S_M2 2'b11: Reserved
11:10	RW	0x0	gpio2a5_sel 2'b00: GPIO2_A5 2'b01: I2S0_8CH_SCLK_TX (8CH) 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio2a4_sel 2'b00: GPIO2_A4 2'b01: I2S0_8CH_MCLK (8CH) 2'b10: Reserved 2'b11: Reserved 2'bxx: PDM_8CH_CLK_M_M2 When SOC_CON12[2] grf_gpio2a4_pad_out_src_sel is set to HIGH, the GPIO2_A4 is drived as PDM_8CH_CLK_M_M2 regardless of IOMUX setting of gpio2a4_sel, and the IO output enable is controlled by GPIO0_B1 IO data out signal. To make sure the functionality works well, the GPIO0_B1 should be configured as PMIC_SLEEP function as application required.
7:6	RW	0x0	gpio2a3_sel 2'b00: GPIO2_A3 2'b01: UART0_RTSN 2'b10: SPI0_CSN0 2'b11: I2C2_SCL
5:4	RW	0x0	gpio2a2_sel 2'b00: GPIO2_A2 2'b01: UART0_CTSN 2'b10: SPI0_CLK 2'b11: I2C2_SDA
3:2	RW	0x0	gpio2a1_sel 2'b00: GPIO2_A1 2'b01: UART0_TX 2'b10: SPI0_MOSI 2'b11: Reserved
1:0	RW	0x0	gpio2a0_sel 2'b00: GPIO2_A0 2'b01: UART0_RX 2'b10: SPI0_MISO 2'b11: Reserved

GRF_GPIO2B_IOMUX

Address: Operational Base + offset (0x0048)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio2b7_sel 2'b00: GPIO2_B7 2'b01: I2S0_8CH_SDI2 (8CH) 2'b10: PDM_8CH_SDI2_M2 2'b11: Reserved
13:12	RW	0x0	gpio2b6_sel 2'b00: GPIO2_B6 2'b01: I2S0_8CH_SDI1 (8CH) 2'b10: PDM_8CH_SDI1_M2 2'b11: Reserved
11:10	RW	0x0	gpio2b5_sel 2'b00: GPIO2_B5 2'b01: I2S0_8CH_SDI0 (8CH) 2'b10: PDM_8CH_SDI0_M2 2'b11: Reserved
9:8	RW	0x0	gpio2b4_sel 2'b00: GPIO2_B4 2'b01: I2S0_8CH_SDO3 (8CH) 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x0	gpio2b3_sel 2'b00: GPIO2_B3 2'b01: I2S0_8CH_SDO2 (8CH) 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio2b2_sel 2'b00: GPIO2_B2 2'b01: I2S0_8CH_SDO1 (8CH) 2'b10: Reserved 2'b11: Reserved
3:2	RW	0x0	gpio2b1_sel 2'b00: GPIO2_B1 2'b01: I2S0_8CH_SDO0 (8CH) 2'b10: Reserved 2'b11: Reserved
1:0	RW	0x0	gpio2b0_sel 2'b00: GPIO2_B0 2'b01: I2S0_8CH_LRCK_RX (8CH) 2'b10: Reserved 2'b11: Reserved

GRF_GPIO2C_IOMUX
Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio2c0_sel 2'b00: GPIO2_C0 2'b01: I2S0_8CH_SDI3 (8CH) 2'b10: PDM_8CH_SDI3_M2 2'b11: Reserved

GRF_GPIO3A_IOMUX

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio3a7_sel 2'b00: GPIO3_A7 2'b01: FLASH_D7 2'b10: EMMC_D7 2'b11: Reserved
13:12	RW	0x0	gpio3a6_sel 2'b00: GPIO3_A6 2'b01: FLASH_D6 2'b10: EMMC_D6 2'b11: Reserved
11:10	RW	0x0	gpio3a5_sel 2'b00: GPIO3_A5 2'b01: FLASH_D5 2'b10: EMMC_D5 2'b11: SFC_CSN0

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio3a4_sel 2'b00: GPIO3_A4 2'b01: FLASH_D4 2'b10: EMMC_D4 2'b11: SFC_CLK
7:6	RW	0x0	gpio3a3_sel 2'b00: GPIO3_A3 2'b01: FLASH_D3 2'b10: EMMC_D3 2'b11: SFC_HOLD_SIO3
5:4	RW	0x0	gpio3a2_sel 2'b00: GPIO3_A2 2'b01: FLASH_D2 2'b10: EMMC_D2 2'b11: SFC_WP_SIO2
3:2	RW	0x0	gpio3a1_sel 2'b00: GPIO3_A1 2'b01: FLASH_D1 2'b10: EMMC_D1 2'b11: SFC_SIO1
1:0	RW	0x0	gpio3a0_sel 2'b00: GPIO3_A0 2'b01: FLASH_D0 2'b10: EMMC_D0 2'b11: SFC_SIO0

GRF_GPIO3B_IOMUX

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RW	0x0	gpio3b5_sel 4'h0: GPIO3_B5 4'h1: FLASH_CSN0 4'h2: I2C3_SCL_M1 4'h3: SPI1_CSN0 4'h4: UART3_TX Others: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x0	gpio3b4_sel 4'h0: GPIO3_B4 4'h1: FLASH_RDY 4'h2: I2C3_SDA_M1 4'h3: SPI1_MOSI 4'h4: UART3_RX Others: Reserved
7:6	RW	0x0	gpio3b3_sel 2'b00: GPIO3_B3 2'b01: FLASH_ALE 2'b10: EMMC_PWREN 2'b11: SPI1_CLK
5:4	RW	0x0	gpio3b2_sel 2'b00: GPIO3_B2 2'b01: FLASH_RDN 2'b10: ~~EMMC_RSTN~~Reserved 2'b11: SPI1_MISO
3:2	RW	0x0	gpio3b1_sel 2'b00: GPIO3_B1 2'b01: FLASH_CLE 2'b10: EMMC_CLK 2'b11: Reserved
1:0	RW	0x0	gpio3b0_sel 2'b00: GPIO3_B0 2'b01: FLASH_WRN 2'b10: EMMC_CMD 2'b11: Reserved

GRF_GPIO4A_IOMUX

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio4a7_sel 2'b00: GPIO4_A7 2'b01: UART4_RTSN 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio4a6_sel 2'b00: GPIO4_A6 2'b01: UART4_CTSN 2'b10: Reserved 2'b11: Reserved
11:10	RW	0x0	gpio4a5_sel 2'b00: GPIO4_A5 2'b01: SDIO_CLK 2'b10: Reserved 2'b11: Reserved
9:8	RW	0x0	gpio4a4_sel 2'b00: GPIO4_A4 2'b01: SDIO_CMD 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x0	gpio4a3_sel 2'b00: GPIO4_A3 2'b01: SDIO_D3 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio4a2_sel 2'b00: GPIO4_A2 2'b01: SDIO_D2 2'b10: Reserved 2'b11: Reserved
3:2	RW	0x0	gpio4a1_sel 2'b00: GPIO4_A1 2'b01: SDIO_D1 2'b10: Reserved 2'b11: Reserved
1:0	RW	0x0	gpio4a0_sel 2'b00: GPIO4_A0 2'b01: SDIO_D0 2'b10: Reserved 2'b11: Reserved

GRF_GPIO4B_IOMUX
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio4b7_sel 2'b00: GPIO4_B7 2'b01: I2S0_2CH_SDO 2'b10: Reserved 2'b11: Reserved
13:12	RW	0x0	gpio4b6_sel 2'b00: GPIO4_B6 2'b01: I2S0_2CH_LRCK_TX 2'b10: Reserved 2'b11: Reserved
11:10	RW	0x0	gpio4b5_sel 2'b00: GPIO4_B5 2'b01: I2S0_2CH_SCLK 2'b10: Reserved 2'b11: Reserved
9:8	RW	0x0	gpio4b4_sel 2'b00: GPIO4_B4 2'b01: I2S0_2CH_MCLK 2'b10: Reserved 2'b11: Reserved
7:6	RW	0x0	gpio4b3_sel 2'b00: GPIO4_B3 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved
5:4	RW	0x0	gpio4b2_sel 2'b00: GPIO4_B2 2'b01: Reserved 2'b10: Reserved 2'b11: Reserved
3:2	RW	0x0	gpio4b1_sel 2'b00: GPIO4_B1 2'b01: UART4_TX 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio4b0_sel 2'b00: GPIO4_B0 2'b01: UART4_RX 2'b10: Reserved 2'b11: Reserved

GRF_GPIO4C_IOMUX

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio4c0_sel 2'b00: GPIO4_C0 2'b01: I2S0_2CH_SDI 2'b10: Reserved 2'b11: Reserved

GRF_GPIO4D_IOMUX

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RO	0x0	reserved
13:12	RW	0x0	gpio4d6_sel 2'b00: GPIO4_D6 2'b01: SDMMC_PWREN 2'b10: Reserved 2'b11: Reserved

Bit	Attr	Reset Value	Description
11:10	RW	0x0	gpio4d5_sel 2'b00: GPIO4_D5/PMU_DEBUG_TX 2'b01: SDMMC_CLK 2'b10: Reserved 2'b11: Reserved When GPIO4_D5 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_debug_tx
9:8	RW	0x0	gpio4d4_sel 2'b00: GPIO4_D4/PMU_POWER_STATE_4 2'b01: SDMMC_CMD 2'b10: Reserved 2'b11: Reserved When GPIO4_D4 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_power_state[4].
7:6	RW	0x0	gpio4d3_sel 2'b00: GPIO4_D3/PMU_POWER_STATE_3 2'b01: SDMMC_D3 2'b10: UART2_TX_M1 2'b11: Reserved When GPIO4_D3 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_power_state[3].
5:4	RW	0x0	gpio4d2_sel 2'b00: GPIO4_D2/PMU_POWER_STATE_2 2'b01: SDMMC_D2 2'b10: UART2_RX_M1 2'b11: Reserved When GPIO4_D2 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_power_state[2].
3:2	RW	0x0	gpio4d1_sel 2'b00: GPIO4_D1/PMU_POWER_STATE_1 2'b01: SDMMC_D1 2'b10: Reserved 2'b11: Reserved When GPIO4_D1 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_power_state[1].

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio4d0_sel 2'b00: GPIO4_D0/PMU_POWER_STATE_0 2'b01: SDMMC_D0 2'b10: Reserved 2'b11: Reserved When GPIO4_D0 is configured as GPIO function mode, it can be controlled by SOC_CON1[15], When set pmu_power_state_out_ctrl HIGH, the gpio out data is derived from pmu_power_state[0].

GRF_GPIO0A_P

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio0a7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio0a6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio0a5_p Driver disabled state control. 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	gpio0a4_p Driver disabled state control. 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
7:6	RW	0x1	gpio0a3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio0a2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio0a1_p Driver disabled state control. 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	gpio0a0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO0B_P

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio0b7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
13:12	RW	0x2	gpio0b6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio0b5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x1	gpio0b4_p Driver disabled state control. 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	gpio0b3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio0b2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio0b1_p Driver disabled state control. 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	gpio0b0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO0C_P

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:10	RW	0x2	gpio0c5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio0c4_p Driver disabled state control. 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	0x0	gpio0c3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio0c2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio0c1_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
1:0	RW	0x2	gpio0c0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO1A_P

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio1a7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio1a6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio1a5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio1a4_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7:6	RW	0x2	gpio1a3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
5:4	RW	0x2	gpio1a2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio1a1_p Driver disabled state control. 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	gpio1a0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO1B_P

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio1b7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio1b6_p Driver disabled state control. 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
11:10	RW	0x2	gpio1b5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio1b4_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7:6	RW	0x2	gpio1b3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio1b2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio1b1_p Driver disabled state control. 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	gpio1b0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO1C_P

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x1	gpio1c7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x1	gpio1c6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio1c5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio1c4_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7:6	RW	0x2	gpio1c3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio1c2_p Driver disabled state control. 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	gpio1c1_p Driver disabled state control. 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	gpio1c0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO1D_P

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:4	RO	0x0	reserved
3:2	RW	0x1	gpio1d1_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
1:0	RW	0x1	gpio1d0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO2A_P

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio2a7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio2a6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio2a5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio2a4_p Driver disabled state control. 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	gpio2a3_p Driver disabled state control. 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	gpio2a2_p Driver disabled state control. 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	0x1	gpio2a1_p Driver disabled state control. 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	gpio2a0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO2B_P

Address: Operational Base + offset (0x00c4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio2b7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio2b6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio2b5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
9:8	RW	0x2	gpio2b4_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7:6	RW	0x2	gpio2b3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio2b2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio2b1_p Driver disabled state control. 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	gpio2b0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO2C_P

Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio2c0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO3A_P

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x1	gpio3a7_p Driver disabled state control. 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	gpio3a6_p Driver disabled state control. 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	gpio3a5_p Driver disabled state control. 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	gpio3a4_p Driver disabled state control. 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
7:6	RW	0x1	gpio3a3_p Driver disabled state control. 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	gpio3a2_p Driver disabled state control. 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	gpio3a1_p Driver disabled state control. 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	gpio3a0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO3B_P

Address: Operational Base + offset (0x00d4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:10	RW	0x1	gpio3b5_p Driver disabled state control. 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	gpio3b4_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7:6	RW	0x2	gpio3b3_p Driver disabled state control. 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	gpio3b2_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
3:2	RW	0x2	gpio3b1_p Driver disabled state control. 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	gpio3b0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO4A_P

Address: Operational Base + offset (0x00e0)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x1	gpio4a7_p Driver disabled state control. 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	gpio4a6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio4a5_p Driver disabled state control. 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	gpio4a4_p Driver disabled state control. 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	gpio4a3_p Driver disabled state control. 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	gpio4a2_p Driver disabled state control. 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	gpio4a1_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio4a0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO4B_P

Address: Operational Base + offset (0x00e4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio4b7_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
13:12	RW	0x2	gpio4b6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio4b5_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
9:8	RW	0x2	gpio4b4_p Driver disabled state control. 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
7:6	RW	0x2	gpio4b3_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
5:4	RW	0x2	gpio4b2_p Driver disabled state control. 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	gpio4b1_p Driver disabled state control. 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	gpio4b0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO4C_P
Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x2	gpio4c0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO4D_P

Address: Operational Base + offset (0x00ec)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RO	0x0	reserved
13:12	RW	0x2	gpio4d6_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
11:10	RW	0x2	gpio4d5_p Driver disabled state control. 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	gpio4d4_p Driver disabled state control. 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	gpio4d3_p Driver disabled state control. 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	gpio4d2_p Driver disabled state control. 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	0x1	gpio4d1_p Driver disabled state control. 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	gpio4d0_p Driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)

GRF_GPIO0A_E

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio0a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio0a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio0a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio0a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio0a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio0a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO0B_E

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio0b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio0b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
11:10	RW	0x0	gpio0b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio0b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio0b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio0b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO0C_E

Address: Operational Base + offset (0x0108)

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

Bit	Attr	Reset Value	Description
11:10	RW	0x0	gpio0c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio0c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio0c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio0c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio0c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio0c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1A_E

Address: Operational Base + offset (0x0110)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x0	gpio1a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio1a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio1a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio1a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio1a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio1a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio1a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio1a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1B_E
Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio1b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio1b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio1b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio1b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio1b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio1b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio1b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio1b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1C_E

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio1c7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio1c6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio1c5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio1c4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio1c3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio1c2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio1c1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio1c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO1D_E

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:4	RO	0x0	reserved
3:2	RW	0x0	gpio1d1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio1d0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2A_E

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio2a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio2a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio2a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio2a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio2a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio2a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio2a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x0	gpio2a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO2B_E

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio2b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio2b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio2b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio2b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio2b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

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

GRF_GPIO2C_E

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio2c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3A_E

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x2	gpio3a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x2	gpio3a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x2	gpio3a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio3a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x2	gpio3a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio3a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio3a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio3a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO3B_E

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:10	RW	0x2	gpio3b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio3b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x2	gpio3b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio3b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio3b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio3b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4A_E

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio4a7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio4a6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio4a5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x0	gpio4a4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio4a3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio4a2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio4a1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio4a0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4B_E

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RW	0x0	gpio4b7_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
13:12	RW	0x0	gpio4b6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x0	gpio4b5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
9:8	RW	0x0	gpio4b4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x0	gpio4b3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x0	gpio4b2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x0	gpio4b1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
1:0	RW	0x0	gpio4b0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4C_E

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio4c0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO4D_E

Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:14	RO	0x0	reserved
13:12	RW	0x2	gpio4d6_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
11:10	RW	0x2	gpio4d5_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	gpio4d4_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
7:6	RW	0x2	gpio4d3_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
5:4	RW	0x2	gpio4d2_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	gpio4d1_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

Bit	Attr	Reset Value	Description
1:0	RW	0x2	gpio4d0_e 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA

GRF_GPIO0A_SR

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio0a7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio0a6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio0a5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio0a4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio0a3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio0a2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio0a1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio0a0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO0B_SR

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio0b7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio0b6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio0b5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio0b4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio0b3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio0b2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio0b1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio0b0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO0C_SR

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:6	RO	0x0	reserved
5	RW	0x0	gpio0c5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio0c4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio0c3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio0c2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio0c1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio0c0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO1A_SR

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1a7_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
6	RW	0x0	gpio1a6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio1a5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio1a4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio1a3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio1a2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio1a1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio1a0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO1B_SR

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1b7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio1b6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio1b5_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio1b4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio1b3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio1b2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio1b1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio1b0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO1C_SR

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1c7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio1c6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio1c5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio1c4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio1c3_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio1c2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio1c1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio1c0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO1D_SR

Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1	RW	0x0	gpio1d1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio1d0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO2A_SR

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio2a7_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
6	RW	0x0	gpio2a6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio2a5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio2a4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio2a3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio2a2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio2a1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio2a0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO2B_SR

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio2b7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio2b6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio2b5_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio2b4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio2b3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio2b2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio2b1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio2b0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO2C_SR

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:1	RO	0x0	reserved
0	RW	0x0	gpio2c0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO3A_SR

Address: Operational Base + offset (0x0180)

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

Bit	Attr	Reset Value	Description
7	RW	0x0	gpio3a7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio3a6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio3a5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio3a4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio3a3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio3a2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio3a1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio3a0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO3B_SR

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:6	RO	0x0	reserved
5	RW	0x0	gpio3b5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio3b4_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3b3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio3b2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio3b1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio3b0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO4A_SR

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio4a7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio4a6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio4a5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio4a4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio4a3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio4a2_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio4a1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio4a0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO4B_SR

Address: Operational Base + offset (0x0194)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio4b7_sr 1'b0: Slow(half frequency) 1'b1: Fast
6	RW	0x0	gpio4b6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio4b5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio4b4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio4b3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio4b2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio4b1_sr 1'b0: Slow(half frequency) 1'b1: Fast
0	RW	0x0	gpio4b0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO4C_SR

Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:1	RO	0x0	reserved
0	RW	0x0	gpio4c0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO4D_SR

Address: Operational Base + offset (0x019c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:7	RO	0x0	reserved
6	RW	0x0	gpio4d6_sr 1'b0: Slow(half frequency) 1'b1: Fast
5	RW	0x0	gpio4d5_sr 1'b0: Slow(half frequency) 1'b1: Fast
4	RW	0x0	gpio4d4_sr 1'b0: Slow(half frequency) 1'b1: Fast
3	RW	0x0	gpio4d3_sr 1'b0: Slow(half frequency) 1'b1: Fast
2	RW	0x0	gpio4d2_sr 1'b0: Slow(half frequency) 1'b1: Fast
1	RW	0x0	gpio4d1_sr 1'b0: Slow(half frequency) 1'b1: Fast

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio4d0_sr 1'b0: Slow(half frequency) 1'b1: Fast

GRF_GPIO0A_SMT
Address: Operational Base + offset (0x01a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio0a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio0a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio0a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO0B_SMT
Address: Operational Base + offset (0x01a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio0b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio0b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio0b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio0b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO0C_SMT

Address: Operational Base + offset (0x01a8)

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

Bit	Attr	Reset Value	Description
15:6	RO	0x0	reserved
5	RW	0x0	gpio0c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio0c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x1	gpio0c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio0c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio0c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio0c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1A_SMT

Address: Operational Base + offset (0x01b0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio1a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1B_SMT

Address: Operational Base + offset (0x01b4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	gpio1b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1C_SMT

Address: Operational Base + offset (0x01b8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio1c7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio1c6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio1c5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio1c4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio1c3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio1c2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio1c1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO1D_SMT

Address: Operational Base + offset (0x01bc)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	reserved
1	RW	0x0	gpio1d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio1d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2A_SMT

Address: Operational Base + offset (0x01c0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio2a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio2a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
2	RW	0x0	gpio2a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio2a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2B_SMT

Address: Operational Base + offset (0x01c4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio2b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio2b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio2b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio2b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio2b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio2b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio2b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	gpio2b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO2C_SMT

Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:1	RO	0x0	reserved
0	RW	0x0	gpio2c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3A_SMT

Address: Operational Base + offset (0x01d0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio3a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio3a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio3a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

Bit	Attr	Reset Value	Description
3	RW	0x0	gpio3a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO3B_SMT

Address: Operational Base + offset (0x01d4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:6	RO	0x0	reserved
5	RW	0x0	gpio3b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio3b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio3b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio3b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio3b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio3b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO4A_SMT

Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio4a7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio4a6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio4a5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio4a4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio4a3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio4a2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio4a1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio4a0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO4B_SMT

Address: Operational Base + offset (0x01e4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:8	RO	0x0	reserved
7	RW	0x0	gpio4b7_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
6	RW	0x0	gpio4b6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio4b5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio4b4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio4b3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio4b2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio4b1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio4b0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO4C_SMT

Address: Operational Base + offset (0x01e8)

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

Bit	Attr	Reset Value	Description
15:1	RO	0x0	reserved
0	RW	0x0	gpio4c0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_GPIO4D_SMT

Address: Operational Base + offset (0x01ec)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:7	RO	0x0	reserved
6	RW	0x0	gpio4d6_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5	RW	0x0	gpio4d5_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
4	RW	0x0	gpio4d4_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
3	RW	0x0	gpio4d3_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
2	RW	0x0	gpio4d2_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
1	RW	0x0	gpio4d1_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
0	RW	0x0	gpio4d0_smt 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_SOC_CON0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:9	RO	0x0	reserved
8	RW	0x0	io_vsel3_ctrl After power up, the software can use io_vsel3 to take over the control. The software should configure the io_vsel3 same value with GPIO0_A4 power-up input before setting this bit. 1'b0: VCC IO 3 voltage is controlled by GPIO0_A4 function input 1'b1: VCC IO 3 voltage is controlled by io_vsel3
7	RO	0x0	reserved
6	RW	0x0	otg_drvvbus_out_ctrl In GPIO function mode. 1'b0: GPIO0_C5 funtion output data from GPIO function 1'b1: GPIO0_C5 funtion output data from USB 2.0 OTG PHY UTMI drvvbus
5	RW	0x0	io_vsel5 VCC IO 5 voltage select. 1'b0: 3.3V 1'b1: 1.8V
4	RW	0x1	io_vsel4 VCC IO 4 voltage select. 1'b0: 3.3V 1'b1: 1.8V
3	RW	0x0	io_vsel3 VCC IO 3 voltage select. 1'b0: 3.3V 1'b1: 1.8V
2	RW	0x0	io_vsel2 VCC IO 2 voltage select. 1'b0: 3.3V 1'b1: 1.8V
1	RW	0x0	io_vsel1 VCC IO 1 voltage select. 1'b0: 3.3V 1'b1: 1.8V
0	RW	0x0	io_vsel0 VCC IO 0 voltage select. 1'b0: 3.3V 1'b1: 1.8V

GRF_SOC_CON1

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	pmu_debug_tx_out_ctrl In GPIO function mode. 1'b0: GPIO4_D5 function output from gpio 1'b1: GPIO4_D5 function output from pmu_debug_tx
14	RW	0x0	spdif_txsdo_src_sel SPDIF transmit serial data out selection. 1'b0: tx sdo select from spdif tx 1'b1: tx sdo select from spdif rx IO input
13:12	RW	0x0	i2s1_2ch_sdi0_sel 2 channel I2S1 sdi 0 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3
11:10	RW	0x1	i2s3_8ch_sdi1_sel 8 channel I2S3 sdi 1 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3
9:8	RW	0x0	i2s3_8ch_sdi0_sel 8 channel I2S3 sdi 0 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3
7:6	RW	0x3	i2s2_8ch_sdi3_sel 8 channel I2S2 sdi 3 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3

Bit	Attr	Reset Value	Description
5:4	RW	0x2	i2s2_8ch_sdi2_sel 8 channel I2S2 sdi 2 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3
3:2	RW	0x1	i2s2_8ch_sdi1_sel 8 channel I2S2 sdi 1 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3
1:0	RW	0x0	i2s2_8ch_sdi0_sel 8 channel I2S2 sdi 0 source selection. 2'b00: select from acodec sdo 0 2'b01: select from acodec sdo 1 2'b10: select from acodec sdo 2 2'b11: select from acodec sdo 3

GRF_SOC_CON2

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	pmu_power_state_out_ctrl In GPIO function mode. 1'b0: GPIO4_D0~GPIO4_D4 function output from gpio 1'b1: GPIO4_D0~GPIO4_D4 function output from pmu_power_state
14	RW	0x0	i2s_16ch_ctrl_en 8 channel I2S0 and 8 channel I2S1 are combined into 16 channels. 1'b0: disable. I2S0 8ch and I2S1 8ch is individual 1'b1: enable. The tx clock (including sclk and lrck) is connected to the other side of I2S controller inside the chip

Bit	Attr	Reset Value	Description
13:12	RW	0x0	pdm_multi_iofunc_src_sel PDM multiplex IO function selection. 2'b00: PDM select multiplex IO function from pad group 0(m0) 2'b01: PDM select multiplex IO function from pad group 1(m1) 2'b10: PDM select multiplex IO function from pad group 2(m2) 2'b11: reserved
11	RO	0x0	reserved
10	RW	0x0	i2s0_8ch_mclk_out_src_sel 8 channel I2S0 mclk out selection. 1'b0: select 8 channel I2S0 mclk IO out from tx mclk 1'b1: select 8 channel I2S0 mclk IO out from rx mclk
9	RW	0x0	i2s0_8ch_sclk_in_rx_src_sel 8 channel I2S0 rx sclk in selection. 1'b0: select 8 channel I2S0 rx sclk in from sclk relative rx IO input 1'b1: select 8 channel I2S0 rx sclk in from sclk relative tx IO input
8	RW	0x0	i2s0_8ch_sclk_in_tx_src_sel 8 channel I2S0 tx sclk in selection. 1'b0: select 8 channel I2S0 tx sclk in from sclk relative tx IO input 1'b1: select 8 channel I2S0 tx sclk in from sclk relative rx IO input
7	RW	0x0	i2s1_8ch_sd_ioe3 8 channel I2S1 sd io3 (GPIO1_B0 and GPIO1_C4) pin output enable. 1'b0: drive disable 1'b1: normal operation
6	RW	0x0	i2s1_8ch_sd_ioe2 8 channel I2S1 sd io2(GPIO1_B1 and GPIO1_C3) pin output enable. 1'b0: drive disable 1'b1: normal operation
5	RW	0x0	i2s1_8ch_sd_ioe1 8 channel I2S1 sd io1(GPIO1_B0 and GPIO1_C2) pin output enable. 1'b0: drive disable 1'b1: normal operation
4	RO	0x0	reserved
3	RW	0x0	i2s1_8ch_multi_iofunc_src_sel 8 channel I2S1 multiplex IO function selection. 1'b0: 8 channel I2S1 select multiplex IO function from pad group 0(m0) 1'b1: 8 channel I2S1 select multiplex IO function from pad group 1(m1)

Bit	Attr	Reset Value	Description
2	RW	0x0	i2s1_8ch_mclk_out_src_sel 8 channel I2S1 mclk out selection. 1'b0: select 8 channel I2S1 mclk IO out from tx mclk 1'b1: select 8 channel I2S1 mclk IO out from rx mclk
1	RW	0x0	i2s1_8ch_sclk_in_rx_src_sel 8 channel I2S1 rx sclk in selection. 1'b0: select 8 channel I2S1 rx sclk in from sclk relative rx IO input 1'b1: select 8 channel I2S1 rx sclk in from sclk relative tx IO input
0	RW	0x0	i2s1_8ch_sclk_in_tx_src_sel 8 channel I2S1 tx sclk in selection. 1'b0: select 8 channel I2S1 tx sclk in from sclk relative tx IO input 1'b1: select 8 channel I2S1 tx sclk in from sclk relative rx IO input

GRF_SOC_CON3

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RO	0x0	reserved
14	RW	0x1	uart4_dma_req_ctrl 1'b0: normal dma request 1'b1: mask dma request
13	RW	0x1	uart3_dma_req_ctrl 1'b0: normal dma request 1'b1: mask dma request
12	RW	0x1	uart2_dma_req_ctrl 1'b0: normal dma request 1'b1: mask dma request
11	RW	0x1	uart1_dma_req_ctrl 1'b0: normal dma request 1'b1: mask dma request
10	RW	0x1	uart0_dma_req_ctrl 1'b0: normal dma request 1'b1: mask dma request

Bit	Attr	Reset Value	Description
9	RW	0x0	uart4_rts_inv_sel 1'b0: normal rts connection 1'b1: invert rts connection
8	RW	0x0	uart3_rts_inv_sel 1'b0: normal rts connection 1'b1: invert rts connection
7	RW	0x0	uart2_rts_inv_sel 1'b0: normal rts connection 1'b1: invert rts connection
6	RW	0x0	uart1_rts_inv_sel 1'b0: normal rts connection 1'b1: invert rts connection
5	RW	0x0	uart0_rts_inv_sel 1'b0: normal rts connection 1'b1: invert rts connection
4	RW	0x0	uart4_cts_inv_sel 1'b0: normal cts connection 1'b1: invert cts connection
3	RW	0x0	uart3_cts_inv_sel 1'b0: normal cts connection 1'b1: invert cts connection
2	RW	0x0	uart2_cts_inv_sel 1'b0: normal cts connection 1'b1: invert cts connection
1	RW	0x0	uart1_cts_inv_sel 1'b0: normal cts connection 1'b1: invert cts connection
0	RW	0x0	uart0_cts_inv_sel 1'b0: normal cts connection 1'b1: invert cts connection

GRF_SOC_CON4

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x1	grf_vop_dcf_idle VOP dcf idle setting. 1'b0: set vop dcf not idle 1'b1: set vop dcf idle(default)
11	RW	0x0	grf_con_id_dmac1_aw Reserved
10	RW	0x0	grf_con_id_dmac1_ar Reserved
9	RW	0x0	grf_con_id_dmac0_aw Reserved
8	RW	0x0	grf_con_id_dmac0_ar Reserved
7	RW	0x0	grf_con_id_vop_aw Reserved
6	RW	0x0	grf_con_id_vop_ar Reserved
5	RW	0x0	grf_con_id_mac_aw Reserved
4	RW	0x0	grf_con_id_mac_ar Reserved
3	RW	0x0	grf_con_id_crypto_aw Reserved
2	RW	0x0	grf_con_id_crypto_ar Reserved
1	RW	0x0	grf_con_id_cpu_aw Reserved
0	RW	0x0	grf_con_id_cpu_ar Reserved

GRF_SOC_CON5

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	grf_acodec_ad2da_loop Acodec AD to DA loop enable control. 1'b0: Acodec AD to DA loop disable 1'b1: Acodec AD to DA loop enable
7	RW	0x0	grf_vnpor_delay_ctrl The delay time of asserting npor high after VDD33 > VDET+ 1'b0: 50ms 1'b1: 100ms
6	RW	0x0	grf_vnpor_delay_ctrl_src_sel The software can set this bit high to enable the grf_vnpor_delay_ctrl takes over the delay control. 1'b0: delay control setting from GPIO0_C5 1'b1: delay control setting from grf_vnpor_delay_ctrl
5	RW	0x0	grf_vnpor_latch VNPOR high level latch enable. When this bit is set to high, the NPOR output from power detection module will keep latched even though the 3.3V supply is power off. 1'b0: latch disable 1'b1: latch enable
4	RW	0x0	grf_i2c3_multi_iofunc_src_sel I2C3 multiplex IO function selection. 1'b0: I2C3 select multiplex IO function from pad group 0(m0) 1'b1: I2C3 select multiplex IO function from pad group 1(m1)
3:2	RW	0x0	grf_uart2_multi_iofunc_src_sel UART2 multiplex IO function selection. 2'b00: UART2 select multiplex IO function from pad group 0(m0) 2'b01: UART2 select multiplex IO function from pad group 1(m1) 2'b10: UART2 select multiplex IO function from usb phy uart debug port 2'b11: reserved
1	RW	0x0	grf_wifi_ioe WIFI clock (REF_CLKOUT) output enable. 1'b0: drive disable 1'b1: output enable
0	RW	0x0	grf_rtc_32k_ioe 32K RTC clock output enable. 1'b0: drive disable 1'b1: output enable

GRF_SOC_CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:9	RO	0x0	reserved
8	RW	0x1	grf_con_fw_req_link_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
7	RW	0x1	grf_con_cpu2servicevoice_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
6	RW	0x1	grf_con_bus2voiceslv_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
5	RW	0x1	grf_con_bus2peri_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
4	RW	0x1	grf_con_cpu2bus_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
3	RW	0x1	grf_con_cpu2srvmsch_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
2	RW	0x1	grf_con_bus2msch_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
1	RW	0x1	grf_con_peri2msch_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
0	RW	0x1	grf_con_cpu2msch_req_pwrDiscTarg_pwrStall 1'b0: bus return error response 1'b1: bus return ok response

GRF_SOC_CON11

Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:10	RW	0x0	grf_wificlk_e REF_CLKOUT output drive strength selection. 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
9:8	RW	0x2	grf_wificlk_p REF_CLKOUT driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
7	RW	0x0	grf_wificlk_sr REF_CLKOUT slew rate control. 1'b0: slow (half frequency) 1'b1: fast
6	RW	0x0	grf_wificlk_smt REF_CLKOUT Schmitt trigger control. 1'b0: No hysteresis 1'b1: Schmitt trigger enabled
5:4	RW	0x0	grf_nporbypass_e NPOR_BYPASS output drive strength selection. 2'b00: 2mA 2'b01: 4mA 2'b10: 8mA 2'b11: 12mA
3:2	RW	0x2	grf_nporbypass_p NPOR_BYPASS driver disabled state control. 2'b00: Z (normal operation) 2'b01: weak 1 (pull-up) 2'b10: weak 0 (pull-down) 2'b11: repeater (bus keeper)
1	RW	0x0	grf_nporbypass_sr NPOR_BYPASS slew rate control. 1'b0: slow (half frequency) 1'b1: fast

Bit	Attr	Reset Value	Description
0	RW	0x0	grf_nporbypass_smt NPOR_BYPASS Schmitt trigger control. 1'b0: No hysteresis 1'b1: Schmitt trigger enabled

GRF_SOC_STATUS0

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RO	0x0	acodec_dac_master_en_st Acodec ADC master enable status. 1'b0: Audio codec adc is in disable state 1'b1: Audio codec dac is in enable state
26	RO	0x0	acodec_adc_master_en_st Acodec ADC master enable status. 1'b0: Audio codec adc is in disable state 1'b1: Audio codec dac is in enable state
25	RO	0x0	vop_dsp_hold VOP display hold status. 1'b0: dispaly is not in hold state 1'b1: dispaly is in hold state
24	RO	0x0	vop_dma_finish VOP dma finish status. 1'b0: dma is not finish 1'b1: dma finish
23:18	RO	0x00	rkstimer_en_st Secure timer0~5 enable status. 1'b0: timer disable 1'b1: timer enable
17:12	RO	0x00	rktimer_en_st Non-secure timer0~5 enable status. 1'b0: timer disable 1'b1: timer enable
11:10	RO	0x0	reserved
9	RO	0x0	otpcns_sbpi_busy_st Non-secure OTP controller sbpi busy status. 1'b0: idle state 1'b1: busy state
8	RO	0x0	otpcns_user_busy_st Non-secure OTP controller user busy status. 1'b0: idle state 1'b1: busy state

Bit	Attr	Reset Value	Description
7	RO	0x0	otpcs_sbpi_busy_st Secure OTP controller sbpi busy status. 1'b0: idle state 1'b1: busy state
6	RO	0x0	otpcs_user_busy_st Secure OTP controller user busy status. 1'b0: idle state 1'b1: busy state
5	RO	0x0	nandc_master_idle_st NANDC master idle status. 1'b0: busy state 1'b1: idle state
4	RO	0x0	reserved
3	RO	0x1	vpll1_lock_st VPLL1 lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state
2	RO	0x1	vpll0_lock_st VPLL0 lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state
1	RO	0x1	dpll_lock_st DPLL lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state
0	RO	0x1	apll_lock_st APLL lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state

GRF_CPU_CON0

Address: Operational Base + offset (0x0400)

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

Bit	Attr	Reset Value	Description
15:12	RW	0x0	grf_con_cfgte Enable T32 exception. It sets the initial value of the TE bit in the SCTLR and HSCTLR register. Each bit is defined below. 1'b0: TE bit is low 1'b1: TE bit is high These bits are sampled only during reset of the core. Tie low for the ARM instruction set for exception handling. Tie high for the Thumb instruction set for exception handling. Only change it when the core is in the reset state
11:8	RW	0x0	grf_con_cfgend Endianness configuration at reset. It sets the initial value of the EE bits in the SCTLR, HSCTLR, SCTLR_EL1, SCTLR_EL2, and SCTLR_EL3 registers. Each bit is defined below. 1'b0: EE bit is low 1'b1: EE bit is high These bits are sampled only during reset of the core. Tie high for big-endian data during exception handling. Tie it low for little-endian data during exception handling. Only change it when the cores are in the state
7:5	RO	0x0	reserved
4	RW	0x0	grf_con_l2rstdisable Disable automatic L2 cache invalidate at reset. 1'b0: enable 1'b1: disable Assert high to disable L2 cache invalidate at reset. Assert low to enable L2 cache invalidate at reset. Only change it when the processor is in the reset state
3:1	RO	0x0	reserved
0	RW	0x0	grf_con_dbgl1rstdisable Disable L1 data cache automatic invalidate on reset functionality. 1'b0: enable 1'b1: disable This value is sampled only during reset of the processor. Assert low for normal L1 data cache behavior on rest. Assert it high to disable automatic invalidation of L1 data cache on reset for debugging purpose only. This bit must be driven Low during normal processor powerup sequences

GRF_CPU_CON1

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:7	RO	0x0	reserved
6	RW	0x0	grf_con_cpu_ema_detect_en CPU memory EMA change detection enable. 1'b0: disable 1'b1: enable When grf_ema_l2d/grf_emaw_l2d/grf_ema_ra/grf_emaw_ra/grf_emas_ra changed, hardware automatically stops CPU and make it valid to CPU
5	RW	0x0	grf_con_evento_clear Set this bit to clear the evento rising edge state
4	RW	0x0	grf_con_eventi Event input for processor wake-up from WFE state. This pin must be asserted for at least one CLKIN clock cycles. When this signal is asserted, it acts as WFE wake-up event to all the cores in the cluster
3:2	RO	0x0	reserved
1	RW	0x0	grf_con_cfgsdisable Prevents modification of certain Secure registers, including bits that correspond to the Lockable SPIs. CFGSDISABLE is typically de-asserted from reset until Secure software has configured the GIC-400 and then subsequently asserted permanently to provide extra security
0	RW	0x0	grf_con_clrexmonreq Request to clear the external global exclusive monitor. This sends a WFE wake-up event to all cores in the cluster. When set high the global exclusive monitor in the system is requesting the processor EVENT registers to be set high

GRF_CPU_CON2

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:11	RO	0x0	reserved
10	WO	0x0	grf_emas_ra Extra margin adjustment sense amplifier pulse, affects cycle time in the read cycle
9:8	RW	0x0	grf_emaw_ra Extra margin adjustment for write access
7:5	RW	0x1	grf_ema_ra Extra margin adjustment
4:3	RW	0x0	grf_emaw_l2d Extra margin adjustment for write access
2:0	RW	0x1	grf_ema_l2d Extra margin adjustment

GRF_CPU_STATUS0

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:15	RW	0x00	grf_st_cpu_boost_fsm Indicates the status of CPU boost state, detail description refers to CRU (CPU boost)
14	RO	0x0	grf_st_l2flushdone High indicates L2 hardware flush complete
13	RO	0x0	grf_st_clrexmonack High indicates the ack from clrexmonreq
12	RO	0x0	grf_st_jtagnsw Current TAP mode of operation. High if JTAG selected, low if SWD selected SWJ-DP
11	RO	0x0	grf_st_jtagtop JTAG TAP controller in one of top 4 states (TLR, RTI, Sel-DR or Sel-IR)
10	RO	0x0	evento_rising_edge When event output rising edge is detected, this bit keeps high until set evento_clear to clear this bit
9:4	RO	0x0	reserved
3:0	RO	0x0	grf_st_smpnamp Indicate whether a core is taking part in coherency

GRF_CPU_STATUS1

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	grf_st_standbywfil2 Indicates whether the L2 memory system is in WFI low-power state. This signal is active when the following conditions are met. 1. All cores are in WFI low-power state, held in reset, or nL2RESET is asserted LOW. 2. If ACE has been configured, ACINACTM is asserted high. 3. If ACP has been configured, AINACTS is asserted high. 4. If CHI has been configured, SINACT is asserted high. 5. L2 memory system is idle. The system power controller must not remove power from the processor when this signal is LOW. This restriction includes configurations where you have set the L2_CACHE_PRESENT parameter to 0
11:8	RO	0x0	reserved
7:4	RO	0x0	grf_st_standbywfi Indicates whether a core is in WFI low-power state. 1'b0: Core not in WFI low-power state 1'b1: Core in WFI low-power state. This is the reset condition The system power controller must not remove power from an individual core when the corresponding bit of this signal is low
3:0	RO	0x0	grf_st_standbywfe Indicates whether a core is in WFE low-power state. 1'b0: Core not in WFE low-power state 1'b1: Core in WFE low-power state

GRF_PVTM_CON0

Address: Operational Base + offset (0x0440)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:2	RW	0x000	pvtm_pmu_clkout_div Clock divider setting of pvtm output clock, the 12bit divider is controlled by pvtm_clkout_div,2'b0

Bit	Attr	Reset Value	Description
1	RW	0x0	pvtm_pmu_osc_en Set high to enable the osc_ring in the PVTM
0	RW	0x0	pvtm_pmu_start Set high to start pmu pvtm

GRF_PVTM_CON1

Address: Operational Base + offset (0x0444)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pvtm_pmu_cal_cnt PMU pvtm calculation counter

GRF_PVTM_STATUS0

Address: Operational Base + offset (0x0448)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	pvtm_pmu_freq_done High indicates pmu pvtm frequency count done

GRF_PVTM_STATUS1

Address: Operational Base + offset (0x044c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pvtm_pmu_freq_cnt Indicates the cycle counts of the osc ring clock

GRF_TSADC_TESTBIT_L

Address: Operational Base + offset (0x0460)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:0	RW	0x0000	grf_tsadc_testbit_l TSADC test low bits

GRF_TSADC_TESTBIT_H

Address: Operational Base + offset (0x0464)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:0	RW	0x0000	grf_tsadc_testbit_h TSADC test high bits

GRF_USB2_HOST0_CON0
Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:12	RO	0x0	reserved
11:6	RW	0x20	grf_con_host0_fladj_val_common Frame Length Adjustment Register per Port. This value must be the same as that of grf_con_host0_fladj_val.
5:0	RW	0x20	grf_con_host0_fladj_val Frame Length Adjustment Register. This feature adjusts any offset from the clock source that drives the uSOF counter. The uSOF cycle is equal to 59,488 plus this value. The default value is 0x20, which gives an SOF cycle time of 60,000.

GRF_USB2_HOST0_CON1
Address: Operational Base + offset (0x0484)

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

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14	RW	0x0	grf_con_usbphy_host_port_wakeup_irq_en USB 2.0 PHY host port wake up interrupt enable control. 1'b0: disable 1'b1: enable
13	RW	0x0	grf_con_host0_arb_pause USB 2.0 HOST0 ehci/ohci arbiter pause control.
12	RW	0x0	grf_con_host0_ohci_susp_lgcy Static OCHI clock control signal. Must be 0 if the OHCI 48/12 Mhz clocks must be suspended when EHCI and OHCI controllers are not active.
11	RW	0x0	grf_con_host0_ohci_cntsel Selects the counter value for simulation or real time for 1ms. 1'b1: Simulation time 1'b0: Count full 1 ms
10	RW	0x1	grf_con_host0_ohci_clkcktrst Initial reset signal for the DPLL. This is needed only for simulation.
9	RW	0x0	grf_con_host0_app_prt_ovrcur Port overcurrent indication from application. When set to 1, the corresponding port enters Disable state.
8	RW	0x0	grf_con_host0_autoppd_on_overcur_en Auto port power disable on overcurrent bit. This bit enables automatic port power disable in the host controller.
7	RW	0x1	grf_con_host0_word_if Selects the data width of the UTMI/UTMI+ PHY interface. 1'b1: 16-bit interface 1'b0: 8-bit interface
6	RW	0x0	grf_con_host0_sim_mode Used only for simulation. When set to 1'b1, this bit sets the PHY in a non-driving mode so the EHCI can detect device connection.
5	RW	0x1	grf_con_host0_incrx_en Burst alignment enable. 1'b1: Start INCRX burst only on burst x-aligned addresses 1'b0: Normal AHB operation; start bursts on any double word boundary
4	RW	0x1	grf_con_host0_incr8_en AHB burst type INCR8 enable.
3	RW	0x1	grf_con_host0_incr4_en AHB burst type INCR4 enable.
2	RW	0x1	grf_con_host0_incr16_en AHB burst type INCR16 enable.

Bit	Attr	Reset Value	Description
1	RW	0x0	grf_con_host0_hubsetup_min Hub setup time control signal. 1'b1: four FS clocks of hub setup time is used 1'b0: five FS clocks of hub setup time is used
0	RW	0x0	grf_con_host0_app_start_clk OHCI Clock control signal. This is an asynchronous primary input to the host core. When the OHCI clocks are suspended, the system has to assert this signal to start the clocks (12 and 48 Mhz). This should be deasserted after the clocks are started and before the host is suspended again.

GRF_USB2_OTG_CON0

Address: Operational Base + offset (0x0488)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:3	RO	0x0	reserved
2	RW	0x0	otg_dbnce_fltr_bypass OTG signals debounce filter bypass.
1:0	RW	0x0	otg_scaledown_mode Scale-Down mode selects. resulting in faster simulations. 2'b00: Disables all scale-downs 2'b01: Enables scale-down of all timing values except Device mode suspend and resume 2'b10: Enables scale-down of Device mode suspend and resume timing values only 2'b11: Enables bit 0 and bit 1 scale-down timing values

GRF_USB2_HOST0_STATUS0

Address: Operational Base + offset (0x048c)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RO	0x0	host0_ehci_power_state_ack This signal indicates the power state change acknowledgement from the EHCI to the PCI for PCI power management.
29	RO	0x0	host0_ehci_pme_status This bit displays the PME status.

Bit	Attr	Reset Value	Description
28	RO	0x0	grf_stat_host0_ehci_bufacc EHCI buffer access. This signal asserted whenever the host controller does a data read/write transfer.
27	RO	0x0	grf_stat_host0_ehci_xfer_prdc Indicates that the current transfer of the EHCI master on the AHB bus belongs to periodic descriptor/data.
26	RO	0x0	grf_stat_host0_ohci_ccs Current connect status. When set, this bit indicates that the port state machine is in a connected state.
25	RO	0x0	grf_stat_host0_ohci_rwe Remote wake up enable. This signal is brought out as a status signal and can be ignored by normal host controller operation.
24	RO	0x0	grf_stat_host0_ohci_drwe Device remote wake-up enable. When active, it causes the host controller to treat a connect or disconnect event as a remote wake-up.
23	RO	0x0	grf_stat_host0_ohci_globalsuspend This signal is asserted 5 ms after the host controller enters the USB Suspend state.
22	RO	0x0	grf_stat_host0_ohci_bufacc When active, this signal indicates that the host controller is currently accessing the data buffer indicated by the TD.
21	RO	0x0	grf_stat_host0_ohci_rmtwkp OHCI remote wakeup status.
20:17	RO	0x0	grf_stat_host0_ehci_lpsmc_state This signal indicates the state of the LPSMC module and used only for debugging.
16:11	RO	0x00	grf_stat_host0_ehci_usbsts This signal indicates pending interrupts and various host controller statuses. These 6 bits reflect the value in USBSTS[5:0] register.
10:0	RO	0x000	grf_stat_host0_ehci_xfer_cnt Transfer byte count from the EHCI master of the current AHB transaction. This is a constant signal and its value is valid when the EHCI starts its AHB address phase.

GRF_MAC_CON0

Address: Operational Base + offset (0x04a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:5	RO	0x0	reserved
4:2	RW	0x0	grf_con_mac2io_phy_intf_sel PHY interface select. 3'b001: RGMII 3'b100: RMII Others: Reserved
1	RW	0x0	grf_con_mac2io_flowctrl MAC transmit flow control. When set high, instructs the MAC to transmit PAUSE Control frame in Full-duplex mode. In Half-duplex mode, the MAC enables the Back-pressure function until this signal is made low again
0	RW	0x0	grf_mac2io_mac_speed MAC speed selection. 1'b1:100-Mbps 1'b0:10-Mbps

GRF_UPCTL_CON0

Address: Operational Base + offset (0x04a4)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:9	RO	0x0	reserved
8	RW	0x0	dfi_init_start_sel DDR phy dfi_init_start source selection. 1'b0: upctl controls dfi_init_start 1'b1: grf dfi_init_start controls DDR phy dfi_init_start
7	RW	0x0	grf_dfi_init_start Software control dfi_init_start bit

Bit	Attr	Reset Value	Description
6	RW	0x0	csysreq_upctl_pmu PMU control upctl csysreq_ddrc enable. 1'b0: disable pmu controls upctl csysreq_ddrc 1'b1: enable pmu controls upctl csysreq_ddrc
5	RW	0x0	csysreq_upctl_ddrstdby Standby control upctl csysreq_ddrc enable. 1'b0: disable standby controls upctl csysreq_ddrc 1'b1: enable standby controls upctl csysreq_ddrc
4	RW	0x0	upctl_c_active_in System low-power active_in
3	RW	0x0	upctl_anfifo Use NFIFO logic. Used to tell uPCTL to use NFIFO module or not. Only useful if NFIFO module is included
2	RW	0x0	upctl_aburstint Burst Interrupt opportunity, rather than Burst terminate(BST). Used to tell uPCTL not to schedule a BST. Only useful in mDDR or LPDDR2
1	RW	0x0	upctl_lp_reset_mode UPCTL low power reset mode. Select between which uPCTL Operational State to reset to: 1'b0: Init_mem 1'b1: Low_power Related output signals and assumed defaults of input signals will differ depending on which state uPCTL is reset to. This signal should only change during system reset (both n_rst_n=0 and p_rst_n=0)
0	RW	0x1	ddr_16bit_en DDR 16 bit enable control. 1'b0: disable 1'b1: enable

GRF_UPCTL_STATUS0

Address: Operational Base + offset (0x04a8)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	fw_req_pwrDiscTarg_pwrActive 1'b0: fw request transmission is idle 1'b1: fw request transmission is active
29	RO	0x0	cpu2servicevoiceslv_req_pwrDiscTarg_pwrActive 1'b0: cpu2servicevoiceslv request transmission is idle 1'b1: cpu2servicevoiceslv request transmission is active
28	RO	0x0	bus2voiceslv_req_pwrDiscTarg_pwrActive 1'b0: bus2voiceslv request transmission is idle 1'b1: bus2voiceslv request transmission is active

Bit	Attr	Reset Value	Description
27	RO	0x0	bus2peri_req_pwrDiscTarg_pwrActive 1'b0: bus2peri request transmission is idle 1'b1: bus2peri request transmission is active
26	RO	0x0	cpu2bus_req_pwrDiscTarg_pwrActive 1'b0: cpu2bus request transmission is idle 1'b1: cpu2bus request transmission is active
25	RO	0x0	cpu2srvmsch_req_pwrDiscTarg_pwrActive 1'b0: cpu2srvmsch request transmission is idle 1'b1: cpu2srvmsch request transmission is active
24	RO	0x0	bus2msch_req_pwrDiscTarg_pwrActive 1'b0: bus2msch request transmission is idle 1'b1: bus2msch request transmission is active
23	RO	0x0	peri2msch_req_pwrDiscTarg_pwrActive 1'b0: peri2msch request transmission is idle 1'b1: peri2msch request transmission is active
22	RO	0x0	cpu2msch_req_pwrDiscTarg_pwrActive 1'b0: cpu2msch request transmission is idle 1'b1: cpu2msch request transmission is active
21	RO	0x0	dfi_scramble_key_ready High indicates scramble key is ready
20	RO	0x0	upctrl_c_active Clock active. Active state is High
19	RO	0x0	upctrl_c_sysack Low-power request acknowledgement. Active state is Low
18:16	RO	0x0	grf_ddrupctl_stat Current state of the uPCTL. Value of the STAT.ctl_stat register bit field is driven on this signal
15:0	RO	0x0000	ddrupctl_bbflags Bank busy indication. 1'b0: idle 1'b1: busy

GRF_OS_REG0

Address: Operational Base + offset (0x0500)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg0 OS register 0, reset only by pmurstn

GRF_OS_REG1

Address: Operational Base + offset (0x0504)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg1 OS register 1, reset only by pmurstn

GRF_OS_REG2

Address: Operational Base + offset (0x0508)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg2 OS register 2, reset only by pmurstn

GRF_OS_REG3

Address: Operational Base + offset (0x050c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg3 OS register 3, reset only by pmurstn

GRF_OS_REG4

Address: Operational Base + offset (0x0510)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg4 OS register 4, reset only by pmurstn

GRF_OS_REG5

Address: Operational Base + offset (0x0514)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg5 OS register 5, reset only by pmurstn

GRF_OS_REG6

Address: Operational Base + offset (0x0518)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg6 OS register 6, reset only by pmurstn

GRF_OS_REG7

Address: Operational Base + offset (0x051c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg7 OS register 7, reset only by pmurstn

GRF_OS_REG8

Address: Operational Base + offset (0x0520)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg8 OS register 8, once these bits are written, they can't be reset

GRF_OS_REG9

Address: Operational Base + offset (0x0524)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg9 OS register 9, once these bits are written, they can't be reset

GRF_OS_REG10

Address: Operational Base + offset (0x0528)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg10 OS register 10, once these bits are written, they can't be reset

GRF_OS_REG11

Address: Operational Base + offset (0x052c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_os_reg11 OS register 11, once these bits are written, they can't be reset

GRF_SOC_CON12

Address: Operational Base + offset (0x0600)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:3	RO	0x0	reserved
2	RW	0x0	grf_gpio2a4_pad_out_src_sel This bit is multiplex IO control for the output of the gpio2_a4 after IOMUX function to provide another same pdm_clk source with gpio2_a6. 1'b0: normal gpio2_a4 function 1'b1: gpio2_a4 output the same clock source from gpio2_a6 IOMUX, and the output enable is controlled by gpio0_b1 pmic_sleep function output
1	RW	0x0	grf_noc_msch_main_partialpop The value is corresponding to DDR config. High indicates 16bit
0	RW	0x0	grf_noc_msch_mainddr3 The value is corresponding to DDR config, if external memory type is DDR3, should set this bit high

GRF_CHIP_ID

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0cea	chip_id Chip ID is 32'h3306, but chip mark as 32'h3308. Errata: The read only chip id value is 32'h0cea(32'd3306)

30.4.4 USB PHY GRF Registers Summary

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

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

30.4.5 USB PHY GRF Detail Register Description

USBPHY_GRF_REG0

Address: Operational Base + offset (0x0000)

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

USBPHY_GRF_REG1

Address: Operational Base + offset (0x0004)

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

USBPHY_GRF_REG2

Address: Operational Base + offset (0x0008)

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

USBPHY_GRF_REG3

Address: Operational Base + offset (0x000c)

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

USBPHY_GRF_REG4

Address: Operational Base + offset (0x0010)

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

USBPHY_GRF_REG5

Address: Operational Base + offset (0x0014)

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

USBPHY_GRF_REG6

Address: Operational Base + offset (0x0018)

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

USBPHY_GRF_REG7

Address: Operational Base + offset (0x001c)

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

USBPHY_GRF_REG8

Address: Operational Base + offset (0x0020)

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

USBPHY_GRF_REG9

Address: Operational Base + offset (0x0024)

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

USBPHY_GRF_REG10

Address: Operational Base + offset (0x0028)

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

USBPHY_GRF_REG11

Address: Operational Base + offset (0x002c)

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

USBPHY_GRF_REG12

Address: Operational Base + offset (0x0030)

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

USBPHY_GRF_REG13

Address: Operational Base + offset (0x0034)

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

USBPHY_GRF_REG14

Address: Operational Base + offset (0x0038)

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

USBPHY_GRF_REG15

Address: Operational Base + offset (0x003c)

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

USBPHY_GRF_REG16

Address: Operational Base + offset (0x0040)

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

USBPHY_GRF_REG17

Address: Operational Base + offset (0x0044)

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

USBPHY_GRF_REG18

Address: Operational Base + offset (0x0048)

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

USBPHY_GRF_REG19

Address: Operational Base + offset (0x004c)

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

USBPHY_GRF_REG20

Address: Operational Base + offset (0x0050)

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

USBPHY_GRF_REG21

Address: Operational Base + offset (0x0054)

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

USBPHY_GRF_REG22

Address: Operational Base + offset (0x0058)

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

USBPHY_GRF_REG23

Address: Operational Base + offset (0x005c)

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

USBPHY_GRF_CON0

Address: Operational Base + offset (0x0100)

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

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

USBPHY_GRF_CON1

Address: Operational Base + offset (0x0104)

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

USBPHY_GRF_CON2

Address: Operational Base + offset (0x0108)

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

USBPHY_GRF_CON3

Address: Operational Base + offset (0x010c)

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

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

USBPHY_GRF_STATUS

Address: Operational Base + offset (0x0120)

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

Bit	Attr	Reset Value	Description
15	RO	0x0	usbhost_utmi_sessend host_utmi_sessend status
14	RO	0x0	usbhost_utmi_vbusvalid host_utmi_vbusvalid status
13	RO	0x0	usbhost_utmi_vmi host_utmi_vmi status
12	RO	0x0	usbhost_utmi_vpi host_utmi_vpi status
11	RO	0x0	usbotg_phy_ls_fs_rcv utmi_phy_ls_fs_rcv_out status
10	RO	0x0	usbotg_utmi_avalid otg_utmi avalid bit status
9	RO	0x0	usbotg_utmi_bvalid otg_utmi bvalid bit status
8	RO	0x0	usbotg_utmi_fs_xver_own OTG utmi_fs_xver_own status
7	RO	0x0	usbotg_utmi_hostdisconnect otg_utmi_hostdisconnect status
6	RO	0x0	usbotg_utmi_iddig usbotg_utmi_iddig status
5:4	RO	0x0	usbotg_utmi_linestate otg_utmi_linestate status
3	RO	0x0	usbotg_utmi_sessend otg_utmi_sessend bit status
2	RO	0x0	usbotg_utmi_vbusvalid otg_utmi_vbusvalid bit status
1	RO	0x0	usbotg_utmi_vmi otg_utmi_vmi bit status
0	RO	0x0	usbotg_utmi_vpi otg_utmi_vpi bit status

30.4.6 DETECT GRF Registers Summary

Name	Offset	Size	Reset Value	Description
DETECT_GRF_SDMMC_DETECT_COUNTER	0x0000	W	0x00030d40	SDMMC detect counter
DETECT_GRF_SDMMC_DETECT_CON	0x0004	W	0x00000000	SDMMC detect control
DETECT_GRF_SDMMC_DETECT_STATUS	0x0008	W	0x00000000	SDMMC detect interrupt status
DETECT_GRF_SDMMC_DETECT_STATUS_CLR	0x000c	W	0x00000000	SDMMC detect interrupt status clear.

Name	Offset	Size	Reset Value	Description
DETECT_GRF_USB2_DISCONNECT_CON	0x0010	W	0x00030d40	USB 2.0 PHY disconnect filter counter control
DETECT_GRF_USB2_LINESTATE_CON	0x0014	W	0x00030d40	USB 2.0 PHY linestate filter counter control
DETECT_GRF_USB2_BVALID_CON	0x0018	W	0x00030d40	USB 2.0 PHY bvalid filter counter control
DETECT_GRF_USB2_ID_CON	0x001c	W	0x00030d40	USB 2.0 PHY id filter counter control
DETECT_GRF_USB2_DETECT_IRQ_ENABLE	0x0020	W	0x00000000	USB 2.0 PHY connect detection interrupt request enable control
DETECT_GRF_USB2_DETECT_IRQ_STATUS	0x0024	W	0x00000000	USB 2.0 PHY connect detection interrupt status
DETECT_GRF_USB2_DETECT_IRQ_STATUS_CLR	0x0028	W	0x00000000	USB 2.0 PHY connect detection interrupt status clear

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

30.4.7 DETECT GRF Detail Register Description

DETECT_GRF_SDMMC_DETECT_COUNTER

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00030d40	sdmmc_detectn_count SDMMC detect filter counter time control, used for sdmmc card detecting, counter clock source is detect_grf pclk.

DETECT_GRF_SDMMC_DETECT_CON

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:2	RO	0x0	Reserved
1	RW	0x0	sdmmc_detectn_neg_irq_enable Enable sdmmc detectn negedge irq. 1'b0: disable 1'b1: enable
0	RW	0x0	sdmmc_detectn_pos_irq_enable Enable sdmmc detectn posedge irq. 1'b0: disable 1'b1: enable

DETECT_GRF_SDMMC_DETECT_STATUS

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	Reserved
1	RO	0x0	sdmmc_detectn_neg_irq SDMMC detectn interrupt status. 1'b0: no interrupt 1'b1: card plug in irq
0	RO	0x0	sdmmc_detectn_pos_irq SDMMC detectn interrupt status. 1'b0: no interrupt 1'b1: card plug out irq

DETECT_GRF_SDMMC_DETECT_STATUS_CLR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	Reserved
1	WO	0x0	sdmmc_detectn_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
0	WO	0x0	sdmmc_detectn_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status

DETECT_GRF_USB2_DISCONNECT_CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00030d40	disconnect_filter_con USB 2.0 host and OTG port host disconnect filter time control, counter clock source is detect_grf pclk.

DETECT_GRF_USB2_LINESTATE_CON

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00030d40	linestate_filter_con USB 2.0 host and OTG port linestate filter time control register, counter clock source is detect_grf pclk.

DETECT_GRF_USB2_BVALID_CON

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00030d40	bvalid_filter_con USB 2.0 OTG port bvalid filter time control register, counter clock source is detect_grf pclk.

DETECT_GRF_USB2_ID_CON

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00030d40	id_filter_con USB2.0 OTG port id filter time control register, counter clock source is detect_grf pclk.

DETECT_GRF_USB2_DETECT_IRQ_ENABLE

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:10	RO	0x0	reserved
9	RW	0x0	host0_disconnect_neg_irq_en Enable host 0 disconnect negedge detect interrupt. 1'b0: disable 1'b1: enable
8	RW	0x0	host0_disconnect_pos_irq_en Enable host 0 disconnect posedge detect interrupt. 1'b0: disable 1'b1: enable
7	RW	0x0	otg0_disconnect_neg_irq_en Enable OTG 0 disconnect negedge detect interrupt. 1'b0: disable 1'b1: enable
6	RW	0x0	otg0_disconnect_pos_irq_en Enable OTG 0 disconnect posedge detect interrupt. 1'b0: disable 1'b1: enable
5	RW	0x0	otg0_id_neg_irq_en Enable OTG 0 id negedge detect interrupt. 1'b0: disable 1'b1: enable
4	RW	0x0	otg0_id_pos_irq_en Enable OTG 0 id posedge detect interrupt. 1'b0: disable 1'b1: enable

Bit	Attr	Reset Value	Description
3	RW	0x0	otg0_bvalid_neg_irq_en Enable OTG 0 bvalid negedge detect interrupt. 1'b0: disable 1'b1: enable
2	RW	0x0	otg0_bvalid_pos_irq_en Enable OTG 0 bvalid posedge detect interrupt. 1'b0: disable 1'b1: enable
1	RW	0x0	host0_linestate_irq_en Enable host 0 linesate detect interrupt. 1'b0: disable 1'b1: enable
0	RW	0x0	otg0_linestate_irq_en Enable OTG 0 linesate detect interrupt. 1'b0: disable 1'b1: enable

DETECT_GRF_USB2_DETECT_IRQ_STATUS

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RO	0x0	host0_disconnect_neg_irq Host 0 disconnect negedge interrupt status. 1'b0: no irq 1'b1: irq asserted
8	RO	0x0	host0_disconnect_pos_irq Host 0 disconnect posedge interrupt status. 1'b0: no irq 1'b1: irq asserted
7	RO	0x0	otg0_disconnect_neg_irq OTG 0 disconnect negedge interrupt status. 1'b0: no irq 1'b1: irq asserted
6	RO	0x0	otg0_disconnect_pos_irq OTG 0 disconnect posedge interrupt status. 1'b0: no irq 1'b1: irq asserted
5	RO	0x0	otg0_id_neg_irq OTG 0 id negedge interrupt status. 1'b0: no irq 1'b1: irq asserted
4	RO	0x0	otg0_id_pos_irq OTG 0 id posedge interrupt status. 1'b0: no irq 1'b1: irq asserted

Bit	Attr	Reset Value	Description
3	RO	0x0	otg0_bvalid_neg_irq OTG 0 bvalid negedge interrupt status. 1'b0: no irq 1'b1: irq asserted
2	RO	0x0	otg0_bvalid_pos_irq OTG 0 bvalid posedge interrupt status. 1'b0: no irq 1'b1: irq asserted
1	RO	0x0	host0_linestate_irq Host 0 linestate change interrupt status. 1'b0: no irq 1'b1: irq asserted
0	RO	0x0	otg0_linestate_irq OTG 0 linestate change interrupt status. 1'b0: no irq 1'b1: irq asserted

DETECT_GRF_USB2_DETECT_IRQ_STATUS_CLR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	host0_disconnect_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
8	WO	0x0	host0_disconnect_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
7	WO	0x0	otg0_disconnect_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
6	WO	0x0	otg0_disconnect_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
5	WO	0x0	otg0_id_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
4	WO	0x0	otg0_id_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status

Bit	Attr	Reset Value	Description
3	WO	0x0	otg0_bvalid_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
2	WO	0x0	otg0_bvalid_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
1	WO	0x0	host0_linestate_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
0	WO	0x0	otg0_linestate_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status

30.4.8 CORE GRF Registers Summary

Name	Offset	Size	Reset Value	Description
CORE_GRF_CA35_PEFF_CON0	0x0000	W	0x00000000	CA35 performance monitor control register0
CORE_GRF_CA35_PEFF_CON1	0x0004	W	0x00000000	CA35 performance monitor control register1
CORE_GRF_CA35_PEFF_CON2	0x0008	W	0x00000000	CA35 performance monitor control register2
CORE_GRF_CA35_PEFF_CON3	0x000c	W	0x00000000	CA35 performance monitor control register3
CORE_GRF_CA35_PEFF_CON4	0x0010	W	0x00000000	CA35 performance monitor control register4
CORE_GRF_CA35_PEFF_CON5	0x0014	W	0x00000000	CA35 performance monitor control register5
CORE_GRF_CA35_PEFF_CON6	0x0018	W	0x00000000	CA35 performance monitor control register6
CORE_GRF_CA35_PEFF_CON7	0x001c	W	0x00000000	CA35 performance monitor control register7
CORE_GRF_CA35_PEFF_CON8	0x0020	W	0x00000000	CA35 performance monitor control register8
CORE_GRF_CA35_PERF_RD_MAX_LATENCY_NUM	0x0030	W	0x00000000	CA35 performance monitor status register
CORE_GRF_CA35_PERF_RD_LATENCY_SAMP_NUM	0x0034	W	0x00000000	CA35 performance monitor status register

Name	Offset	Size	Reset Value	Description
CORE_GRF_CA35_PERF_RD_LATENCY_ACC_NUM	0x0038	W	0x00000000	CA35 performance monitor status register
CORE_GRF_CA35_PERF_RD_AXI_TOTAL_BYTE	0x003c	W	0x00000000	CA35 performance monitor status register
CORE_GRF_CA35_PERF_WR_AXI_TOTAL_BYTE	0x0040	W	0x00000000	CA35 performance monitor status register
CORE_GRF_CA35_PERF_WORKING_CNT	0x0044	W	0x00000000	CA35 performance monitor status register
CORE_GRF_CA35_PERF_INT_STATUS	0x0048	W	0x00000000	CA35 performance monitor status register
CORE_GRF_COREPVTM_CON0	0x0080	W	0x00000000	CORE PVTM control register0
CORE_GRF_COREPVTM_CON1	0x0084	W	0x00000000	CORE PVTM control register1
CORE_GRF_COREPVTM_STATUS0	0x0088	W	0x00000000	CORE PVTM status register0
CORE_GRF_COREPVTM_STATUS1	0x008c	W	0x00000000	CORE PVTM status register1

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

30.4.9 CORE GRF Detail Register Description

CORE_GRF_CA35_PEFF_CON0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	ca35_sw_rd_latency_id_range_e AXI read channel id for latency AXI_PERFormance test.
14	RO	0x0	reserved
13:8	RW	0x00	ca35_sw_rd_latency_id 6'h0: 16-Byte align 6'h1: 32-Byte align 6'h2: 64-Byte align 6'h3: 128-Byte align

Bit	Attr	Reset Value	Description
7:6	RW	0x0	ca35_sw_ddr_align_type axi_perf counter id control. 2'b0: count all write channel id 2'b1: count sw_ar_count_id write channel only
5	RW	0x0	ca35_sw_aw_cnt_id_type axi_perf counter id control. 1'b0: count all write channel id 1'b1: count sw_aw_count_id read channel only
4	RW	0x0	ca35_sw_ar_cnt_id_type axi_perf counter id control. 1'b0: count all read channel id 1'b1: count sw_ar_count_id read channel only
3	RW	0x0	ca35_sw_axi_cnt_type_wrap axi_perf counter type wrap. 1'b0: no wrap test 1'b1: wrap test
2	RW	0x0	ca35_sw_axi_cnt_type axi_perf counter type. 1'b0: axi transfer test 1'b1: ddr align transfer test
1	RW	0x0	ca35_sw_axi_perf_clr axi_perf clear bit. 1'b0: disable 1'b1: enable
0	RW	0x0	ca35_sw_axi_perf_work axi_perf enable bit. 1'b0: disable 1'b1: enable

CORE_GRF_CA35_PEFF_CON1

Address: Operational Base + offset (0x0004)

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

CORE_GRF_CA35_PEFF_CON2
Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	ca35_sw_axi_perf_int_clr Interrupt clear. 1'b1: clear 1'b0: no op
14	RW	0x0	ca35_sw_axi_perf_int_e Interrupt enable. 1'b1: enable 1'b0: disable
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_aw_count_id When sw_aw_cnt_id_type=1, only count the sw_aw_count_id channel.
7:6	RO	0x0	reserved
5:0	RW	0x00	ca35_sw_ar_count_id When sw_ar_cnt_id_type=1, only count the sw_ar_count_id channel.

CORE_GRF_CA35_PEFF_CON3
Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	ca35_sw_ar_mon_id mon_id_bmsk bit control.
14	RW	0x0	ca35_sw_ar_mon_id_bmsk mon_id_bmsk bit control.
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_ar_mon_id_type mon_id_type bit control.

Bit	Attr	Reset Value	Description
7:6	RO	0x0	reserved
5:0	RW	0x00	ca35_sw_ar_mon_id_msk mon_id_msk bit control.

CORE_GRF_CA35_PEFF_CON4

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15	RW	0x0	ca35_sw_aw_mon_id mon_id_bmsk bit control.
14	RW	0x0	ca35_sw_aw_mon_id_bmsk mon_id_bmsk bit control.
13	RO	0x0	reserved
12:8	RW	0x00	ca35_sw_aw_mon_id_type mon_id_type bit control.
7:6	RO	0x0	reserved
5:0	RW	0x00	ca35_sw_aw_mon_id_msk mon_id_msk bit control.

CORE_GRF_CA35_PEFF_CON5

Address: Operational Base + offset (0x0014)

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

CORE_GRF_CA35_PEFF_CON6

Address: Operational Base + offset (0x0018)

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

CORE_GRF_CA35_PEFF_CON7

Address: Operational Base + offset (0x001c)

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

CORE_GRF_CA35_PEFF_CON8
Address: Operational Base + offset (0x0020)

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

CORE_GRF_CA35_PERF_RD_MAX_LATENCY_NUM
Address: Operational Base + offset (0x0030)

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

CORE_GRF_CA35_PERF_RD_LATENCY_SAMP_NUM
Address: Operational Base + offset (0x0034)

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

CORE_GRF_CA35_PERF_RD_LATENCY_ACC_NUM
Address: Operational Base + offset (0x0038)

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

CORE_GRF_CA35_PERF_RD_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x003c)

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

CORE_GRF_CA35_PERF_WR_AXI_TOTAL_BYTE
Address: Operational Base + offset (0x0040)

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

CORE_GRF_CA35_PERF_WORKING_CNT
Address: Operational Base + offset (0x0044)

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

CORE_GRF_CA35_PERF_INT_STATUS
Address: Operational Base + offset (0x0048)

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

CORE_GRF_COREPVTM_CON0
Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable When bit 16=1, bit 0 can be written by software. When bit 16=0, bit 0 can not be written by software. When bit 17=1, bit 1 can be written by software. When bit 17=0, bit 1 can not be written by software. ...... When bit 31=1, bit 15 can be written by software. When bit 31=0, bit 15 can not be written by software.
15:4	RO	0x0	reserved
3:2	RW	0x0	corepvtm_osc_sel osc_ring selection. 2'b00: osc_ring 0 2'b01: osc_ring 1 2'b10: osc_ring 2 Others: reserved
1	RW	0x0	corepvtm_osc_en Set high to enable the osc_ring in the PVTM.
0	RW	0x0	corepvtm_start Set high to start pmu PVTM.

CORE_GRF_COREPVTM_CON1
Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	corepvtm_cal_cnt PMU pvtm calculation counter.

CORE_GRF_COREPVTM_STATUS0
Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	corepvtm_freq_done High indicates pmu pvtm frequency count done.

CORE_GRF_COREPVTM_STATUS1
Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	corepvtm_freq_cnt Indicates the cycle counts of the osc ring clock.

30.5 Interface Description

NA

30.6 Application Notes

NA