

Rockchip
RK2206ABCD
TRM

Rockchip

Revision 1.0
Sep. 2019

Revision History

Date	Revision	Description
2019-9-12	1.0	Initial Release

Rockchip Confidential

Table of Content

Table of Content	3
Figure Index	9
Table Index.....	12
Warranty Disclaimer.....	13
Chapter 1 System Overview	14
1.1 Address Mapping	14
1.2 System Boot.....	16
1.3 System Interrupt Connection	17
1.4 System DMA Hardware Request Connection.....	20
Chapter 2 Clock & Reset Unit (CRU)	21
2.1 Overview.....	21
2.2 Block Diagram	21
2.3 Function Description	21
2.4 Register Description.....	23
2.5 Application Notes	78
Chapter 3 General Register Files (GRF)	82
3.1 Overview.....	82
3.2 Block Diagram	82
3.3 Function Description	82
3.4 Register Description.....	82
3.5 Interface Description	143
3.6 Application Notes	143
Chapter 4 Cortex-M4 with Cache.....	144
4.1 Overview.....	144
4.2 Block Diagram	146
4.3 Function Description	147
4.4 Register Description.....	149
4.5 Interface Description	155
4.6 Application Notes	155
Chapter 5 Cortex-M0.....	161
5.1 Overview.....	161
5.2 Block Diagram	162
5.3 Function Description	162
5.4 Register Description.....	162
5.5 Interface Description	162

5.6 Application Notes	163
Chapter 6 Power Management Unit (PMU)	165
6.1 Overview.....	165
6.2 Block Diagram	165
6.3 Function Description	165
6.4 Register Description.....	166
6.5 Timing Diagram	176
6.6 Application Notes	176
Chapter 7 I2S	180
7.1 Overview.....	180
7.2 Block Diagram	181
7.3 Function description	181
7.4 Register description	184
7.5 Interface Description	195
7.6 Application Notes	197
Chapter 8 I2C Interface.....	198
8.1 Overview.....	198
8.2 Block Diagram	198
8.3 Function Description	198
8.5 Interface Description	211
8.6 Application Notes	212
Chapter 9 LPO Trim Controller.....	216
9.1 Overview.....	216
9.2 Block Diagram	216
9.3 Function Description	216
9.5 Interface Description	220
9.6 Application Notes	221
Chapter 10 DSP.....	223
10.1 Overview	223
10.2 Block Diagram.....	223
10.3 Application Notes.....	224
Chapter 11 DMAC	225
11.1 Overview	225
11.2 Block Diagram.....	225
11.3 Register Description	226

11.4 Application Notes.....	292
Chapter 12 Mailbox.....	297
12.1 Overview	297
12.2 Block Diagram.....	297
12.3 Function Description	297
12.4 Register Description.....	298
12.5 Application Notes.....	309
Chapter 13 Internal SRAM	311
13.1 Overview	311
13.2 Block Diagram.....	311
13.3 Function Description	311
Chapter 14 Voice Activity Detect (VAD).....	313
14.1 Overview	313
14.2 Block Diagram.....	313
14.3 Function Description	314
14.4 Register Description.....	315
14.5 Application Notes.....	325
Chapter 15 WLAN Subsystem.....	328
15.1 Overview	328
15.2 Block Diagram.....	328
15.3 Function Description.....	329
15.4 Register Description.....	331
15.5 Interface Description.....	338
15.6 Application Notes.....	339
Chapter 16 Pulse Density Modulation Interface Controller	341
16.1 Overview	341
16.2 Block Diagram.....	341
16.3 Function Description	342
16.4 Register Description	343
16.5 Interface Description.....	350
16.6 Application Notes.....	350
Chapter 17 FSPI (Flexible Serial Peripheral Interface)	352
17.1 Overview	352
17.2 Block Diagram.....	352
17.3 Function Description	352

17.4 Register Description	353
17.5 Interface Description	377
17.6 Application Notes	379
Chapter 18 HyperBus	383
18.1 Overview	383
18.2 Block Diagram	383
18.3 Function Description	383
18.5 Interface Description	396
18.6 Application Notes	396
Chapter 19 SPI2APB	399
19.1 Overview	399
19.2 Block Diagram	399
19.3 Function Description	399
19.4 Register Description	402
19.5 Interface Description	404
19.6 Application Notes	404
Chapter 20 Serial Peripheral Interface (SPI)	405
20.1 Overview	405
20.2 Block Diagram	405
20.3 Function Description	406
20.4 Register Description	408
20.5 Interface Description	418
20.6 Application Notes	419
Chapter 21 UART	422
21.1 Overview	422
21.2 Block Diagram	422
21.3 Function Description	423
21.4 Register Description	426
21.5 Interface Description	441
21.6 Application Notes	444
Chapter 22 GPIO	447
22.1 Overview	447
22.2 Block Diagram	447
22.3 Function Description	447

22.4 Register Description	448
22.5 Interface Description	458
22.6 Application Notes.....	458
Chapter 23 Interrupt Controller	459
23.1 Overview	459
23.2 Block Diagram.....	459
23.3 Function Description	459
23.4 Register Description	460
23.5 Application Notes.....	465
Chapter 24 Timer	467
24.1 Overview	467
24.2 Block Diagram.....	467
24.3 Function Description	467
24.4 Register Description	468
24.5 Application Notes.....	471
Chapter 25 WDT	472
25.1 Overview	472
25.2 Block Diagram.....	472
25.3 Function Description	472
25.4 Register Description	474
25.5 Application Notes.....	476
Chapter 26 Pulse Width Modulation (PWM)	477
26.1 Overview	477
26.2 Block Diagram.....	477
26.3 Function Description	478
26.4 Register Description	479
Chapter 27 SARADC.....	502
27.1 Overview	502
27.2 Block Diagram.....	502
27.3 Function Description	502
27.4 Register description	502
27.5 Timing Diagram.....	505
27.6 Application Notes.....	506
Chapter 28 Temperature-Sensor ADC(TSADC)	507
28.1 Overview	507

28.2 Block Diagram.....	507
28.3 Function Description	507
28.5 Application Notes.....	514
Chapter 29 eFuse	517
29.1 Overview	517
29.2 Block Diagram.....	517
29.3 Function Description	517
29.4 Register Description.....	518
29.5 Timing Diagram.....	523
29.6 Application Notes.....	524
Chapter 30 Touch Key Controller.....	525
30.1 Overview	525
30.2 Block Diagram.....	525
30.3 Function Description	525
30.4 Register Description.....	526
30.5 Interface Description.....	548
30.6 Application Notes.....	550

Figure Index

Fig. 1-1 RK2206 boot procedure flow	17
Fig. 2-1 CRU Block Diagram	21
Fig. 2-2 PLL Block Diagram	22
Fig. 2-3 Reset Architecture Diagram	22
Fig. 2-4 Clock Architecture Diagram	23
Fig. 2-5 PLL setting change timing	79
Fig. 4-1 Block Diagram of Cortex-M4	146
Fig. 4-2 Block Diagram of Cache	146
Fig. 4-3 Block Diagram of Cortex-M4 with Cache	147
Fig. 4-4 System Address Map of the Cortex-M4	148
Fig. 5-1 Cortex-M0 Integration Architecture	162
Fig. 5-2 Block Diagram of Cortex-M0 with Cache	162
Fig. 5-3 M0 BootCode Map	163
Fig. 6-1 PMU Block Diagram	165
Fig. 6-2 Power Switch Timing	176
Fig. 6-3 PMU Info Timing	178
Fig. 6-4 PMU Control Register Relationship	179
Fig. 7-1 I2S/PCM controller (8 channel) Block Diagram	181
Fig. 7-2 I2S transmitter-master & receiver-slave condition	182
Fig. 7-3 I2S transmitter-slave & receiver-master condition	182
Fig. 7-4 I2S normal mode timing format	182
Fig. 7-5 I2S left justified mode timing format	182
Fig. 7-6 I2S right justified mode timing format	183
Fig. 7-7 PCM early mode timing format	183
Fig. 7-8 PCM late1 mode timing format	183
Fig. 7-9 PCM late2 mode timing format	184
Fig. 7-10 PCM late3 mode timing format	184
Fig. 7-11 I2S/PCM controller transmit operation flow chart	197
Fig. 8-1 I2C architecture	198
Fig. 8-2 I2C DATA Validity	201
Fig. 8-3 I2C Start and stop conditions	201
Fig. 8-4 I2C Acknowledge	201
Fig. 8-5 I2C byte transfer	201
Fig. 8-6 I2C Flow chat for transmit only mode	213
Fig. 8-7 I2C Flow chat for receive only mode	214
Fig. 8-8 I2C Flow chat for mix mode	215
Fig. 9-1 LPO Diagram	216
Fig. 9-2 LPO TRIM Diagram	216
Fig. 9-3 Calibration State Machine	217
Fig. 10-1 DSP Block Diagram	224
Fig. 11-1 Host Controller Block Diagram	225
Fig. 11-2 Multi-Block Transfer Using Linked Lists When DMAH_CHx_STAT_SRC Set toTrue	293
Fig. 11-3 Mapping of Block Descriptor (LLI) in Memory to Channel Registers	296
Fig. 12-1 Block Diagram of Mailbox	297
Fig. 13-1 System SRAM0/1 BUFFER TCM block diagram	311
Fig. 14-1 VAD Block Diagram	314
Fig. 14-2 vad_det Block Diagram	314
Fig. 14-3 VAD usage flow (only used ram write interface mode)	325
Fig. 14-4 VAD usage flow (use bus mode and ram write interface mode)	326
Fig. 15-1 WLAN Subsystem Block Diagram	329
Fig. 16-1 PDM Block Diagram	341
Fig. 16-2 PDM with two mono mic	342
Fig. 16-3 PDM with Four Stereo mic	342
Fig. 16-4 PDM interface diagram with external mic	342

Fig. 16-5 PDM Clock Structure	343
Fig. 16-6 PDM Operation Flow	351
Fig. 17-1 FSPI Architecture	352
Fig. 17-2 Program Flow	379
Fig. 17-3 Read Flow	380
Fig. 17-4 Command with DMA Flow	381
Fig. 17-5 SPI mode	382
Fig. 17-6 Idle cycles	382
Fig. 18-1 HyperBus Architecture	383
Fig. 18-2 HyperRAM Read Waveform	384
Fig. 18-3 HyperRAM Write Waveform	384
Fig. 18-4 Transaction Arbitration	397
Fig. 18-5 Separated Transaction	397
Fig. 18-6 Timing Adjustment	398
Fig. 19-1 SPI2APB Block Diagram	399
Fig. 19-2 Write operation	400
Fig. 19-3 Read operation	400
Fig. 19-4 Query operation	400
Fig. 19-5 Write message operation	401
Fig. 20-1 SPI Controller Block Diagram	406
Fig. 20-2 SPI Master and Slave Interconnection	406
Fig. 20-3 SPI Format (SCPH=0 SCPOL=0)	407
Fig. 20-4 SPI Format (SCPH=0 SCPOL=1)	407
Fig. 20-5 SPI Format (SCPH=1 SCPOL=0)	408
Fig. 20-6 SPI Format (SCPH=1 SCPOL=1)	408
Fig. 20-7 SPI Master transfer flow diagram	420
Fig. 20-8 SPI Slave transfer flow diagram	421
Fig. 21-1 UART Architecture	422
Fig. 21-2 UART Serial protocol	423
Fig. 21-3 IrDA 1.0	423
Fig. 21-4 UART baud rate	424
Fig. 21-5 UART Auto flow control block diagram	425
Fig. 21-6 UART AUTO RTS TIMING	425
Fig. 21-7 UART AUTO CTS TIMING	425
Fig. 21-8 UART none fifo mode	444
Fig. 21-9 UART fifo mode	445
Fig. 21-10 UART clock generation	446
Fig. 22-1 GPIO Block Diagram	447
Fig. 23-1 INTC Block Diagram	459
Fig. 23-2 IRQ interrupt processing for INTR	459
Fig. 24-1 Timer Block Diagram	467
Fig. 24-2 Timer Usage Flow	468
Fig. 24-3 Timing between timer_en and timer_clk	471
Fig. 25-1 WDT Block Diagram	472
Fig. 25-2 WDT Operation Flow (RMOD=1)	473
Fig. 26-1 PWM Block Diagram	477
Fig. 26-2 PWM Capture Mode	478
Fig. 26-3 PWM Continuous Left-aligned Output Mode	478
Fig. 26-4 PWM Continuous Center-aligned Output Mode	479
Fig. 26-5 PWM One-shot Center-aligned Output Mode	479
Fig. 27-1 RK2206SAR-ADC block diagram	502
Fig. 27-2 SAR-ADC timing diagram in single-sample conversion mode	505
Fig. 28-1 TSADC Controller Block Diagram	507
Fig. 28-2 the start flow to enable the sensor and adc	514
Fig. 28-3 tsadc timing diagram in bypass mode	514
Fig. 28-4 tsadc timing diagram in normal mode with tsadc_clk_sel = 1'b0	514
Fig. 28-5 tsadc timing diagram in normal mode with tsadc_clk_sel = 1'b1	515

Fig. 29-1 eFuse Block Diagram	517
Fig. 29-2 efuse128×8 timing diagram in program mode	524
Fig. 29-3 efuse128×8 timing diagram in read mode.....	524
Fig. 30-1 Host Controller Block Diagram.....	525
Fig. 30-2 Processor Detect mode	551
Fig. 30-3 Touch key controller Detect mode	551

Rockchip Confidential

Table Index

Table 1-1 RK2206 Address Mapping from 0x00000000 to 0x40110000	14
Table 1-2 RK2206 Address Mapping from 0x40110000 to 0xffffffff	15
Table 1-3 RK2206 remap function	16
Table 1-4 RK2206 Cortex-M4F Interrupt connection list	17
Table 1-5 RK2206 Cortex-M0 Interrupt connection list.....	19
Table 1-6 DMAC Request Mapping Table	20
Table 2-1 Source Clock Limitation of Fractional Divider.....	79
Table 4-1 Serial Wire Debug Interface Description	155
Table 4-2 Memory Address Map for the Cortex-M4.....	155
Table 4-3 IRQ Source ID of the NVIC.....	156
Table 4-4 Address Map for the Cache RAM Debug	159
Table 5-1 Cortex-M0 Interface Description.....	162
Table 5-2 M0 System Configure Signals	163
Table 5-3 M0 System Status Signals	163
Table 5-4 M0 Interrupt	164
Table 6-1 RK2206 Power Domain	165
Table 6-2 Low Power State	177
Table 6-3 Debug IO MUX	177
Table 7-1 I2S0 Group 0 Interface Description	195
Table 7-2 I2S0 Group 1 Interface Description	195
Table 8-1 I2C Interface Description	211
Table 9-1 Trim Table.....	221
Table 11-1 Programming of Transfer Types and Channel Register Update Method	294
Table 15-1 WLAN Subsystem embedded memory	329
Table 15-2 WLAN Interface Description	338
Table 16-1 Relation between PDM_CLK and sample rate	343
Table 16-4 Group 0 PDM IO Interface Description	350
Table 16-5 Group 1 PDM IO Interface Description	350
Table 17-1 FSPI Address Mapping Table	353
Table 17-2 FSPIO interface description	377
Table 17-3 FSPI1 interface description	378
Table 18-1 C/A Format of HyperBus.....	385
Table 18-2 HyperBus Address Mapping Table	385
Table 18-3 FSPI1 interface description	396
Table 19-1SPI2APB interface description	404
Table 20-1 SPI interface description	418
Table 21-1 UART Interface Description	441
Table 21-2 UART baud rate configuration.....	446
Table 22-1 GPIO Interface Description	458
Table 23-1 IRQ Source ID	465
Table 24-1 Register Base Address	471
Table 26-1 PWM Interface Description.....	498
Table 28-1 Temperature Code Mapping	515
Table 29-1 1024-bits Dout Format.....	517
Table 29-2 list of allowed modes	518
Table 30-1 Touch key Interface Description.....	548

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 expressedand patent or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Rockchip Electronics Co.,Ltd does not convey any license under its copyright and patent rights nor the rights of others.

All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.

Trademarks

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

Confidentiality

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

Reverse engineering or disassembly is prohibited.

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

Copyright © 2019 Rockchip Electronics Co.,Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Rockchip Electronics Co.,Ltd.

Chapter 1 System Overview

1.1 Address Mapping

RK2206 has two processors, Cortex-M4F and Cortex-M0. Cortex-M4F is the main core. When power up, Cortex-M4F boot from BootRoM first, and Cortex-M0 is default in reset state until Cortex-M4F release the reset signal of Cortex-M0.

RK2206 boot from internal BootROM, which supports remap function by software programming. Remap is controlled by GRF_SOC_CON0[0]. When remap is set to 1, the BootROM is mapped to address 0x0010000 and System SRAM0 is mapped to address 0x00000000.

Table 1-1 RK2206 Address Mapping from 0x00000000 to 0x40110000

Addr	IP	Addr	IP	Addr	IP
20000000				40110000	
	HyperBus(64M)/ FSPI1(64M)				MailBox0
18000000	128M			40100000	64K
	FSPI0(32M)				EFUSE_CTL0
10000000	128M			400f0000	64K
	Reserved				SPI1
00800000	248M			400e0000	64K
	DSP DTCM(192K)				SPI0
00600000	2M			400d0000	64K
	DSP ITCM(32K)				PWM2
00400000	2M			400c0000	64K
	Reserved				PWM1
00210000	1984K	40000000	HyperBus/FSPI1	400b0000	64K
	Reserved				PWM0
00200000	64K	38000000	128M	400a0000	64K
	Reserved		FSPI0		UART2
00120000	896K	30000000	128M	40090000	64K
	BOOTROM(16K)		Reserved		UART1
00100000	128K	20800000	248M	40080000	64K
	Reserved		DSP DTCM(192K)		UART0
000f0000	64K	20600000	2M	40070000	64K
	ICache0 ram debug(32K) Cache1 ram debug(32K)		DSP ITCM(32K)		I2C2
000e0000	64K	20400000	2M	40060000	64K
	BUFFER(96K)		Reserved		I2C1
000c0000	128K	200f0000	3136K	40050000	64K
	INTMEM1		DCache0 ram debug(32K)		I2C0
000a0000	128K	200e0000	64K	40040000	64K
	INTMEM0		Reserved		WDT2
00080000	128K	20060000	512K	40030000	64K

Addr	IP	Addr	IP	Addr	IP
	Reserved		BUFFER(96K)		WDT1
00021000	380K	20040000	128K	40020000	64K
	Reserved		INTMEM1		WDT0
00020000	4K	20020000	128K	40010000	64K
	BOOTROM(16K)		INTMEM0		TIMERO
00000000	128K	20000000	128K	40000000	64K

Table 1-2 RK2206 Address Mapping from 0x40110000 to 0xffffffff

Addr	IP	Addr	IP	Addr	IP
41000000		42000000			
	HyperBus config		Reserved		
40300000	13312K	41140000	15104K	b8000000	
	Reserved		VAD		WLAN GRAM
402b0000	320K	41130000	64K	b4000000	64M
	NOC Service		PDM 8ch		
40280000	192K	41120000	64K		
	Reserved		I2S1		
40270000	64K	41110000	64K		
	AUDPWM		I2S0		
40260000	64K	41100000	64K		
	VOP		Reserved		
40250000	64K	410b0000	320K		
	Cache1(16KB)		tsadc		
40240000	64K	410a0000	64K	44000000	
	Reserved		TouchKey controller		Reserved
40238000	32K	41090000	64K	43090000	15808K
	D-Cache0		PVTM		SPI2APB
40234000	16K	41080000	64K	43080000	64K
	I-Cache0		32K_TRIM		USB2 OTG
40230000	16K	41070000	64K	43040000	256K
	FSPI1 config		CRU		Reserved
40220000	64K	41060000	64K	43030000	64K
	FSPI0 config		GRF		CRYPTO
40210000	64K	41050000	64K	43020000	64K
	DMA		ACODEC PHY		SD/MMC
40200000	64K	41040000	64K	43010000	64K
	Reserved		TIMER1		VICAP
40140000	768K	41030000	64K	43000000	64K
	INT controller		GPIO1		Reserved
40130000	64K	41020000	64K	42080000	15872K
	SARADC		GPIO0		WLAN PBus Register
40120000	64K	41010000	64K	42040000	256K

Addr	IP		Addr	IP		Addr	IP
	MailBox1			PMU			WLAN SysBus Register
40110000	64K		41000000	64K		42000000	256K

The following table shows the BootROM /System SRAM0 address before and after remapping. Before remapping, 0x00080000 and 0x20000000 is the base address for System SRAM0, and 0x00000000 is the base address for BootROM. After remapping, BootROM could be accessed from 0x00100000, both 0x00080000, 0x20000000 and 0x00000000 are the base address for System SRAM0 and only 16KB System SRAM0 space could be accessed from 0x00000000 base address.

Table 1-3 RK2206 remap function

before remap		after remap	
	INTMEM0		INTMEM0
0x00080000 0x20000000	128K	0x00080000 0x20000000 0x00000000	128K
	BOOTROM(16K)		BOOTROM(16K)
0x00000000 0x00100000	128K	0x00100000	128K

M0 gets first instruction from 0x00000000. When M0 wants to access the address from 0x00000000 to 0x000003FF, M0 actually accesses the address from 0x000BFC00 to 0x000BFFFF.

1.2 System Boot

RK2206 provides system boot from off-chip devices such as Serial NOR Flash, SDMMC card. Also, the boot code can be programmed through USB2OTG interface or SPI2APB 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:
 - Serial NOR Flash, 1bit data width
 - SDMMC Card, 1bits data width
- Support system code download by USB2OTG
- Support system code download by SPI2APB

Following figure shows RK2206 boot procedure flow.

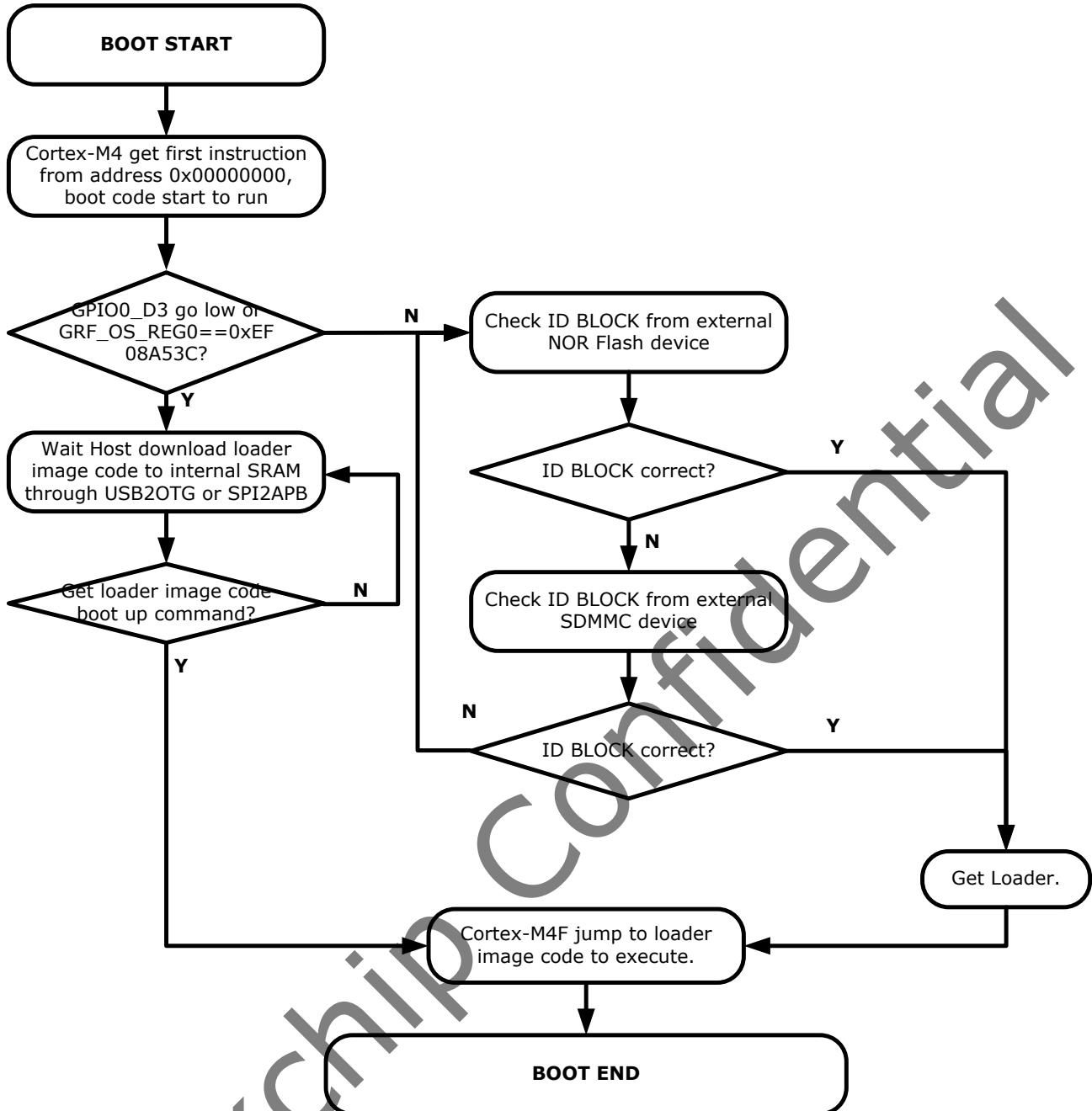


Fig. 1-1 RK2206 boot procedure flow

1.3 System Interrupt Connection

There is a Nested Vectored Interrupt Controller (NVIC) in embedded in Cortex-M4F, which has general interrupt sources for internal blocks or external devices. Each interrupts triggered type is high level, not programmable. The detailed interrupt sources connection is in the following table.

Table 1-4 RK2206 Cortex-M4F Interrupt connection list

Interrupt ID	Source	Polarity
0	dmac_ch0	High level
1	wdt0	High level
2	touchkey_irq_neg	High level
3	dmac_ch1	High level
4	timer0_ch0	High level

Interrupt ID	Source	Polarity
5	timer0_ch1	High level
6	timer0_ch2	High level
7	timer0_ch3	High level
8	timer0_ch4	High level
9	timer0_ch5	High level
10	timer1	High level
11	i2c0	High level
12	i2c1	High level
13	i2c2	High level
14	spi0	High level
15	spi2	High level
16	uart0	High level
17	uart1	High level
18	uart2	High level
19	pwm0	High level
20	pwm1	High level
21	pwm2	High level
22	saradc controller	High level
23	b2a0 irq0(mailbox0)	High level
24	b2a0 irq1(mailbox0)	High level
25	b2a0 irq2(mailbox0)	High level
26	b2a0 irq3(mailbox0)	High level
27	b2a1 irq0(mailbox1)	High level
28	b2a1 irq1(mailbox1)	High level
29	b2a2 irq2(mailbox1)	High level
30	b2a1 irq3(mailbox1)	High level
31	fspi0	High level
32	fspi1	High level
33	vop	High level
34	cache0_i	High level
35	cache0_d	High level
36	dsp error	High level
37	hyperbus	High level
38	usb2otg	High level
39	usb2otg bvalid	High level
40	usb2otg id	High level
41	usb2otg linestate	High level
42	usb2otg disconnect	High level
43	sdmmc	High level
44	adupwm	High level
45	pvtm	High level

Interrupt ID	Source	Polarity
46	crypto	High level
47	cif	High level
48	pmu	High level
49	gpio0	High level
50	gpio1	High level
51	trim	High level
52	i2s0	High level
53	i2s1	High level
54	pdm	High level
55	vad	High level
56	efuse controller	High level
57	touchkey_irq_pos	High level
58	pmic	High level
59	spi2apb	High level
60	dmac_ch2	High level
61	dmac_ch3	High level
62	dmac_ch4	High level
63	dmac_ch5	High level
64	tsadc controller	High level
65	wlan_write_trigger	High level
65-79	NA	High level

There is a Nested Vectored Interrupt Controller (NVIC) in embedded in Cortex-M0, which has general interrupt sources for internal blocks or external devices. Each interrupts triggered type is high level, not programmable. The detailed interrupt sources connection is in the following table.

Table 1-5 RK2206 Cortex-M0 Interrupt connection list

Interrupt ID	Source	Polarity
0	dmac ch0	High level
1	tsadc controller	High level
2	wdt1	High level
3	NA	High level
4	timer0_ch0	High level
5	timer0_ch1	High level
6	timer0_ch2	High level
7	timer0_ch3	High level
8	timer0_ch4	High level
9	timer0_ch5	High level
10	timer1	High level
11	uart0	High level
12	uart1	High level
13	uart2	High level

Interrupt ID	Source	Polarity
14	a2b0 irq0(mailbox0)	High level
15	a2b0 irq1(mailbox0)	High level
16	a2b0 irq2(mailbox0)	High level
17	a2b0 irq3(mailbox0)	High level
18	fspi0	High level
19	fspi1	High level
20	cache1	High level
21	hyperbus	High level
22	pmu	High level
23	wifi sleep	High level
24	wifi wakeup	High level
25	wifi ready	High level
26	wlan	High level
27	dmac_ch1	High level
28	dmac_ch2	High level
29	dmac_ch3	High level
30	dmac_ch4	High level
31	dmac_ch5	High level

1.4 System DMA Hardware Request Connection

RK2206 provides one 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 1-6 DMAC Request Mapping Table

Req number	Source	Polarity
0	UART0/1/2 tx	High level
1	UART0/1/2 rx	High level
2	UART0/1/2 tx	High level
3	UART0/1/2 rx	High level
4	SPI0 tx	High level
5	SPI0 rx	High level
6	SPI1 tx	High level
7	SPI1 rx	High level
8	PWM0/PWM1/PWM2	High level
9	audio_pwm	High level
10	I2S0 tx(I2S0 connect to IO)	High level
11	I2S0 rx	High level
12	I2S1 tx(I2S1 connect to codec inside)	High level
13	I2S1 rx	High level
14	PDM rx	High level
15	VOP	High level

Note: Req Number 0/1/2/3/8, please refer to GRF_SOC_CON17 in GRF chapter for more details.

Chapter 2 Clock & Reset Unit (CRU)

2.1 Overview

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

CRU supports the following features:

- Compliant to the AMBA APB interface
- Embedded 2 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

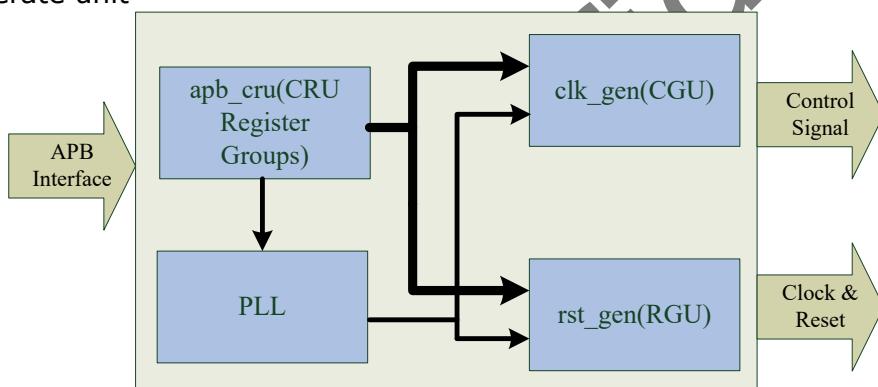


Fig. 2-1 CRU Block Diagram

2.3 Function Description

2.3.1 PLL

There are 2 PLLs in the chip: General PLL and Voice PLL.

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 all 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: 12MHz to 2.4GHz
- 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 or 2.5V) allows for excellent supply rejection in noisy SoC applications.
- Lock Detect Signal indicates when frequency lock has been achieved.

Following figure shows block diagram of PLL.

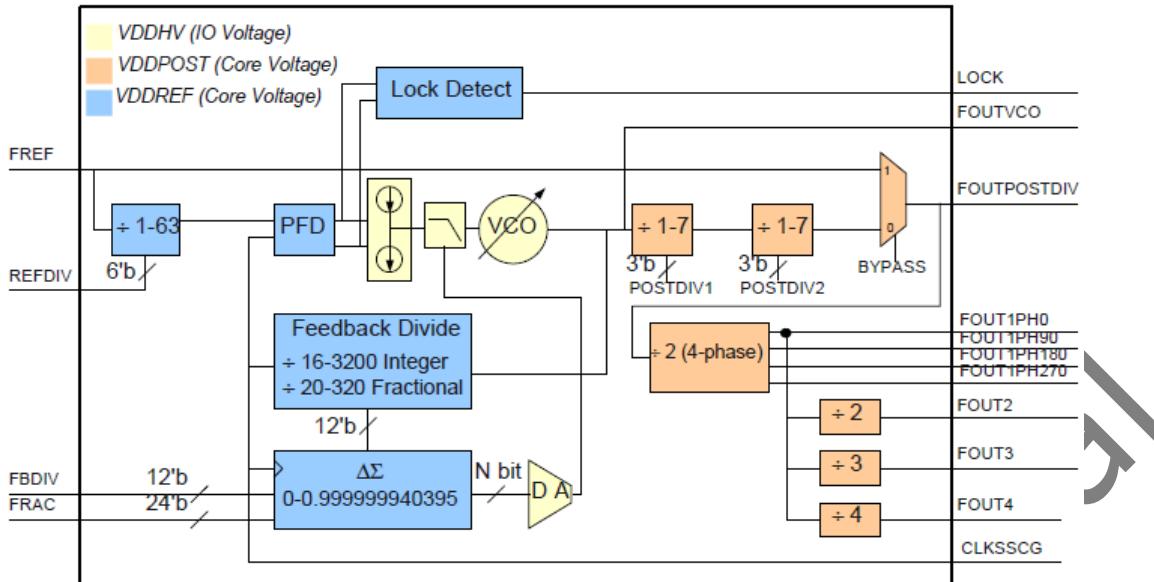


Fig. 2-2 PLL Block Diagram

The FREF (reference clock) of all PLL is xin_osc0 (40MHz 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.

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 is controlled by CRU registers.

2.3.2 Reset Generate Unit(RGU)

RGU generate the resets 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.

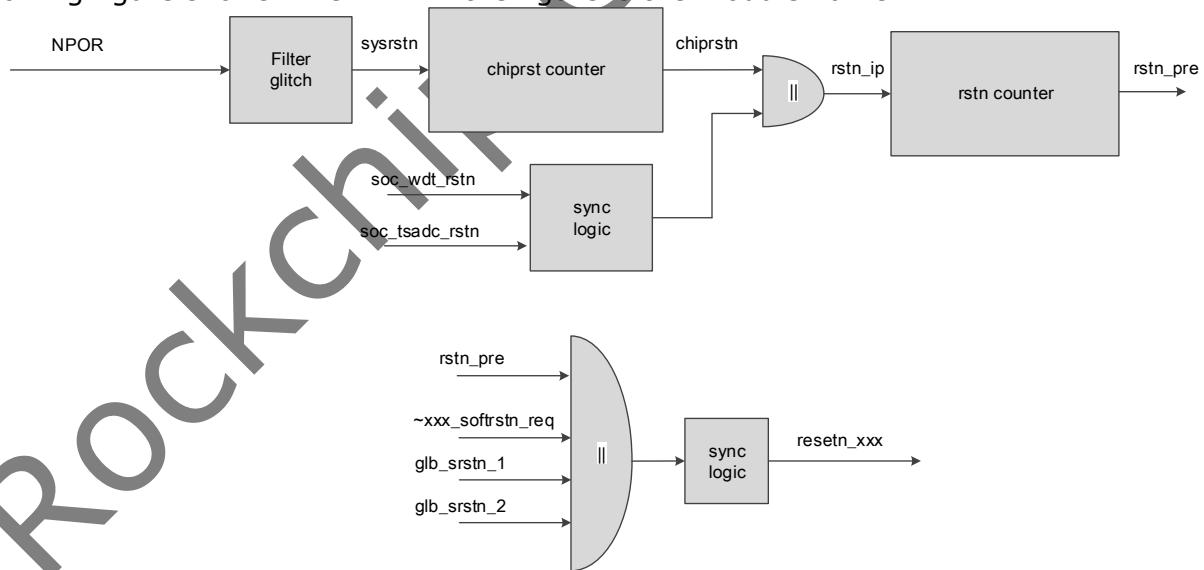


Fig. 2-3 Reset Architecture Diagram

The reset sources include:

- NPOR (Negative Power On Reset): hardware reset from IO or Internal POR circuit.
- 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 CRU_SOFRST_CONxxx. Each bit is separated to different module.
- globsrstn_1: First global software reset by programming CRU register CRU_GLB_SRST_FST as 0xfdb9
- globsrstn_2: Second global software reset by programming CRU register

CRU_GLB_SRST_SND as 0xecaa8

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

2.3.3 Clock Generate Unit(CGU)

CGU generates 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.

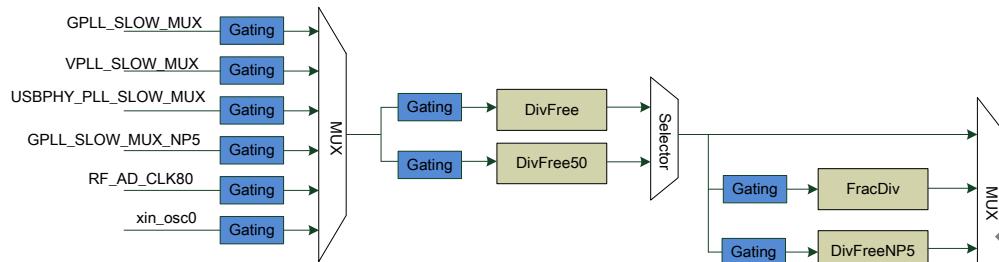


Fig. 2-4 Clock Architecture Diagram

The basic units are:

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

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

2.4 Register Description

2.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CRU_GPLL_CON00	0x0000	W	0x00003064	GPLL configuration register 0
CRU_GPLL_CON01	0x0004	W	0x00001041	GPLL configuration register 1
CRU_GPLL_CON02	0x0008	W	0x00000001	GPLL configuration register 2
CRU_GPLL_CON03	0x000c	W	0x00000007	GPLL configuration register 3
CRU_GPLL_CON04	0x0010	W	0x00007f00	GPLL configuration register 4
CRU_VPLL_CON00	0x0020	W	0x00002064	VPLL configuration register 0
CRU_VPLL_CON01	0x0024	W	0x00001041	VPLL configuration register 1
CRU_VPLL_CON02	0x0028	W	0x00000001	VPLL configuration register 2
CRU_VPLL_CON03	0x002c	W	0x00000007	VPLL configuration register 3
CRU_VPLL_CON04	0x0030	W	0x00007f00	VPLL configuration register 4
CRU_MODE_CON00	0x00a0	W	0x00000000	Internal clock select and divide register 0

Name	Offset	Size	Reset Value	Description
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_CLKSEL_CON00	0x0100	W	0x00000040	Internal clock select and divide register 0
CRU_CLKSEL_CON01	0x0104	W	0x00000000	Internal clock select and divide register 1
CRU_CLKSEL_CON02	0x0108	W	0x00000101	Internal clock select and divide register 2
CRU_CLKSEL_CON03	0x010c	W	0x00000142	Internal clock select and divide register 3
CRU_CLKSEL_CON04	0x0110	W	0x00000080	Internal clock select and divide register 4
CRU_CLKSEL_CON05	0x0114	W	0x00000000	Internal clock select and divide register 5
CRU_CLKSEL_CON06	0x0118	W	0x00000080	Internal clock select and divide register 6
CRU_CLKSEL_CON07	0x011c	W	0x00000000	Internal clock select and divide register 7
CRU_CLKSEL_CON08	0x0120	W	0x00000080	Internal clock select and divide register 8
CRU_CLKSEL_CON09	0x0124	W	0x00000000	Internal clock select and divide register 9
CRU_CLKSEL_CON10	0x0128	W	0x00000101	Internal clock select and divide register 10
CRU_CLKSEL_CON11	0x012c	W	0x00000101	Internal clock select and divide register 11
CRU_CLKSEL_CON12	0x0130	W	0x00008080	Internal clock select and divide register 12
CRU_CLKSEL_CON13	0x0134	W	0x00000380	Internal clock select and divide register 13
CRU_CLKSEL_CON14	0x0138	W	0x00008080	Internal clock select and divide register 14
CRU_CLKSEL_CON15	0x013c	W	0x00004080	Internal clock select and divide register 15
CRU_CLKSEL_CON16	0x0140	W	0x0000c0c0	Internal clock select and divide register 16
CRU_CLKSEL_CON17	0x0144	W	0x000000c0	Internal clock select and divide register 17

Name	Offset	Size	Reset Value	Description
CRU_CLKSEL_CON18	0x0148	W	0x00000003	Internal clock select and divide register 18
CRU_CLKSEL_CON19	0x014c	W	0x00000000	Internal clock select and divide register 19
CRU_CLKSEL_CON20	0x0150	W	0x00000301	Internal clock select and divide register 20
CRU_CLKSEL_CON21	0x0154	W	0x0000030f	Internal clock select and divide register 21
CRU_CLKSEL_CON22	0x0158	W	0x00000102	Internal clock select and divide register 22
CRU_CLKSEL_CON23	0x015c	W	0x00000303	Internal clock select and divide register 23
CRU_CLKSEL_CON24	0x0160	W	0x00000303	Internal clock select and divide register 24
CRU_CLKSEL_CON25	0x0164	W	0x00000001	Internal clock select and divide register 25
CRU_CLKSEL_CON26	0x0168	W	0x00000007	Internal clock select and divide register 26
CRU_CLKSEL_CON27	0x016c	W	0x00000000	Internal clock select and divide register 27
CRU_CLKSEL_CON28	0x0170	W	0x00000007	Internal clock select and divide register 28
CRU_CLKSEL_CON29	0x0174	W	0x00000000	Internal clock select and divide register 29
CRU_CLKSEL_CON30	0x0178	W	0x00000007	Internal clock select and divide register 30
CRU_CLKSEL_CON31	0x017c	W	0x00000000	Internal clock select and divide register 31
CRU_CLKSEL_CON32	0x0180	W	0x00000007	Internal clock select and divide register 32
CRU_CLKSEL_CON33	0x0184	W	0x00000000	Internal clock select and divide register 33
CRU_CLKSEL_CON34	0x0188	W	0x0000000f	Internal clock select and divide register 34
CRU_CLKSEL_CON35	0x018c	W	0x0000000f	Internal clock select and divide register 35
CRU_CLKSEL_CON36	0x0190	W	0x00000303	Internal clock select and divide register 36
CRU_GATE_CON00	0x0300	W	0x00000000	Internal clock gate and divide register 0
CRU_GATE_CON01	0x0304	W	0x00000000	Internal clock gate and divide register 1
CRU_GATE_CON02	0x0308	W	0x00000000	Internal clock gate and divide register 2

Name	Offset	Size	Reset Value	Description
CRU_GATE_CON03	0x030c	W	0x00000000	Internal clock gate and divide register 3
CRU_GATE_CON04	0x0310	W	0x00000000	Internal clock gate and divide register 4
CRU_GATE_CON05	0x0314	W	0x00000000	Internal clock gate and divide register 5
CRU_GATE_CON06	0x0318	W	0x00000000	Internal clock gate and divide register 6
CRU_GATE_CON07	0x031c	W	0x00000000	Internal clock gate and divide register 7
CRU_GATE_CON08	0x0320	W	0x00000000	Internal clock gate and divide register 8
CRU_GATE_CON09	0x0324	W	0x00000000	Internal clock gate and divide register 9
CRU_GATE_CON10	0x0328	W	0x00000000	Internal clock gate and divide register 10
CRU_GATE_CON11	0x032c	W	0x00000000	Internal clock gate and divide register 11
CRU_GATE_CON12	0x0330	W	0x00000000	Internal clock gate and divide register 12
CRU_SSCGTBL0_3	0x0380	W	0x00000000	SSCG external wave table register 0
CRU_SSCGTBL4_7	0x0384	W	0x00000000	SSCG external wave table register 1
CRU_SSCGTBL8_11	0x0388	W	0x00000000	SSCG external wave table register 2
CRU_SSCGTBL12_15	0x038c	W	0x00000000	SSCG external wave table register 3
CRU_SSCGTBL16_19	0x0390	W	0x00000000	SSCG external wave table register 4
CRU_SSCGTBL20_23	0x0394	W	0x00000000	SSCG external wave table register 5
CRU_SSCGTBL24_27	0x0398	W	0x00000000	SSCG external wave table register 6
CRU_SSCGTBL28_31	0x039c	W	0x00000000	SSCG external wave table register 7
CRU_SSCGTBL32_35	0x03a0	W	0x00000000	SSCG external wave table register 8
CRU_SSCGTBL36_39	0x03a4	W	0x00000000	SSCG external wave table register 9
CRU_SSCGTBL40_43	0x03a8	W	0x00000000	SSCG external wave table register 10
CRU_SSCGTBL44_47	0x03ac	W	0x00000000	SSCG external wave table register 11

Name	Offset	Size	Reset Value	Description
CRU_SSCGTBL48_51	0x03b0	W	0x00000000	SSCG external wave table register 12
CRU_SSCGTBL52_55	0x03b4	W	0x00000000	SSCG external wave table register 13
CRU_SSCGTBL56_59	0x03b8	W	0x00000000	SSCG external wave table register 14
CRU_SSCGTBL60_63	0x03bc	W	0x00000000	SSCG external wave table register 15
CRU_SSCGTBL64_67	0x03c0	W	0x00000000	SSCG external wave table register 16
CRU_SSCGTBL68_71	0x03c4	W	0x00000000	SSCG external wave table register 17
CRU_SSCGTBL72_75	0x03c8	W	0x00000000	SSCG external wave table register 18
CRU_SSCGTBL76_79	0x03cc	W	0x00000000	SSCG external wave table register 19
CRU_SSCGTBL80_83	0x03d0	W	0x00000000	SSCG external wave table register 20
CRU_SSCGTBL84_87	0x03d4	W	0x00000000	SSCG external wave table register 21
CRU_SSCGTBL88_91	0x03d8	W	0x00000000	SSCG external wave table register 22
CRU_SSCGTBL92_95	0x03dc	W	0x00000000	SSCG external wave table register 23
CRU_SSCGTBL96_99	0x03e0	W	0x00000000	SSCG external wave table register 24
CRU_SSCGTBL100_103	0x03e4	W	0x00000000	SSCG external wave table register 25
CRU_SSCGTBL104_107	0x03e8	W	0x00000000	SSCG external wave table register 26
CRU_SSCGTBL108_111	0x03ec	W	0x00000000	SSCG external wave table register 27
CRU_SSCGTBL112_115	0x03f0	W	0x00000000	SSCG external wave table register 28
CRU_SSCGTBL116_119	0x03f4	W	0x00000000	SSCG external wave table register 29
CRU_SSCGTBL120_123	0x03f8	W	0x00000000	SSCG external wave table register 30
CRU_SSCGTBL124_127	0x03fc	W	0x00000000	SSCG external wave table register 31
CRU_SOFRST_CON00	0x0400	W	0x00000000	Internal clock reset register 0
CRU_SOFRST_CON01	0x0404	W	0x00000070	Internal clock reset register 1
CRU_SOFRST_CON02	0x0408	W	0x00000000	Internal clock reset register 2
CRU_SOFRST_CON03	0x040c	W	0x00000000	Internal clock reset register 3

Name	Offset	Size	Reset Value	Description
CRU_SOFTRST_CON04	0x0410	W	0x00000006	Internal clock reset register 4
CRU_SOFTRST_CON05	0x0414	W	0x00000000	Internal clock reset register 5
CRU_SOFTRST_CON06	0x0418	W	0x00000000	Internal clock reset register 6
CRU_SOFTRST_CON07	0x041c	W	0x00000000	Internal clock reset register 7
CRU_SOFTRST_CON08	0x0420	W	0x00000000	Internal clock reset register 8
CRU_SOFTRST_CON09	0x0424	W	0x00000000	Internal clock reset register 9
CRU_SOFTRST_CON10	0x0428	W	0x00000000	Internal clock reset register 10
CRU_SOFTRST_CON11	0x042c	W	0x00000000	Internal clock reset register 11
CRU_SOFTRST_CON12	0x0430	W	0x00000000	Internal clock reset register 12
CRU_SDMMC_CON00	0x0480	W	0x00000004	SDMMC control register 0
CRU_SDMMC_CON01	0x0484	W	0x00000000	SDMMC control register 1

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_GPLL_CON00

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
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	fbdv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: No plus one operation

CRU_GPLL_CON01

Address: Operational Base + offset (0x0004)

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

Bit	Attr	Reset Value	Description
13	RW	0x0	pllpd PLL power down setting 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
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 GPLL CON02

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 GPLL CON03

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	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%
7:4	RW	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset SSMOD reset 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Disable sperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	RW	0x1	ssmod_bp SSMOD bypass 1'b0: No bypass 1'b1: Bypass

CRU GPL CON04

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid.
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU_VPLL_CON00

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	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	fbdv PLL feedback divide value: 16-3200 is valid in integer mode 20-320 is valid in fractional mode Tips: No plus one operation

CRU_VPLL_CON01

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	pllpd PLL power down setting 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

Bit	Attr	Reset Value	Description
9	RO	0x0	reserved
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 VPLL CON02

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 VPLL CON03

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	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	RW	0x0	ssmod_divval Divider of SSMOD required to set the modulation frequency
3	RW	0x0	ssmod_downspread Select center spread or down spread of SSMOD 1'b0: Down spread 1'b1: Center spread
2	RW	0x1	ssmod_reset SSMOD reset 1'b0: No reset 1'b1: Reset
1	RW	0x1	ssmod_disable_sscg Disable saperad spectrum clock generation of SSMOD, SSMOD is bypass when disable. 1'b0: Enable 1'b1: Disable
0	RW	0x1	ssmod_bp SSMOD bypass 1'b0: No bypass 1'b1: Bypass

CRU VPLL CON04

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x7f	ssmod_ext_maxaddr SSMOD maximum address of external wave table, 0-255 is valid.
7:1	RO	0x0	reserved
0	RW	0x0	ssmod_sel_ext_wave SSMOD select external wave table 1'b0: Don't select 1'b1: Select

CRU MODE CON00

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	clk_vpll_mode clk_vpll_mux clock MUX: 2'b00: xin_osc0_func 2'b01: clk_vpll 2'b10: clk_32k
3:2	RW	0x0	clk_usbpll_mode clk_usbpll_mux clock MUX: 2'b00: xin_osc0_func 2'b01: clk_usbpll 2'b10: clk_32k
1:0	RW	0x0	clk_gpll_mode clk_gpll_mux clock MUX: 2'b00: xin_osc0_func 2'b01: clk_gpll 2'b10: clk_32k

CRU GLB CNT TH

Address: Operational Base + offset (0x00b0)

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

CRU GLB RST ST

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	W1C	0x0	glb_wdt2_rst_st 1'b0: Last hot reset is not global reset triggered by WDT2 1'b1: Last hot reset is global reset triggered by WDT2
7	W1C	0x0	glb_wdt1_rst_st 1'b0: Last hot reset is not global reset triggered by WDT1 1'b1: Last hot reset is global reset triggered by WDT1
6	W1C	0x0	glb_wdt0_rst_st 1'b0: Last hot reset is not global reset triggered by WDT0 1'b1: Last hot reset is global reset triggered by WDT0
5	W1C	0x0	snd_glb_wdt_rst_st 1'b0: Last hot reset is not second global reset triggered by WDT 1'b1: Last hot reset is second global reset triggered by WDT
4	W1C	0x0	fst_glb_wdt_rst_st 1'b0: Last hot reset is not first global reset triggered by WDT 1'b1: Last hot reset is first global reset triggered by WDT

Bit	Attr	Reset Value	Description
3	W1C	0x0	snd_glb_tsadc_rst_st 1'b0: Last hot reset is not second global reset triggered by TSADC 1'b1: Last hot reset is second global reset triggered by TSADC
2	W1C	0x0	fst_glb_tsadc_rst_st 1'b0: Last hot reset is not first global reset triggered by TSADC 1'b1: Last hot reset is first global reset triggered by TSADC
1	W1C	0x0	snd_glb_RST_st 1'b0: Last hot reset is not second global reset 1'b1: Last hot reset is second 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 configure 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 configure value

CRU GLB RST CON

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	wdt2_glb_srst_ctrl 1'b0: WDT2 trigger second global reset 1'b1: WDT2 trigger first global reset
13	RW	0x0	wdt1_glb_srst_ctrl 1'b0: WDT1 trigger second global reset 1'b1: WDT1 trigger first global reset
12	RW	0x0	wdt0_glb_srst_ctrl 1'b0: WDT0 trigger second global reset 1'b1: WDT0 trigger first global reset
11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	wdt2_glb_srst_en 1'b0: Disable WDT1 reset as global reset source 1'b1: Enable WDT1 reset as global reset source
9	RW	0x0	wdt1_glb_srst_en 1'b0: Disable WDT1 reset as global reset source 1'b1: Enable WDT1 reset as global reset source
8	RW	0x0	wdt0_glb_srst_en 1'b0: Disable WDT0 reset as global reset source 1'b1: Enable WDT0 reset as global reset source
7	RW	0x0	wdt_reset_ext_en 1'b0: Disable WDT reset extend. 1'b1: Enable WDT reset extend, reset extend time depend on bit31~0 of GLB_CNT_TH
6	RW	0x0	tsadc_reset_ext_en 1'b0: Disable TSADC reset extend. 1'b1: Enable TSADC reset extend, reset extend time depend on bit31~0 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	tsadc_glb_srst_ctrl 1'b0: TSADC trigger second global reset 1'b1: TSADC trigger first global reset
0	RW	0x0	tsadc_glb_srst_en 1'b0: Disable TSADC trigger global reset 1'b1: Enable TSADC trigger global reset

CRU CLKSEL CON00

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:8	RW	0x0	clk_gpll_mux_np5_div clk_gpll_mux_np5 DivfreeNP5 division: Divide as 2*clk_gpll_mux_np5(2*div_con + 3)

Bit	Attr	Reset Value	Description
7	RW	0x0	clk_32k_frac_div_sel clk_32k_frac_div clock MUX: 1'b0: xin_osc0_func 1'b1: clk_RTC_pvtm
6:5	RW	0x2	clk_32k_sel clk_32k clock MUX: 2'b00: clk_32k_frac_div 2'b01: clk_RTC 2'b10: clk_RTC_npor
4:0	RO	0x0	reserved

CRU CLKSEL CON01

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_32k_frac_div_div clk_32k_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON02

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x01	pclk_mcu_bus_div Divide pclk_mcu_bus by (div_con + 1)
7:6	RW	0x0	hclk_mcu_bus_sel hclk_mcu_bus clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: clk_gpll_mux_np5
5	RO	0x0	reserved
4:0	RW	0x01	hclk_mcu_bus_div Divide hclk_mcu_bus by (div_con + 1)

CRU CLKSEL CON03

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_dsp_sel clk_dsp clock MUX: 1'b0: clk_dsp_div 1'b1: clk_dsp_np5_div
14:13	RO	0x0	reserved
12:8	RW	0x01	clk_dsp_np5_div_div clk_dsp_np5_div DivfreeNP5 division: Divide as $2 * \text{clk_dsp_np5_div} * (\text{div_con} + 3)$
7:6	RW	0x1	clk_dsp_src_sel clk_dsp_src clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vppll_mux 2'b10: xin_osc0_func
5	RO	0x0	reserved
4:0	RW	0x02	clk_dsp_div_div Divide clk_dsp_div by (div_con + 1)

CRU CLKSEL CON04

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_uart0_sel sclk_uart0 clock MUX: 1'b0: clk_uart0_div 1'b1: clk_uart0_frac_div
14:8	RO	0x0	reserved
7	RW	0x1	clk_uart0_div_sel clk_uart0_div clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:5	RO	0x0	reserved
4:0	RW	0x00	clk_uart0_div_div Divide clk_uart0_div by (div_con + 1)

CRU CLKSEL CON05

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart0_frac_div_div clk_uart0_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON06

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_uart1_sel sclk_uart1 clock MUX: 1'b0: clk_uart1_div 1'b1: clk_uart1_frac_div
14:8	RO	0x0	reserved
7	RW	0x1	clk_uart1_div_sel clk_uart1_div clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:5	RO	0x0	reserved
4:0	RW	0x00	clk_uart1_div_div Divide clk_uart1_div by (div_con + 1)

CRU CLKSEL CON07

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart1_frac_div_div clk_uart1_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON08

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_uart2_sel sclk_uart2 clock MUX: 1'b0: clk_uart2_div 1'b1: clk_uart2_frac_div

Bit	Attr	Reset Value	Description
14:8	RO	0x0	reserved
7	RW	0x1	clk_uart2_div_sel clk_uart2_div clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:5	RO	0x0	reserved
4:0	RW	0x00	clk_uart2_div_div Divide clk_uart2_div by (div_con + 1)

CRU CLKSEL CON09

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_uart2_frac_div_div clk_uart2_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON10

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	clk_i2c1_sel clk_i2c1 clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_gpll_mux_np5 2'b10: xin_osc0_func
13:8	RW	0x01	clk_i2c1_div Divide clk_i2c1 by (div_con + 1)
7:6	RW	0x0	clk_i2c0_sel clk_i2c0 clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_gpll_mux_np5 2'b10: xin_osc0_func
5:0	RW	0x01	clk_i2c0_div Divide clk_i2c0 by (div_con + 1)

CRU CLKSEL CON11

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	clk_i2c_codec_sel clk_i2c_codec clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_gpll_mux_np5 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
13:8	RW	0x01	clk_i2c_codec_div Divide clk_i2c_codec by (div_con + 1)
7:6	RW	0x0	clk_i2c2_sel clk_i2c2 clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_gpll_mux_np5 2'b10: xin_osc0_func
5:0	RW	0x01	clk_i2c2_div Divide clk_i2c2 by (div_con + 1)

CRU CLKSEL CON12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	clk_pwm1_sel clk_pwm1 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
14	RO	0x0	reserved
13:8	RW	0x00	clk_pwm1_div Divide clk_pwm1 by (div_con + 1)
7	RW	0x1	clk_pwm0_sel clk_pwm0 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6	RO	0x0	reserved
5:0	RW	0x00	clk_pwm0_div Divide clk_pwm0 by (div_con + 1)

CRU CLKSEL CON13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	clk_efuse_div Divide clk_efuse by (div_con + 1)
7	RW	0x1	clk_pwm2_sel clk_pwm2 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6	RO	0x0	reserved
5:0	RW	0x00	clk_pwm2_div Divide clk_pwm2 by (div_con + 1)

CRU CLKSEL CON14

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	clk_spi1_sel clk_spi1 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
14	RO	0x0	reserved
13:8	RW	0x00	clk_spi1_div Divide clk_spi1 by (div_con + 1)
7	RW	0x1	clk_spi0_sel clk_spi0 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6	RO	0x0	reserved
5:0	RW	0x00	clk_spi0_div Divide clk_spi0 by (div_con + 1)

CRU CLKSEL CON15

Address: Operational Base + offset (0x013c)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x1	clk_timer1_sel clk_timer1 clock MUX: 2'b00: clk_gpll_mux 2'b01: xin_osc0_func 2'b10: clk_32k
13:8	RW	0x00	clk_timer1_div Divide clk_timer1 by (div_con + 1)
7	RW	0x1	clk_timer0_sel clk_timer0 clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6	RO	0x0	reserved
5:0	RW	0x00	clk_timer0_div Divide clk_timer0 by (div_con + 1)

CRU CLKSEL CON16

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x3	clk_fspi1_sel clk_fspi1 clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: clk_usbpll_mux 2'b11: xin_osc0_func
13:8	RW	0x00	clk_fspi1_div Divide clk_fspi1 by (div_con + 1), it's Divfree50 divider
7:6	RW	0x3	clk_fspi0_sel clk_fspi0 clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: clk_usbpll_mux 2'b11: xin_osc0_func
5:0	RW	0x00	clk_fspi0_div Divide clk_fspi0 by (div_con + 1), it's Divfree50 divider

CRU CLKSEL CON17

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x0	reserved
7:6	RW	0x3	clk_hyperbus_sel clk_hyperbus clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: clk_usbpll_mux 2'b11: xin_osc0_func
5:0	RW	0x00	clk_hyperbus_div Divide clk_hyperbus by (div_con + 1), it's Divfree50 divider

CRU CLKSEL CON18

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	sclk_aud pwm_sel sclk_aud pwm clock MUX: 1'b0: clk_aud pwm_div 1'b1: clk_aud pwm_frac_div
14:8	RO	0x0	reserved
7:6	RW	0x0	clk_aud pwm_div_sel clk_aud pwm_div clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: clk_usbpll_mux 2'b11: xin_osc0_func
5	RO	0x0	reserved
4:0	RW	0x03	clk_aud pwm_div_div Divide clk_aud pwm_div by (div_con + 1)

CRU CLKSEL CON19

Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_aud pwm_frac_div_div clk_aud pwm_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON20

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12:8	RW	0x03	hclk_peri_bus_div Divide hclk_peri_bus by (div_con + 1)
7	RW	0x0	aclk_peri_bus_sel aclk_peri_bus clock MUX: 1'b0: clk_gpll_mux 1'b1: clk_gpll_mux_np5
6:5	RO	0x0	reserved
4:0	RW	0x01	aclk_peri_bus_div divide aclk_peri_bus by (div_con + 1)

CRU CLKSEL CON21

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_sdmmc_sel clk_sdmmc clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
14:8	RW	0x03	clk_sdmmc_div Divide clk_sdmmc by (div_con + 1), it's Divfree50 divider
7:6	RW	0x0	clk_vip_out_sel clk_vip_out clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vppll_mux 2'b10: xin_osc0_func
5:0	RW	0x0f	clk_vip_out_div Divide clk_vip_out by (div_con + 1), it's Divfree50 divider

CRU CLKSEL CON22

Address: Operational Base + offset (0x0158)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	clk_crypto_sel clk_crypto clock MUX: 1'b0: clk_crypto_div 1'b1: clk_crypto_np5_div
14	RW	0x0	clk_crypto_pka_sel clk_crypto_pka clock MUX: 1'b0: clk_crypto_div 1'b1: clk_crypto_np5_div
13	RO	0x0	reserved
12:8	RW	0x01	clk_crypto_np5_div_div clk_crypto_np5_div DivfreeNP5 division: Divide as $2 * \text{clk_crypto_np5_div} (2 * \text{div_con} + 3)$
7	RW	0x0	clk_crypto_src_sel clk_crypto_src clock MUX: 1'b0: clk_gpll_mux 1'b1: xin_osc0_func
6:5	RO	0x0	reserved
4:0	RW	0x02	clk_crypto_div_div Divide clk_crypto_div by (div_con + 1)

CRU CLKSEL CON23

Address: Operational Base + offset (0x015c)

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

CRU CLKSEL CON24

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	pclk_top_bus_sel pclk_top_bus clock MUX: 2'b00: clk_gpll_mux 2'b01: rf_ad_clk80_soc 2'b10: xin_osc0_func

Bit	Attr	Reset Value	Description
13:8	RW	0x03	pclk_top_bus_div Divide pclk_top_bus by (div_con + 1)
7:6	RW	0x0	hclk_top_bus_sel hclk_top_bus clock MUX: 2'b00: clk_gpll_mux 2'b01: rf_ad_clk80_soc 2'b10: xin_osc0_func
5:0	RW	0x03	hclk_top_bus_div Divide hclk_top_bus by (div_con + 1)

CRU CLKSEL CON25

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x0	reserved
7:6	RW	0x0	mclk_pdm_sel mclk_pdm clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
5:0	RW	0x01	mclk_pdm_div divide mclk_pdm by (div_con + 1)

CRU CLKSEL CON26

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x0	mclkout_i2s0_sel mclkout_i2s0 clock MUX: 2'b00: mclk_i2s0_tx_mux 2'b01: mclk_i2s0_rx_mux 2'b10: xin_osc0_half
13:12	RO	0x0	reserved
11:10	RW	0x0	mclk_i2s0_tx_mux_sel mclk_i2s0_tx_mux clock MUX: 2'b00: clk_i2s0_tx_div 2'b01: clk_i2s0_tx_frac_div 2'b10: i2s0_mclkin

Bit	Attr	Reset Value	Description
9:8	RW	0x0	clk_i2s0_tx_div_sel clk_i2s0_tx_div clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vppll_mux 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x07	clk_i2s0_tx_div_div Divide clk_i2s0_tx_div by (div_con + 1)

CRU CLKSEL CON27

Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s0_tx_frac_div_div clk_i2s0_tx_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON28

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x0	mclk_i2s0_rx_mux_sel mclk_i2s0_rx_mux clock MUX: 2'b00: clk_i2s0_rx_div 2'b01: clk_i2s0_rx_frac_div 2'b10: i2s0_mclkin
9:8	RW	0x0	clk_i2s0_rx_div_sel clk_i2s0_rx_div clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vppll_mux 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x07	clk_i2s0_rx_div_div Divide clk_i2s0_rx_div by (div_con + 1)

CRU CLKSEL CON29

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s0_rx_frac_div_div clk_i2s0_rx_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON30

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	clk_codec_sel clk_codec clock MUX: 1'b0: mclk_i2s1_tx_mux 1'b1: mclk_i2s1_rx_mux
14:12	RO	0x0	reserved
11:10	RW	0x0	mclk_i2s1_tx_mux_sel mclk_i2s1_tx_mux clock MUX: 2'b00: clk_i2s1_tx_div 2'b01: clk_i2s1_tx_frac_div 2'b10: i2s1_mclkin
9:8	RW	0x0	clk_i2s1_tx_div_sel clk_i2s1_tx_div clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vpll_mux 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x07	clk_i2s1_tx_div_div Divide clk_i2s1_tx_div by (div_con + 1)

CRU CLKSEL CON31

Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s1_tx_frac_div_div clk_i2s1_tx_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON32

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x0	mclk_i2s1_rx_mux_sel mclk_i2s1_rx_mux clock MUX: 2'b00: clk_i2s1_rx_div 2'b01: clk_i2s1_rx_frac_div 2'b10: i2s1_mclkin
9:8	RW	0x0	clk_i2s1_rx_div_sel clk_i2s1_rx_div clock MUX: 2'b00: clk_gpll_mux 2'b01: clk_vppll_mux 2'b10: rf_ad_clk80_soc 2'b11: xin_osc0_func
7	RO	0x0	reserved
6:0	RW	0x07	clk_i2s1_rx_div_div Divide clk_i2s1_rx_div by (div_con + 1)

CRU CLKSEL CON33

Address: Operational Base + offset (0x0184)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clk_i2s1_rx_frac_div_div clk_i2s1_rx_frac_div fraction division: High 16-bit for numerator Low 16-bit for denominator

CRU CLKSEL CON34

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x0	reserved
4:0	RW	0x0f	clk_otg_usbphy_div Divide clk_otg_usbphy by (div_con + 1), it's Divfree50 divider

CRU CLKSEL CON35

Address: Operational Base + offset (0x018c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x0	reserved
8:4	RW	0x00	outclock_test_sel 5'h00: xin_osc0 5'h01: clk_32k 5'h02: clk_gpll_mux 5'h03: clk_vpll_mux 5'h04: clk_usbpll_mux 5'h05: hclk_mcu_bus 5'h06: aclk_peri_bus 5'h07: hclk_peri_bus 5'h08: hclk_top_bus 5'h09: aclk_wlan_bus 5'h0a: clk_dsp 5'h0b: sclk_audpwm 5'h0c: clk_efuse 5'h0d: clk_i2c0 5'h0e: clk_i2c1 5'h0f: clk_pwm0 5'h10: clk_spi0 5'h11: clk_spi1 5'h12: clk_timer0 5'h13: clk_hyperbus 5'h14: clk_fspi0 5'h15: clk_fspi1 5'h16: sclk_uart0 5'h17: sclk_uart1 5'h18: clk_crypto 5'h19: clk_sdmmc 5'h1a: clk_vip_out 5'h1b: clk_codec 5'h1c: mclk_i2s1_rx 5'h1d: mclk_i2s1_tx 5'h1e: usb_phyclk 5'h1f: clk_otg_usbphy
3:0	RW	0xf	outclock_test_div divide by (div_con + 1)

CRU CLKSEL CON36

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:8	RW	0x03	stclk_m0_div divide stclk_m0 by (div_con + 1)
7:6	RO	0x0	reserved
5:0	RW	0x03	stclk_m4f0_div divide stclk_m4f0 by (div_con + 1)

CRU GATE CON00

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x0	reserved
4	RW	0x0	clk_gpll_mux_div_np5_en clk_gpll_mux_np5 clock gating control: When high, disable clock.
3	RO	0x0	reserved
2	RW	0x0	clk_32k_en clk_32k clock gating control: When high, disable clock.
1	RW	0x0	clk_32k_frac_div_en clk_32k_frac_div clock gating control: When high, disable clock.
0	RO	0x0	reserved

CRU GATE CON01

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hclk_fspi0_en hclk_fspi0 clock gating control: When high, disable clock.
14	RW	0x0	hclk_vop_en hclk_vop clock gating control: When high, disable clock.
13	RW	0x0	hclk_dsp_tcm_en hclk_dsp_tcm clock gating control: When high, disable clock.
12	RW	0x0	hclk_rom_en hclk_rom clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
11	RW	0x0	hclk_dmac_en hclk_dmac clock gating control: When high, disable clock.
10	RW	0x0	hclk_intmem1_mcu_en hclk_intmem1 clock gating control: When high, disable clock.
9	RW	0x0	hclk_intmem0_mcu_en hclk_intmem0 clock gating control: When high, disable clock.
8	RO	0x0	reserved
7	RW	0x0	sclk_m0_en sclk_m0 clock gating control: When high, disable clock.
6	RW	0x0	dclk_m0_en dclk_m0 clock gating control: When high, disable clock.
5	RO	0x0	reserved
4	RW	0x0	fclk_m0_en fclk_m0 clock gating control: When high, disable clock.
3	RO	0x0	reserved
2	RW	0x0	fclk_m4f0_en fclk_m4f0 clock gating control: When high, disable clock.
1	RW	0x0	hclk_mcu_bus_niu_en hclk_mcu_bus_niu clock gating control: When high, disable clock.
0	RW	0x0	hclk_mcu_bus_pll_en hclk_mcu_bus clock gating control: When high, disable clock.

CRU GATE CON02

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_i2c2_en pclk_i2c2 clock gating control: When high, disable clock..
14	RW	0x0	pclk_i2c1_en pclk_i2c1 clock gating control: When high, disable clock..
13	RW	0x0	pclk_i2c0_en pclk_i2c0 clock gating control: When high, disable clock..
12	RW	0x0	pclk_uart2_en pclk_uart2 clock gating control: When high, disable clock..
11	RW	0x0	pclk_uart1_en pclk_uart1 clock gating control: When high, disable clock.
10	RW	0x0	pclk_uart0_en pclk_uart0 clock gating control: When high, disable clock.
9	RW	0x0	hclk_ahb2apb_en hclk_ahb2apb clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
8	RW	0x0	pclk_mcu_bus_pll_en pclk_mcu_bus clock gating control: When high, disable clock.
7:5	RO	0x0	reserved
4	RW	0x0	hclk_aud pwm_en hclk_aud pwm clock gating control: When high, disable clock.
3	RW	0x0	ackl_hyperbus_en ackl_hyperbus clock gating control: When high, disable clock.
2	RW	0x0	hclk_fspi1_xip_en hclk_fspi1_xip clock gating control: When high, disable clock.
1	RW	0x0	hclk_fspi1_en hclk_fspi1 clock gating control: When high, disable clock.
0	RW	0x0	hclk_fspi0_xip_en hclk_fspi0_xip clock gating control: When high, disable clock.

CRU GATE CON03

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	pclk_int_ctrl_en pclk_int_ctrl clock gating control: When high, disable clock.
12	RW	0x0	pclk_mailbox1_en pclk_mailbox1 clock gating control: When high, disable clock.
11	RW	0x0	pclk_mailbox0_en pclk_mailbox0 clock gating control: When high, disable clock.
10	RW	0x0	pclk_wdt2_en pclk_wdt2 clock gating control: When high, disable clock.
9	RW	0x0	pclk_wdt1_en pclk_wdt1 clock gating control: When high, disable clock.
8	RW	0x0	pclk_wdt0_en pclk_wdt0 clock gating control: When high, disable clock.
7	RW	0x0	pclk_timer0_en pclk_timer clock gating control: When high, disable clock.
6	RW	0x0	pclk_efuse_en pclk_efuse clock gating control: When high, disable clock.
5	RW	0x0	pclk_saradc_control_en pclk_saradc_control clock gating control: When high, disable clock.
4	RW	0x0	pclk_spi1_en pclk_spi1 clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
3	RW	0x0	pclk_spi0_en pclk_spi0 clock gating control: When high, disable clock.
2	RW	0x0	pclk_pwm2_en pclk_pwm2 clock gating control: When high, disable clock.
1	RW	0x0	pclk_pwm1_en pclk_pwm1 clock gating control: When high, disable clock.
0	RW	0x0	pclk_pwm0_en pclk_pwm0 clock gating control: When high, disable clock.

CRU GATE CON04

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	sclk_uart2_en sclk_uart2 clock gating control: When high, disable clock.
11	RW	0x0	clk_uart2_frac_en clk_uart2_frac_div clock gating control: When high, disable clock.
10	RW	0x0	clk_uart2_pll_en clk_uart2_div clock gating control: When high, disable clock.
9	RW	0x0	sclk_uart1_en sclk_uart1 clock gating control: When high, disable clock.
8	RW	0x0	clk_uart1_frac_en clk_uart1_frac_div clock gating control: When high, disable clock.
7	RW	0x0	clk_uart1_pll_en clk_uart1_div clock gating control: When high, disable clock.
6	RW	0x0	sclk_uart0_en sclk_uart0 clock gating control: When high, disable clock.
5	RW	0x0	clk_uart0_frac_en clk_uart0_frac_div clock gating control: When high, disable clock.
4	RW	0x0	clk_uart0_pll_en clk_uart0_div clock gating control: When high, disable clock.
3	RW	0x0	aclk_dsp_niu_en aclk_dsp_niu clock gating control: When high, disable clock.
2	RW	0x0	clk_dsp_en clk_dsp clock gating control: When high, disable clock.
1	RW	0x0	clk_dsp_np5_div_en clk_dsp_np5_div clock gating control: When high, disable clock.
0	RW	0x0	clk_dsp_div_en clk_dsp_div clock gating control: When high, disable clock.

CRU GATE CON05

Address: Operational Base + offset (0x0314)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_timer0_ch5_en clk_timer0_ch5 clock gating control: When high, disable clock.
13	RW	0x0	clk_timer0_ch4_en clk_timer0_ch4 clock gating control: When high, disable clock.
12	RW	0x0	clk_timer0_ch3_en clk_timer0_ch3 clock gating control: When high, disable clock.
11	RW	0x0	clk_timer0_ch2_en clk_timer0_ch2 clock gating control: When high, disable clock.
10	RW	0x0	clk_timer0_ch1_en clk_timer0_ch1 clock gating control: When high, disable clock.
9	RW	0x0	clk_timer0_ch0_en clk_timer0_ch0 clock gating control: When high, disable clock.
8	RW	0x0	clk_timer0_pll_en clk_timer0 clock gating control: When high, disable clock.
7	RW	0x0	clk_spi1_pll_en clk_spi1 clock gating control: When high, disable clock.
6	RW	0x0	clk_spi0_pll_en clk_spi0 clock gating control: When high, disable clock.
5	RW	0x0	clk_pwm2_pll_en clk_pwm2 clock gating control: When high, disable clock.
4	RW	0x0	clk_pwm1_pll_en clk_pwm1 clock gating control: When high, disable clock.
3	RW	0x0	clk_pwm0_pll_en clk_pwm0 clock gating control: When high, disable clock.
2	RW	0x0	clk_i2c2_pll_en clk_i2c2 clock gating control: When high, disable clock.
1	RW	0x0	clk_i2c1_pll_en clk_i2c1 clock gating control: When high, disable clock.
0	RW	0x0	clk_i2c0_pll_en clk_i2c0 clock gating control: When high, disable clock.

CRU GATE CON06

Address: Operational Base + offset (0x0318)

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

Bit	Attr	Reset Value	Description
15:14	RO	0x0	reserved
13	RW	0x0	stclk_m0_en stclk_m0 clock gating control: When high, disable clock.
12	RW	0x0	clk_pwm_capture2_pll_en clk_pwm_capture2 clock gating control: When high, disable clock.
11	RW	0x0	clk_pwm_capture1_pll_en clk_pwm_capture1 clock gating control: When high, disable clock.
10	RW	0x0	clk_pwm_capture0_pll_en clk_pwm_capture0 clock gating control: When high, disable clock.
9	RW	0x0	sclk_aud pwm_en sclk_aud pwm clock gating control: When high, disable clock.
8	RW	0x0	clk_aud pwm_frac_en clk_aud pwm_frac_div clock gating control: When high, disable clock.
7	RW	0x0	clk_aud pwm_pll_en clk_aud pwm_div clock gating control: When high, disable clock.
6	RW	0x0	stclk_m4f0_en stclk_m4f0 clock gating control: When high, disable clock.
5	RW	0x0	clk_hyperbus_dt50_en clk_hyperbus clock gating control: When high, disable clock.
4	RW	0x0	clk_fspi1_dt50_en clk_fspi1 clock gating control: When high, disable clock.
3	RW	0x0	clk_fspi0_dt50_en clk_fspi0 clock gating control: When high, disable clock.
2	RW	0x0	clk_efuse_en clk_efuse clock gating control: When high, disable clock.
1:0	RO	0x0	reserved

CRU GATE CON07

Address: Operational Base + offset (0x031c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	clk_vip_pll_en clk_vip_out clock gating control: When high, disable clock.
13	RW	0x0	pclk_spi2apb_en pclk_spi2apb clock gating control: When high, disable clock.
12:11	RO	0x0	reserved
10	RW	0x0	hclk_usbotg_pmu_en hclk_usbotg_pmu clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
9	RW	0x0	hclk_usbotg_en hclk_usbotg clock gating control: When high, disable clock.
8	RW	0x0	hclk_sdmmc_en hclk_sdmmc clock gating control: When high, disable clock.
7	RW	0x0	hclk_crypto_en hclk_crypto clock gating control: When high, disable clock.
6	RW	0x0	hclk_vip_en hclk_vip clock gating control: When high, disable clock.
5	RW	0x0	ackl_crypto_en ackl_crypto clock gating control: When high, disable clock.
4	RW	0x0	ackl_vip_en ackl_vip clock gating control: When high, disable clock.
3	RW	0x0	hclk_peri_bus_niu_en hclk_peri_bus_niu clock gating control: When high, disable clock.
2	RW	0x0	ackl_peri_bus_niu_en ackl_peri_bus_niu clock gating control: When high, disable clock.
1	RW	0x0	hclk_peri_bus_pll_en hclk_peri_bus clock gating control: When high, disable clock.
0	RW	0x0	ackl_peri_bus_pll_en ackl_peri_bus clock gating control: When high, disable clock.

CRU GATE CON08

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	clk40_wlan_en clk40_wlan clock gating control: When high, disable clock.
12	RW	0x0	clk80_wlan_en clk80_wlan clock gating control: When high, disable clock.
11	RW	0x0	hclk_buffer_en hclk_buffer clock gating control: When high, disable clock.
10	RO	0x0	reserved
9	RW	0x0	ackl_wlan_bus_niu_en ackl_wlan_bus_niu clock gating control: When high, disable clock.
8	RW	0x0	ackl_wlan_bus_en ackl_wlan_bus clock gating control: When high, disable clock.
7	RO	0x0	reserved
6	RW	0x0	clk_crypto_pka_en clk_crypto_pka clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
5	RW	0x0	clk_crypto_en clk_crypto clock gating control: When high, disable clock.
4	RW	0x0	clk_crypto_np5_div_en clk_crypto_np5_div clock gating control: When high, disable clock.
3	RW	0x0	clk_crypto_div_en clk_crypto_div clock gating control: When high, disable clock.
2	RW	0x0	clk_sdmmc_dt50_en clk_sdmmc clock gating control: When high, disable clock.
1	RW	0x0	clk_otg_adp_en clk_otg_adp clock gating control: When high, disable clock.
0	RO	0x0	reserved

CRU GATE CON09

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	pclk_timer1_en pclk_timer1 clock gating control: When high, disable clock.
14	RW	0x0	pclk_32ktrim_en pclk_32ktrim clock gating control: When high, disable clock.
13	RW	0x0	pclk_acodec_en pclk_acodec clock gating control: When high, disable clock.
12	RW	0x0	pclk_grf_en pclk_grf clock gating control: When high, disable clock.
11	RW	0x0	pclk_cru_en pclk_cru clock gating control: When high, disable clock.
10	RW	0x0	pclk_pmu_en pclk_pmu clock gating control: When high, disable clock.
9	RW	0x0	pclk_gpio1_en pclk_gpio1 clock gating control: When high, disable clock.
8	RW	0x0	pclk_gpio0_en pclk_gpio0 clock gating control: When high, disable clock.
7	RW	0x0	pclk_top_bus_niu_en pclk_top_bus_niu clock gating control: When high, disable clock.
6	RW	0x0	pclk_top_bus_pll_en pclk_top_bus clock gating control: When high, disable clock.
5	RW	0x0	hclk_vad_en hclk_vad clock gating control: When high, disable clock.
4	RW	0x0	hclk_i2s1_en hclk_i2s1 clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
3	RW	0x0	hclk_i2s0_en hclk_i2s0 clock gating control: When high, disable clock.
2	RW	0x0	hclk_pdm_en hclk_pdm clock gating control: When high, disable clock.
1	RW	0x0	hclk_top_bus_niu_en hclk_top_bus_niu clock gating control: When high, disable clock.
0	RW	0x0	hclk_top_bus_pll_en hclk_top_bus clock gating control: When high, disable clock.

CRU GATE CON10

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mclk_i2s0_tx_en mclk_i2s0_tx clock gating control: When high, disable clock.
14	RW	0x0	clk_i2s0_tx_frac_div_en clk_i2s0_tx_frac_div clock gating control: When high, disable clock.
13	RW	0x0	clk_i2s0_tx_pll_en clk_i2s0_tx_div clock gating control: When high, disable clock.
12	RW	0x0	mclk_pdm_pll_en mclk_pdm clock gating control: When high, disable clock.
11	RO	0x0	reserved
10	RW	0x0	clk_pmu_osc_en clk_pmu_osc clock gating control: When high, disable clock.
9	RW	0x0	dbclk_gpio1_en dbclk_gpio1 clock gating control: When high, disable clock.
8	RW	0x0	dbclk_gpio0_en dbclk_gpio0 clock gating control: When high, disable clock.
7	RW	0x0	clk_pmu_en clk_pmu clock gating control: When high, disable clock.
6	RW	0x0	clk_32k_top_en clk_32k_top clock gating control: When high, disable clock.
5	RW	0x0	aon_sleep_clk_en aon_sleep_clk clock gating control: When high, disable clock.
4:3	RO	0x0	reserved
2	RW	0x0	pclk_tsadc_en pclk_tsadc clock gating control: When high, disable clock.
1	RW	0x0	pclk_touch_key_controller_en pclk_touch_key_controller clock gating control: When high, disable clock.

Bit	Attr	Reset Value	Description
0	RW	0x0	pclk_pvtm_en pclk_pvtm clock gating control: When high, disable clock.

CRU GATE CON11

Address: Operational Base + offset (0x032c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_i2c_codec_pll_en clk_i2c_codec clock gating control: When high, disable clock.
10	RW	0x0	sclk_codec_en clk_codec clock gating control: When high, disable clock.
9	RW	0x0	mclk_i2s1_rx_en mclk_i2s1_rx clock gating control: When high, disable clock.
8	RW	0x0	clk_i2s1_rx_frac_div_en clk_i2s1_rx_frac_div clock gating control: When high, disable clock.
7	RW	0x0	clk_i2s1_rx_pll_en clk_i2s1_rx_div clock gating control: When high, disable clock.
6	RW	0x0	mclk_i2s1_tx_en mclk_i2s1_tx clock gating control: When high, disable clock.
5	RW	0x0	clk_i2s1_tx_frac_div_en clk_i2s1_tx_frac_div clock gating control: When high, disable clock.
4	RW	0x0	clk_i2s1_tx_pll_en clk_i2s1_tx_div clock gating control: When high, disable clock.
3	RW	0x0	mclkout_i2s0_en mclkout_i2s0 clock gating control: When high, disable clock.
2	RW	0x0	mclk_i2s0_rx_en mclk_i2s0_rx clock gating control: When high, disable clock.
1	RW	0x0	clk_i2s0_rx_frac_div_en clk_i2s0_rx_frac_div clock gating control: When high, disable clock.
0	RW	0x0	clk_i2s0_rx_pll_en clk_i2s0_rx_div clock gating control: When high, disable clock.

CRU GATE CON12

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	clk_tsadc_en clk_tsadc clock gating control: When high, disable clock.
10	RW	0x0	clk_saradc_en clk_saradc clock gating control: When high, disable clock.
9	RW	0x0	clk_timer1_pll_en clk_timer1 clock gating control: When high, disable clock.
8	RW	0x0	clk_touch_key_controller_en clk_touch_key_controller clock gating control: When high, disable clock.
7	RW	0x0	clk_pvtm_en clk_pvtm clock gating control: When high, disable clock.
6	RO	0x0	reserved
5	RW	0x0	outclock_test_en outclock_test clock gating control: When high, disable clock.
4	RW	0x0	clk_32ktrim_en clk_32ktrim clock gating control: When high, disable clock.
3	RW	0x0	clk_otg_usbphy_pll_en clk_otg_usbphy clock gating control: When high, disable clock.
2:0	RO	0x0	reserved

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 CON00

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x0	reserved
8	R/W SC	0x0	preseth_cru_ac When high, reset relative logic, the reset will be automatically self-cleared
7	R/W SC	0x0	preseth_top_bus_ac When high, reset relative logic, the reset will be automatically self-cleared.
6	RW	0x0	dbresetn_m0_ac When high, reset relative logic, the reset will be automatically self-cleared.
5	RW	0x0	hresetn_m0_ac When high, reset relative logic, the reset will be automatically self-cleared.
4	R/W SC	0x0	poreseth_m0_ac When high, reset relative logic, the reset will be automatically self-cleared.
3	RW	0x0	hresetn_m4f0_ac When high, reset relative logic, the reset will be automatically self-cleared.
2	R/W SC	0x0	poreseth_m4f0_ac When high, reset relative logic, the reset will be automatically self-cleared.
1	R/W SC	0x0	hresetn_mcusbus_ac When high, reset relative logic, the reset will be automatically self-cleared.
0	RO	0x0	reserved

CRU SOFTRST CON01

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	hresetn_fspi0 When high, reset relative logic.
14	RW	0x0	hresetn_vop When high, reset relative logic.
13	RW	0x0	hresetn_dsp_tcm When high, reset relative logic.

Bit	Attr	Reset Value	Description
12	RW	0x0	hresetn_rom When high, reset relative logic.
11	RW	0x0	hresetn_dmac When high, reset relative logic.
10	RW	0x0	hresetn_intmem1 When high, reset relative logic.
9	RW	0x0	hresetn_intmem0 When high, reset relative logic.
8:7	RO	0x0	reserved
6	RW	0x1	dbresetn_m0 When high, reset relative logic.
5	RW	0x1	hresetn_m0 When high, reset relative logic.
4	RW	0x1	poresetn_m0 When high, reset relative logic.
3	RW	0x0	hresetn_m4f0 When high, reset relative logic.
2	RW	0x0	poresetn_m4f0 When high, reset relative logic.
1	RW	0x0	hresetn_mcu_bus_niu When high, reset relative logic.
0	RO	0x0	reserved

CRU SOFTRST CON02

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_i2c2 When high, reset relative logic.
14	RW	0x0	presetn_i2c1 When high, reset relative logic.
13	RW	0x0	presetn_i2c0 When high, reset relative logic.
12	RW	0x0	presetn_uart2 When high, reset relative logic.
11	RW	0x0	presetn_uart1 When high, reset relative logic.
10	RW	0x0	presetn_uart0 When high, reset relative logic.
9	RW	0x0	hresetn_ahb2apb When high, reset relative logic.

Bit	Attr	Reset Value	Description
8:5	RO	0x0	reserved
4	RW	0x0	hresetn_aud pwm When high, reset relative logic.
3	RW	0x0	aresetn_hyperbus When high, reset relative logic.
2	RW	0x0	hresetn_fspi1_xip When high, reset relative logic.
1	RW	0x0	hresetn_fspi1 When high, reset relative logic.
0	RW	0x0	hresetn_fspi0_xip When high, reset relative logic.

CRU SOFTRST CON03

Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	presetn_int_ctrl When high, reset relative logic.
12	RW	0x0	presetn_mailbox1 When high, reset relative logic.
11	RW	0x0	presetn_mailbox0 When high, reset relative logic.
10	RW	0x0	presetn_wdt2 When high, reset relative logic.
9	RW	0x0	presetn_wdt1 When high, reset relative logic.
8	RW	0x0	presetn_wdt0 When high, reset relative logic.
7	RW	0x0	presetn_timer0 When high, reset relative logic.
6	RW	0x0	presetn_efuse When high, reset relative logic.
5	RW	0x0	presetn_saradc_control When high, reset relative logic.
4	RW	0x0	presetn_spi1 When high, reset relative logic.
3	RW	0x0	presetn_spi0 When high, reset relative logic.
2	RW	0x0	presetn_pwm2 When high, reset relative logic.

Bit	Attr	Reset Value	Description
1	RW	0x0	presetn_pwm1 When high, reset relative logic.
0	RW	0x0	presetn_pwm0 When high, reset relative logic.

CRU SOFTRST CON04

Address: Operational Base + offset (0x0410)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	sresetn_uart2 When high, reset relative logic.
11:10	RO	0x0	reserved
9	RW	0x0	sresetn_uart1 When high, reset relative logic.
8:7	RO	0x0	reserved
6	RW	0x0	sresetn_uart0 When high, reset relative logic.
5:4	RO	0x0	reserved
3	RW	0x0	aresetn_dsp_niu When high, reset relative logic.
2	RW	0x1	bresetn_dsp When high, reset relative logic.
1	RW	0x1	dresetn_dsp When high, reset relative logic.
0	RO	0x0	reserved

CRU SOFTRST CON05

Address: Operational Base + offset (0x0414)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	m4f0_jrst When high, reset relative logic.
14	RW	0x0	resetn_timer0_ch5 When high, reset relative logic.
13	RW	0x0	resetn_timer0_ch4 When high, reset relative logic.

Bit	Attr	Reset Value	Description
12	RW	0x0	resetn_timer0_ch3 When high, reset relative logic.
11	RW	0x0	resetn_timer0_ch2 When high, reset relative logic.
10	RW	0x0	resetn_timer0_ch1 When high, reset relative logic.
9	RW	0x0	resetn_timer0_ch0 When high, reset relative logic.
8	RO	0x0	reserved
7	RW	0x0	resetn_spi1 When high, reset relative logic.
6	RW	0x0	resetn_spi0 When high, reset relative logic.
5	RW	0x0	resetn_pwm2 When high, reset relative logic.
4	RW	0x0	resetn_pwm1 When high, reset relative logic.
3	RW	0x0	resetn_pwm0 When high, reset relative logic.
2	RW	0x0	resetn_i2c2 When high, reset relative logic.
1	RW	0x0	resetn_i2c1 When high, reset relative logic.
0	RW	0x0	resetn_i2c0 When high, reset relative logic.

CRU SOFTRST CON06

Address: Operational Base + offset (0x0418)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x0	reserved
9	RW	0x0	sresetn_audpwm When high, reset relative logic.
8:5	RO	0x0	reserved
4	RW	0x0	resetn_fspi1 When high, reset relative logic.
3	RW	0x0	resetn_fspi0 When high, reset relative logic.
2	RW	0x0	resetn_efuse When high, reset relative logic.

Bit	Attr	Reset Value	Description
1	RW	0x0	dsp_jtrst When high, reset relative logic.
0	RW	0x0	m0_jtrst When high, reset relative logic.

CRU SOFTRST CON07

Address: Operational Base + offset (0x041c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	resetn_vip When high, reset relative logic.
14	RO	0x0	reserved
13	RW	0x0	presetn_spi2apb When high, reset relative logic.
12:11	RO	0x0	reserved
10	RW	0x0	hresetn_usbotg_pmu When high, reset relative logic.
9	RW	0x0	hresetn_usbotg When high, reset relative logic.
8	RW	0x0	hresetn_sdmmc When high, reset relative logic.
7	RW	0x0	hresetn_crypto When high, reset relative logic.
6	RW	0x0	hresetn_vip When high, reset relative logic.
5	RW	0x0	aresetn_crypto When high, reset relative logic.
4	RW	0x0	aresetn_vip When high, reset relative logic.
3	RW	0x0	hresetn_peri_bus_niu When high, reset relative logic.
2	RW	0x0	aresetn_peri_bus_niu When high, reset relative logic.
1:0	RO	0x0	reserved

CRU SOFTRST CON08

Address: Operational Base + offset (0x0420)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	hresetn_buffer When high, reset relative logic.
10	RO	0x0	reserved
9	RW	0x0	aresetn_wlan_bus_niu When high, reset relative logic.
8:7	RO	0x0	reserved
6	RW	0x0	resetn_crypto_pka When high, reset relative logic.
5	RW	0x0	resetn_crypto When high, reset relative logic.
4:2	RO	0x0	reserved
1	RW	0x0	resetn_otg_adp When high, reset relative logic.
0	RW	0x0	resetn_utmi When high, reset relative logic.

CRU SOFTRST CON09

Address: Operational Base + offset (0x0424)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	presetn_timer1 When high, reset relative logic.
14	RW	0x0	presetn_32ktrim When high, reset relative logic.
13	RW	0x0	presetn_acodec When high, reset relative logic.
12	RW	0x0	presetn_grf When high, reset relative logic.
11	RW	0x0	presetn_cru When high, reset relative logic.
10	RO	0x0	reserved
9	RW	0x0	presetn_gpio1 When high, reset relative logic.
8	RW	0x0	presetn_gpio0 When high, reset relative logic.

Bit	Attr	Reset Value	Description
7	RW	0x0	presetn_top_bus_niu When high, reset relative logic.
6	RO	0x0	reserved
5	RW	0x0	hresetn_vad When high, reset relative logic.
4	RW	0x0	hresetn_i2s1 When high, reset relative logic.
3	RW	0x0	hresetn_i2s0 When high, reset relative logic.
2	RW	0x0	hresetn_pdm When high, reset relative logic.
1	RW	0x0	hresetn_top_bus_niu When high, reset relative logic.
0	RO	0x0	reserved

CRU SOFTRST CON10

Address: Operational Base + offset (0x0428)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	mresetn_i2s0_tx When high, reset relative logic.
14:13	RO	0x0	reserved
12	RW	0x0	mresetn_pdm When high, reset relative logic.
11:10	RO	0x0	reserved
9	RW	0x0	dbresetn_gpio1 When high, reset relative logic.
8	RW	0x0	dbresetn_gpio0 When high, reset relative logic.
7:6	RO	0x0	reserved
5	RW	0x0	resetn_aon When high, reset relative logic.
4:3	RO	0x0	reserved
2	RW	0x0	presetn_tsadc When high, reset relative logic.
1	RW	0x0	presetn_touch_key_controller When high, reset relative logic.
0	RW	0x0	presetn_pvtm When high, reset relative logic.

CRU SOFTRST CON11

Address: Operational Base + offset (0x042c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x0	reserved
10	RW	0x0	resetn_codec When high, reset relative logic.
9	RW	0x0	mresetn_i2s1_rx When high, reset relative logic.
8:7	RO	0x0	reserved
6	RW	0x0	mresetn_i2s1_tx When high, reset relative logic.
5:3	RO	0x0	reserved
2	RW	0x0	mresetn_i2s0_rx When high, reset relative logic.
1:0	RO	0x0	reserved

CRU SOFTRST CON12

Address: Operational Base + offset (0x0430)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11	RW	0x0	resetn_tsadc When high, reset relative logic.
10	RO	0x0	reserved
9	RW	0x0	resetn_timer1_ch0 When high, reset relative logic.
8	RW	0x0	resetn_touch_detect When high, reset relative logic.
7	RW	0x0	resetn_pvtm When high, reset relative logic.
6:5	RO	0x0	reserved
4	RW	0x0	resetn_32ktrim When high, reset relative logic.
3	RW	0x0	resetn_otg_usbphy When high, reset relative logic.
2	RO	0x0	reserved
1	RW	0x0	aon_jtrst When high, reset relative logic.
0	RO	0x0	reserved

CRU SDMMC CON00

Address: Operational Base + offset (0x0480)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	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 CON01

Address: Operational Base + offset (0x0484)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	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_CON00.

DSMPD, REFIDIV, POSTDIV2 can be configured by programming CRU_xPLL_CON01.

FRAC can be configured by programming CRU_xPLL_CON02. If DSMPD = 1, PLL is in integer mode. If DSMPD = 0, PLL is in fractional mode.

A. PLL integer mode configuration

$$\text{FOUTVCO} = (\text{FREF} / \text{REFDIV}) * \text{FBDIV}$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / (\text{POSTDIV1} * \text{POSTDIV2})$$

When FREF is 24MHz, and if 700MHz FOUTPOSTDIV is needed. The configuration can be:

$$\text{DSMPD} = 1$$

$$\text{REFDIV} = 6$$

$$\text{FBDIV} = 175$$

$$\text{POSTDIV1} = 1$$

$$\text{POSTDIV2} = 1$$

And then:

$$\text{FOUTVCO} = (\text{FREF} / \text{REFDIV}) * \text{FBDIV} = 24/6*175=700$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / (\text{POSTDIV1} * \text{POSTDIV2}) = 700/1/1=700$$

B. PLL fractional mode configuration

$$\text{FOUTVCO} = (\text{FREF} / \text{REFDIV}) * (\text{FBDIV} + \text{FRAC} / (2^{24}))$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / (\text{POSTDIV1} * \text{POSTDIV2})$$

When FREF is 24MHz, and if 491.52MHz FOUTPOSTDIV is needed. The configuration can be:

$$\text{DSMPD} = 0$$

$$\text{REFDIV} = 1$$

$$\text{FBDIV} = 40$$

$$\text{FRAC} = 24'hf5c28f$$

$$\text{POSTDIV1}=2$$

$$\text{POSTDIV2}=1$$

And then:

$$\text{FOUTVCO} = (\text{FREF} / \text{REFDIV}) * (\text{FBDIV} + \text{FRAC} / (2^{24})) = 983.04$$

$$\text{FOUTPOSTDIV} = \text{FOUTVCO} / (\text{POSTDIV1} * \text{POSTDIV2}) = 983.04/(2*1)=491.52$$

C. PLL setting consideration

- VCO output clock from 375MHz to 2.4GHz.
- 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_CON01[10] register. The max lock time for PLL is 1500 period of REF_CLK/RFFDIV.

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

- Change PLL from normal to slow mode by programming CRU_MODE_CON00.
- Configure CRU_xPLL_CON01[13] to power down PLL.
- Change PLL setting.
- Release PLL power down at least 1us after valid settings.
- Wait until PLL is lock state by checking CRU_xPLL_CON01[10] register or after delay about 1500 period of REF_CLK/RFFDIV.
- Change PLL into normal mode by programming CRU_MODE_CON00.

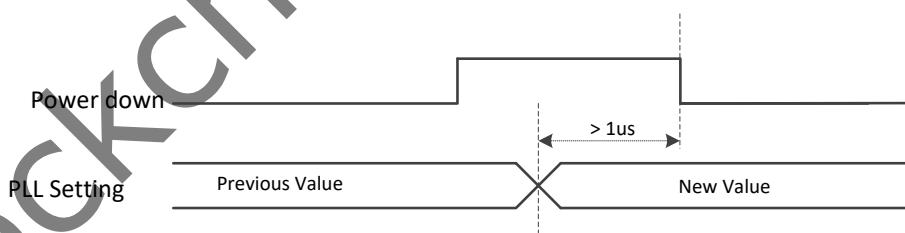


Fig. 2-5 PLL setting change timing

2.5.3 Fractional divider usage

In order to get specific frequency, clocks of AUDPWM, I2S, UART and Internal 32k RTC clock can be generated by fractional divider. User must set that denominator is 20 times larger than numerator to generate precise clock frequency when the requirement of clock accuracy and jitter is high. For implementation issue, the input source clocks of some fractional divider also have following limitation.

Table 2-1 Source Clock Limitation of Fractional Divider

Clock Name	Fractional divider source clock limitation
clk_uart0/1/2	400Mhz
clk_i2s0/1_tx	400Mhz
clk_i2s0/1_rx	400Mhz

2.5.4 Divfree50 divider usage

Some IPs, such as HyperBus, SDMMC, FSPI, USB OTG and VICAP need clock of 50% duty cycle, then Divfree50 divider is used.

2.5.5 DivfreeNP5 divider usage

Some IPs, such as DSP and Crypto need some special frequency, then DivfreeNP5 divider is used.

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 0xecaa to assert the second global software reset glb_srstn_2. These two software resets are self-cleared by hardware. Resetting pulse length 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 resets are controlled by CRU registers. For Cortex-M4F and Cortex-M0, when they assert system reset instruction, they also can be reset, and the reset will be cleared automatic.

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_CON03[12:8]) and ssmod_downspread(CRU_XPLL_CON03[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:

$$\text{Modulation Amplitude} = \pm 5'd(\text{ssmod_spread}) * 0.1\%$$

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_CON04[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_CON03[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_CON03[0])=1'b0
- Setting ssmod_disable_sscg(CRU_XPLL_CON03[1])=1'b0
- Setting ssmod_reset(CRU_XPLL_CON03[2])=1'b0

B. SSCG use External Point Table

In addition to the internal shape table, an external shape table can be used.

This enables customization tables in both shape and the number of sample points for 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_CON03[12:8]) and ssmod_downspread(CRU_XPLL_CON03[3])

Same with internal point table usage.

- Setting ssmod_sel_ext_wave (CRU_XPLL_CON04[0])=1'b1
- Setting ssmod_ext_maxaddr(CRU_XPLL_CON04[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_CON03[7:4])

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

- Setting ssmod_bp(CRU_XPLL_CON03[0])=1'b0
- Setting ssmod_disable_sscg(CRU_XPLL_CON03[1])=1'b0
- Setting ssmod_reset(CRU_XPLL_CON03[2])=1'b0

Chapter 3 General Register Files (GRF)

3.1 Overview

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

3.2 Block Diagram

N/A

3.3 Function Description

The function of general register file are:

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

3.4 Register Description

3.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0A_IOMUX_L	0x0000	W	0x00000066	GPIO0A IOMUX function select low bits control register
GRF_GPIO0A_IOMUX_H	0x0004	W	0x00000000	GPIO0A IOMUX function select high bits control register
GRF_GPIO0B_IOMUX_L	0x0008	W	0x00000555	GPIO0B IOMUX function select low bits control register
GRF_GPIO0B_IOMUX_H	0x000c	W	0x00000000	GPIO0B IOMUX function select high bits control register
GRF_GPIO0C_IOMUX_L	0x0010	W	0x00000000	GPIO0C IOMUX function select low bits control register
GRF_GPIO0C_IOMUX_H	0x0014	W	0x00000000	GPIO0C IOMUX function select high bits control register
GRF_GPIO0D_IOMUX_L	0x0018	W	0x00000000	GPIO0D IOMUX function select low bits control register
GRF_GPIO0D_IOMUX_H	0x001c	W	0x00000000	GPIO0D IOMUX function select high bits control register
GRF_GPIO1A_IOMUX_L	0x0020	W	0x00000000	GPIO1A IOMUX function select low bits control register
GRF_GPIO1A_IOMUX_H	0x0024	W	0x00000000	GPIO1A IOMUX function select high bits control register
GRF_GPIO1B_IOMUX_L	0x0028	W	0x00000000	GPIO1B IOMUX function select low bits control register
GRF_GPIO1B_IOMUX_H	0x002c	W	0x00000000	GPIO1B IOMUX function select high bits control register

Name	Offset	Size	Reset Value	Description
GRF GPIO1C IOMUX_L	0x0030	W	0x00000000	GPIO1C IOMUX function select low bits control register
GRF GPIO1D IOMUX_L	0x0038	W	0x00000000	GPIO1D IOMUX function select low bits control register
GRF GPIO0A_P	0x0100	W	0x0000a955	GPIO0A driver disabled state control register
GRF GPIO0B_P	0x0104	W	0x0000aa55	GPIO0B driver disabled state control register
GRF GPIO0C_P	0x0108	W	0x00009595	GPIO0C driver disabled state control register
GRF GPIO0D_P	0x010c	W	0x00001555	GPIO0D driver disabled state control register
GRF GPIO1A_P	0x0110	W	0x00000555	GPIO1A driver disabled state control register
GRF GPIO1B_P	0x0114	W	0x00005595	GPIO1B driver disabled state control register
GRF GPIO1C_P	0x0118	W	0x00000055	GPIO1C driver disabled state control register
GRF GPIO1D_P	0x011c	W	0x00000001	GPIO1D driver disabled state control register
GRF SOC CON0	0x0200	W	0x0000000e	SoC control register 0
GRF SOC CON1	0x0204	W	0x00000000	SoC control register 1
GRF SOC CON2	0x0208	W	0x00000ec0	SoC control register 2
GRF SOC CON3	0x020c	W	0x00000000	SoC control register 3
GRF SOC CON4	0x0210	W	0x00000000	SoC control register 4
GRF SOC CON5	0x0214	W	0x00000c00	SoC control register 5
GRF SOC CON6	0x0218	W	0x0000000c	SoC control register 6
GRF SOC CON7	0x021c	W	0x00000000	SoC control register 7
GRF SOC CON8	0x0220	W	0x00000000	SoC control register 8
GRF SOC CON9	0x0224	W	0x00000000	SoC control register 9
GRF SOC CON10	0x0228	W	0x0000318c	SoC control register 10
GRF SOC CON11	0x022c	W	0x00000000	SoC control register 11
GRF SOC CON12	0x0230	W	0x00000000	SoC control register 12
GRF SOC CON13	0x0234	W	0x00000000	SoC control register 13
GRF SOC CON14	0x0238	W	0x00000000	SoC control register 14
GRF SOC CON15	0x023c	W	0x000000ff	SoC control register 15
GRF SOC CON16	0x0240	W	0x00000000	SoC control register 16
GRF SOC CON17	0x0244	W	0x00000000	SoC control register 17
GRF SOC CON18	0x0248	W	0x00000000	SoC control register 18
GRF SOC CON19	0x024c	W	0x00000000	SoC control register 18
GRF SOC CON20	0x0250	W	0x00000000	SoC control register 18
GRF SOC CON21	0x0254	W	0x00000000	SoC control register 21
GRF SOC CON22	0x0258	W	0x00000000	SoC control register 22

Name	Offset	Size	Reset Value	Description
GRF_SOC_CON23	0x025c	W	0x00000000	SoC control register 23
GRF_SOC_CON24	0x0260	W	0x00000000	SoC control register 24
GRF_SOC_CON25	0x0264	W	0x00000000	SoC control register 25
GRF_SOC_CON26	0x0268	W	0x2003f000	SoC control register 26
GRF_SOC_CON27	0x026c	W	0x20040000	SoC control register 27
GRF_SOC_CON28	0x0270	W	0x00000000	SoC control register 28
GRF_SOC_CON29	0x0274	W	0x00000010	SoC control register 29
GRF_SOC_CON30	0x0278	W	0x00000004	SoC control register 30
GRF_SOC_CON31	0x027c	W	0x00000000	SoC control register 31
GRF_SOC_STATUS	0x0280	W	0x00000000	SOC status register
GRF MCU0_CON0	0x0300	W	0x00000000	MCU0 control register 0
GRF MCU0_CON1	0x0304	W	0x00000300	MCU0 control register 1
GRF MCU1_CON0	0x0308	W	0x00000000	MCU1 control register 0
GRF MCU1_CON1	0x030c	W	0x00000000	MCU1 control register 1
GRF_DSP_CON0	0x0320	W	0x00000000	DSP control register 0
GRF_DSP_CON1	0x0324	W	0x00000000	DSP control register 1
GRF_DSP_CON2	0x0328	W	0x00000000	DSP control register 2
GRF_SOC_UOC0	0x0340	W	0x0000c963	SOC USB control register 0
GRF_SOC_UOC1	0x0344	W	0x000016fb	SOC USB control register 1
GRF_SOC_UOC2	0x0348	W	0x00004408	SOC USB control register 2
GRF MCU0_STATUS	0x0380	W	0x00000000	MCU0 status register
GRF MCU1_STATUS	0x0384	W	0x00000000	MCU1 status register
GRF_DSP_STAT0	0x0388	W	0x00000000	SOC DSP status register 0
GRF_DSP_STAT1	0x038c	W	0x00000000	SOC DSP status register 1
GRF_GRF_FAST_BOOT	0x0400	W	0x00000001	SoC fast boot enable control register
GRF_GRF_FAST_BOOT_A DDR	0x0404	W	0x00000000	SoC fast boot address register
GRF_WLAN_CON	0x0500	W	0x00000007	WLAN control register
GRF_WLANCLK_CON	0x0504	W	0x00000000	WLAN Clock control register
GRF_WLAN_GPIO_IN	0x0508	W	0x00000000	WLAN GPIO in register
GRF_WLAN_GPIO_OUT	0x050c	W	0x00000000	WLAN GPIO out register
GRF_WLAN_STATUS	0x0580	W	0x00000000	LPW status register
GRF_USB2_DISCONNECT_CON	0x0680	W	0x00030d40	USB 2.0 PHY disconnect signal filter counter register
GRF_USB2_LINESTATE_C ON	0x0684	W	0x00030d40	USB 2.0 PHY linestate signal filter counter register
GRF_USB2_BVALID_CON	0x0688	W	0x00030d40	USB 2.0 PHY bvalid signal filter counter register
GRF_USB2_ID_CON	0x068c	W	0x00030d40	USB 2.0 PHY id signal filter counter register

Name	Offset	Size	Reset Value	Description
GRF_USB2_DETECT_IRQ_ENABLE	0x0690	W	0x00000001	USB 2.0 PHY connect detection interrupt request enable control register
GRF_USB2_DETECT_IRQ_STATUS	0x0694	W	0x00000001	USB 2.0 PHY connect detection interrupt status register
GRF_USB2_DETECT_IRQ_STATUS_CLR	0x0698	W	0x00000001	USB 2.0 PHY connect detection interrupt status clear register
GRF_SPINLOCK_REG_X	0x0700	W	0x00000000	Spinlock register x : Range of x is 0-63, offset is 0x700+4*n.
GRF_OS_REG0	0x0800	W	0x00000000	OS register 0
GRF_OS_REG1	0x0804	W	0x00000000	OS register 1
GRF_OS_REG2	0x0808	W	0x00000000	OS register 2
GRF_OS_REG3	0x080c	W	0x00000000	OS register 3
GRF_OS_REG4	0x0810	W	0x00000000	OS register 4
GRF_OS_REG5	0x0814	W	0x00000000	OS register 5
GRF_OS_REG6	0x0818	W	0x00000000	OS register 6
GRF_OS_REG7	0x081c	W	0x00000000	OS register 7
GRF_SOC_VERSION	0x0820	W	0x00002206	SoC version register

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

3.4.2 Detail Register Description

GRF_GPIO0A_IOMUX_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RW	0x0	gpio0a3_sel 4'b0000: GPIO0_A3_u 4'b0001: LCD_CSN 4'b0010: CIF_VSYNC 4'b0011: I2C1_SCL_M2 4'b0100: TKEY15 4'b0101: PMU_DEBUG 4'b0110: PMU_STATE1 4'b0111: AONJTAG_TDO 4'b1000: DSPJTAG_TDO 4'b1001: BT_MBSY Others: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x0	<p>gpio0a2_sel 4'b0000: GPIO0_A2_u 4'b0001: LCD_RS 4'b0010: CIF_HREF 4'b0011: I2C1_SDA_M2 4'b0100: TKEY14 4'b0101: TKEY_DRIVE_M3 4'b0110: PMU_STATE0 4'b0111: AONJTAG_TDI 4'b1000: DSPJTAG_TDI 4'b1001: BT_CONFIRM 4'b1010: BT_PTI_0 Others: Reserved</p>
7:4	RW	0x6	<p>gpio0a1_sel 4'b0000: GPIO0_A1_u 4'b0001: LCD_D1 4'b0010: CIF_D1 4'b0011: I2C0_SCL_M2 4'b0100: TKEY13 4'b0101: M0_WFI 4'b0110: M4F_JTAG_TMS 4'b0111: M0_JTAG_TMS 4'b1000: AONJTAG_TMS 4'b1001: DSPJTAG_TMS Others: Reserved</p>
3:0	RW	0x6	<p>gpio0a0_sel 4'b0000: GPIO0_A0_u 4'b0001: LCD_D0 4'b0010: CIF_D0 4'b0011: I2C0_SDA_M2 4'b0100: TKEY12 4'b0101: M4F_WFI 4'b0110: M4F_JTAG_TCK 4'b0111: M0_JTAG_TCK 4'b1000: AONJTAG_TCK 4'b1001: DSPJTAG_TCK Others: Reserved</p>

GRF GPIO0A_IOMUX_H

Address: Operational Base + offset (0x0004)

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

Bit	Attr	Reset Value	Description
15	RO	0x0	reserved
14:12	RW	0x0	gpio0a7_sel 3'b000: GPIO0_A7_d 3'b001: LCD_D3 3'b010: CIF_D3 3'b011: UART1_TX_M1 3'b100: PDM_CLK_S_M0 3'b101: TKEY19 3'b110: TEST_CLKOUT 3'b111: CODEC_DAC_DL_M1
11	RO	0x0	reserved
10:8	RW	0x0	gpio0a6_sel 3'b000: GPIO0_A6_d 3'b001: LCD_D2 3'b010: CIF_D2 3'b011: UART1_RX_M1 3'b100: PDM_SDI_M0 3'b101: TKEY18 3'b110: PMU_STATE4 3'b111: CODEC_ADC_D_M1
7:4	RW	0x0	gpio0a5_sel 4'b0000: GPIO0_A5_d 4'b0001: LCD_WRN 4'b0010: CIF_CLKIN 4'b0011: UART1_RTSN_M1 4'b0100: PDM_CLK_M0 4'b0101: TKEY17 4'b0110: PMU_STATE3 4'b0111: CODEC_SYNC_M1 4'b1000: BT_DENY
3:0	RW	0x0	gpio0a4_sel 4'b0000: GPIO0_A4_u 4'b0001: LCD_RDN 4'b0010: CIF_CLKOUT 4'b0011: UART1_CTSN_M1 4'b0100: TKEY16 4'b0101: PMU_SLEEP 4'b0110: PMU_STATE2 4'b0111: CODEC_CLK_M1 4'b1000: AONJTAG_TRSTn 4'b1001: DSPJTAG_TRSTn 4'b1010: BT_RXNTX Others: Reserved

GRF GPIO0B IOMUX L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0b3_sel 3'b000: GPIO0_B3_u 3'b001: LCD_D7 3'b010: CIF_D7 3'b011: UART2_TX_M1 3'b100: SPI1_MISO_M1 3'b101: SPI_SLV_MISO Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x5	gpio0b2_sel 3'b000: GPIO0_B2_u 3'b001: LCD_D6 3'b010: CIF_D6 3'b011: UART2_RX_M1 3'b100: SPI1_MOSI_M1 3'b101: SPI_SLV_MOSI Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x5	gpio0b1_sel 3'b000: GPIO0_B1_u 3'b001: LCD_D5 3'b010: CIF_D5 3'b011: UART2_RTSN_M1 3'b100: SPI1_CLK_M1 3'b101: SPI_SLV_CLK Others: Reserved
3	RO	0x0	reserved
2:0	RW	0x5	gpio0b0_sel 3'b000: GPIO0_B0_u 3'b001: LCD_D4 3'b010: CIF_D4 3'b011: UART2_CTSN_M1 3'b100: SPI1_CS0n_M1 3'b101: SPI_SLV_CSN Others: Reserved

GRF GPIO0B_IOMUX_H

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:12	RW	0x0	gpio0b7_sel 3'b000: GPIO0_B7_d 3'b001: PWM3_M1 3'b010: UART0_TX_M0 3'b011: SPI0_MISO_M0 3'b100: I2C1_SCL_M0 3'b101: TKEY3_M0 3'b110: TKEY_DRIVE_M2 Others: Reserved
11	RO	0x0	reserved
10:8	RW	0x0	gpio0b6_sel 3'b000: GPIO0_B6_d 3'b001: PWM2_M1 3'b010: UART0_RX_M0 3'b011: SPI0_MOSI_M0 3'b100: I2C1_SDA_M0 3'b101: TKEY2_M0 Others: Reserved
7	RO	0x0	reserved
6:4	RW	0x0	gpio0b5_sel 3'b000: GPIO0_B5_d 3'b001: PWM1_M1 3'b010: UART0_RTSN_M0 3'b011: SPI0_CLK_M0 3'b100: I2C0_SCL_M0 3'b101: TKEY1_M0 3'b110: PWM_AUDIO_R_M0 3'b111: I2C2_SCL_M1
3:0	RW	0x0	gpio0b4_sel 4'b0000: GPIO0_B4_d 4'b0001: PWM0_M1 4'b0010: UART0_CTSN_M0 4'b0011: SPI0_CS0n_M0 4'b0100: I2C0_SDA_M0 4'b0101: TKEY0_M0 4'b0110: TKEY_DRIVE_M0 4'b0111: PWM_AUDIO_L_M0 4'b1000: I2C2_SDA_M1 Others: Reserved

GRF GPIOOC IOMUX L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable</p>
15:12	RW	0x0	<p>gpio0c3_sel 4'b0000: GPIO0_C3_d 4'b0001: SRADC3 4'b0010: PWM3_M0 4'b0011: UART1_TX_M0 4'b0100: SPI0_MISO_M1 4'b0101: PDM_CLK_S_M1 4'b0110: I2S_MCLK_M0 4'b0111: TKEY4_M0 4'b1000: TKEY_DRIVE_M1 Others: Reserved Note: SOC_CON15[3] should be configured to 1'b1 when select SRADC3, otherwise it should be congured to 1'b0.</p>
11:8	RW	0x0	<p>gpio0c2_sel 4'b0000: GPIO0_C2_u 4'b0001: SRADC2 4'b0010: PWM2_M0 4'b0011: UART1_RX_M0 4'b0100: SPI0_MOSI_M1 4'b0101: I2C1_SCL_M1 4'b0110: I2S_LRCK_RX_M0 4'b0111: SDMMC_D0 4'b1000: TKEY11 Others: Reserved Note: SOC_CON15[2] should be configured to 1'b1 when select SRADC2, otherwise it should be congured to 1'b0.</p>

Bit	Attr	Reset Value	Description
7:4	RW	0x0	<p>gpio0c1_sel 4'b0000: GPIO0_C1_u 4'b0001: SRADC1 4'b0010: PWM1_M0 4'b0011: UART1_RTSN_M0 4'b0100: SPI0_CLK_M1 4'b0101: I2C1_SDA_M1 4'b0110: I2S_SCLK_RX_M0 4'b0111: SDMMC_CMD 4'b1000: TKEY10 Others: Reserved Note: SOC_CON15[1] should be configured to 1'b1 when select SRADC1, otherwise it should be configured to 1'b0.</p>
3	RO	0x0	reserved
2:0	RW	0x0	<p>gpio0c0_sel 3'b000: GPIO0_C0_u 3'b001: SRADC0 3'b010: PWM0_M0 3'b011: UART1_CTSN_M0 3'b100: SPI0_CS0n_M1 3'b101: I2S_SDO1_M0 3'b110: SDMMC_CLKOUT 3'b111: TKEY9 Note: SOC_CON15[0] should be configured to 1'b1 when select SRADC0, otherwise it should be configured to 1'b0.</p>

GRF GPIO0C_IOMUX_H

Address: Operational Base + offset (0x0014)

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

Bit	Attr	Reset Value	Description
15:12	RW	0x0	<p>gpio0c7_sel 4'b0000: GPIO0_C7_d 4'b0001: SRADC7 4'b0010: PWM7_M0 4'b0011: UART0_TX_M1 4'b0100: SPI1_MISO_M0 4'b0101: I2C0_SCL_M1 4'b0110: I2S_SDI_M0 4'b0111: TKEY8 4'b1000: PWM_AUDIO_R_M1 4'b1001: PMIC_INT_M1 Others: Reserved Note: SOC_CON15[7] should be configured to 1'b1 when select SRADC7, otherwise it should be configured to 1'b0.</p>
11:8	RW	0x0	<p>gpio0c6_sel 4'b0000: GPIO0_C6_u 4'b0001: SRADC6 4'b0010: PWM6_M0 4'b0011: UART0_RX_M1 4'b0100: SPI1_MOSI_M0 4'b0101: I2C0_SDA_M1 4'b0110: I2S_SDO_M0 4'b0111: TKEY7 4'b1000: PWM_AUDIO_L_M1 Others: Reserved Note: SOC_CON15[6] should be configured to 1'b1 when select SRADC6, otherwise it should be configured to 1'b0.</p>
7	RO	0x0	reserved
6:4	RW	0x0	<p>gpio0c5_sel 3'b000: GPIO0_C5_u 3'b001: SRADC5 3'b010: PWM5_M0 3'b011: UART0_RTSN_M1 3'b100: SPI1_CLK_M0 3'b101: PDM_SDI_M1 3'b110: I2S_LRCK_TX_M0 3'b111: TKEY6_M0 Note: SOC_CON15[5] should be configured to 1'b1 when select SRADC5, otherwise it should be configured to 1'b0.</p>
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	<p>gpio0c4_sel 3'b000: GPIO0_C4_u 3'b001: SRADC4 3'b010: PWM4_M0 3'b011: UART0_CTSN_M1 3'b100: SPI1_CS0n_M0 3'b101: PDM_CLK_M1 3'b110: I2S_SCLK_TX_M0 3'b111: TKEY5_M0</p> <p>Note: SOC_CON15[4] should be configured to 1'b1 when select SRADC4, otherwise it should be configured to 1'b0.</p>

GRF GPIOOD_IOMUX_L

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable</p>
15	RO	0x0	reserved
14:12	RW	0x0	<p>gpio0d3_sel 3'b000: GPIO0_D3_u 3'b001: CODEC_DAC_DL_M0 3'b010: PWM8 3'b011: TKEY3_M1 3'b100: UART2_RTSN_M0 3'b101: SPI1_MOSI_M2 3'b110: I2S_SCLK_TX_M1 Others: Reserved</p>
11	RO	0x0	reserved
10:8	RW	0x0	<p>gpio0d2_sel 3'b000: GPIO0_D2_u 3'b001: CODEC_ADC_D_M0 3'b010: PWM6_M1 3'b011: TKEY2_M1 3'b100: UART2_CTSN_M0 3'b101: SPI1_CLK_M2 3'b110: I2S_MCLK_M1 Others: Reserved</p>
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio0d1_sel 3'b000: GPIO0_D1_u 3'b001: CODEC_SYNC_M0 3'b010: PWM5_M1 3'b011: TKEY1_M1 3'b100: UART1_TX_M2 3'b101: SPI1_CS0n_M2 3'b110: I2C0_SCL_M3 3'b111: PWM_AUDIO_R_M2
3	RO	0x0	reserved
2:0	RW	0x0	gpio0d0_sel 3'b000: GPIO0_D0_u 3'b001: CODEC_CLK_M0 3'b010: PWM4_M1 3'b011: TKEY0_M1 3'b100: UART1_RX_M2 3'b101: TKEY_DRIVE_M4 3'b110: I2C0_SDA_M3 3'b111: PWM_AUDIO_L_M2

GRF GPIOOD_IOMUX_H

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:11	RO	0x0	reserved
10:8	RW	0x0	gpio0d6_sel 3'b000: GPIO0_D6_u 3'b001: I2C2_SCL_M0 3'b010: PWM11 3'b011: TKEY6_M1 3'b100: TKEY_DRIVE_M5 3'b101: I2C1_SCL_M3 3'b110: I2S_SDI_M1 3'b111: PWM_AUDIO_R_M3
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	gpio0d5_sel 3'b000: GPIO0_D5_u 3'b001: I2C2_SDA_M0 3'b010: PWM10 3'b011: TKEY5_M1 3'b100: UART2_TX_M0 3'b101: I2C1_SDA_M3 3'b110: I2S_SDO0_M1 3'b111: PWM_AUDIO_L_M3
3	RO	0x0	reserved
2:0	RW	0x0	gpiood4_sel 3'b000: GPIO0_D4_u 3'b001: PMIC_INT_M0 3'b010: PWM9 3'b011: TKEY4_M1 3'b100: UART2_RX_M0 3'b101: SPI1_MISO_M2 3'b110: I2S_LRCK_TX_M1 Others: Reserved

GRF GPIO1A_IOMUX_L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	gpio1a3_sel 1'b0: GPIO1_A3 1'b1: FSPI0_D3
11:9	RO	0x0	reserved
8	RW	0x0	gpio1a2_sel 1'b0: GPIO1_A2 1'b1: FSPI0_D2
7:5	RO	0x0	reserved
4	RW	0x0	gpio1a1_sel 1'b0: GPIO1_A1 1'b1: FSPI0_D1
3:1	RO	0x0	reserved
0	RW	0x0	gpio1a0_sel 1'b0: GPIO1_A0 1'b1: FSPI0_D0

GRF GPIO1A_IOMUX_H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x0	reserved
4	RW	0x0	gpio1a5_sel 1'b0: GPIO1_A5 1'b1: FSPI0_CSn
3:1	RO	0x0	reserved
0	RW	0x0	gpio1a4_sel 1'b0: GPIO1_A4 1'b1: FSPI0_CLK

GRF GPIO1B_IOMUX_L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	gpio1b3_sel 1'b0: GPIO1_B3 1'b1: HyperBus_RWDS
11:10	RO	0x0	reserved
9:8	RW	0x0	gpio1b2_sel 2'b00: GPIO1_B2 2'b01: HyperBus_CSn 2'b10: FSPI1_CSn 2'b11: Reserved
7:5	RO	0x0	reserved
4	RW	0x0	gpio1b1_sel 1'b0: GPIO1_B1 1'b1: HyperBus_CKN
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio1b0_sel 2'b00: GPIO1_B0 2'b01: HyperBus_CKP 2'b10: FSPI1_CLK 2'b11: Reserved

GRF GPIO1B_IOMUX_H

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x0	gpio1b7_sel 2'b00: GPIO1_B7 2'b01: HyperBus_D3 2'b10: FSPI1_D1 2'b11: Reserved
11:10	RO	0x0	reserved
9:8	RW	0x0	gpio1b6_sel 2'b00: GPIO1_B6 2'b01: HyperBus_D2 2'b10: FSPI1_D2 2'b11: Reserved
7:5	RO	0x0	reserved
4	RW	0x0	gpio1b5_sel 1'b0: GPIO1_B5 1'b1: HyperBus_D1
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio1b4_sel 2'b00: GPIO1_B4 2'b01: HyperBus_D0 2'b10: FSPI1_D0 2'b11: Reserved

GRF GPIO1C IOMUX

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	gpio1c3_sel 1'b0: GPIO1_C3 1'b1: HyperBus_D7
11:9	RO	0x0	reserved
8	RW	0x0	gpio1c2_sel 1'b0: GPIO1_C2 1'b1: HyperBus_D6
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio1c1_sel 1'b0: GPIO1_C1 1'b1: HyperBus_D5
3:2	RO	0x0	reserved
1:0	RW	0x0	gpio1c0_sel 2'b00: GPIO1_C0 2'b01: HyperBus_D4 2'b10: FSPI1_D3 2'b11: Reserved

GRF_GPIO1D_IOMUX_L

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0	reserved
1:0	RW	0x0	gpio1d0_sel 2'b00: GPIO1_D0_u 2'b01: PWM7_M1 2'b10: SPI0_CS1n 2'b11: SPI1_CS1n

GRF_GPIO0A_P

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0a7_p 2'b00: high-z 2'b10: pull down Other: reserved
13:12	RW	0x2	gpio0a6_p 2'b00: high-z 2'b10: pull down Other: reserved
11:10	RW	0x2	gpio0a5_p 2'b00: high-z 2'b10: pull down Other: reserved

Bit	Attr	Reset Value	Description
9:8	RW	0x1	gpio0a4_p 2'b00: high-z 2'b01: pull up Other: reserved
7:6	RW	0x1	gpio0a3_p 2'b00: high-z 2'b01: pull up Other: reserved
5:4	RW	0x1	gpio0a2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio0a1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio0a0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIO0B_P

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0b7_p 2'b00: high-z 2'b10: pull down Other: reserved
13:12	RW	0x2	gpio0b6_p 2'b00: high-z 2'b10: pull down Other: reserved
11:10	RW	0x2	gpio0b5_p 2'b00: high-z 2'b10: pull down Other: reserved
9:8	RW	0x2	gpio0b4_p 2'b00: high-z 2'b10: pull down Other: reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x1	gpio0b3_p 2'b00: high-z 2'b01: pull up Other: reserved
5:4	RW	0x1	gpio0b2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio0b1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio0b0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIOOC_P

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RW	0x2	gpio0c7_p 2'b00: high-z 2'b10: pull down Other: reserved
13:12	RW	0x1	gpio0c6_p 2'b00: high-z 2'b01: pull up Other: reserved
11:10	RW	0x1	gpio0c5_p 2'b00: high-z 2'b01: pull up Other: reserved
9:8	RW	0x1	gpio0c4_p 2'b00: high-z 2'b01: pull up Other: reserved
7:6	RW	0x2	gpio0c3_p 2'b00: high-z 2'b10: pull down Other: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio0c2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio0c1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio0c0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIOOD_P

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:12	RW	0x1	gpio0d6_p 2'b00: high-z 2'b01: pull up Other: reserved
11:10	RW	0x1	gpio0d5_p 2'b00: high-z 2'b01: pull up Other: reserved
9:8	RW	0x1	gpio0d4_p 2'b00: high-z 2'b01: pull up Other: reserved
7:6	RW	0x1	gpio0d3_p 2'b00: high-z 2'b01: pull up Other: reserved
5:4	RW	0x1	gpio0d2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio0d1_p 2'b00: high-z 2'b01: pull up Other: reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x1	gpio0d0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIO1A_P

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:10	RW	0x1	gpio1a5_p 2'b00: high-z 2'b01: pull up Other: reserved
9:8	RW	0x1	gpio1a4_p 2'b00: high-z 2'b01: pull up Other: reserved
7:6	RW	0x1	gpio1a3_p 2'b00: high-z 2'b01: pull up Other: reserved
5:4	RW	0x1	gpio1a2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio1a1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio1a0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIO1B_P

Address: Operational Base + offset (0x0114)

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

Bit	Attr	Reset Value	Description
15:14	RW	0x1	gpio1b7_p 2'b00: high-z 2'b01: pull up Other: reserved
13:12	RW	0x1	gpio1b6_p 2'b00: high-z 2'b01: pull up Other: reserved
11:10	RW	0x1	gpio1b5_p 2'b00: high-z 2'b01: pull up Other: reserved
9:8	RW	0x1	gpio1b4_p 2'b00: high-z 2'b01: pull up Other: reserved
7:6	RW	0x2	gpio1b3_p 2'b00: high-z 2'b10: pull down Other: reserved
5:4	RW	0x1	gpio1b2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio1b1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio1b0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIO1C_P

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x0	reserved
7:6	RW	0x1	gpio1c3_p 2'b00: high-z 2'b01: pull up Other: reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x1	gpio1c2_p 2'b00: high-z 2'b01: pull up Other: reserved
3:2	RW	0x1	gpio1c1_p 2'b00: high-z 2'b01: pull up Other: reserved
1:0	RW	0x1	gpio1c0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF GPIO1D_P

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0	reserved
1:0	RW	0x1	gpio1d0_p 2'b00: high-z 2'b01: pull up Other: reserved

GRF SOC CON0

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:9	RO	0x0	reserved
8	RW	0x0	dsp_tcm_sel3 1'b0: DSP dtcm sram(192KB) is divided into three 64KB slice, the third slice is used as system sram, can not be accessed by DSP 1'b1: DSP dtcm sram(192KB)is divided into three 64KB slice, the third slice is used by DSP sram, can not be accessed by other master except DSP

Bit	Attr	Reset Value	Description
7	RW	0x0	dsp_tcm_sel2 1'b0: DSP dtcm sram(192KB)is divided into three 64KB slice, the second slice is used as system sram, can not be accessed by DSP 1'b1: DSP dtcm sram(192KB)is divided into three 64KB slice, the second slice is used by DSP sram, can not be accessed by other master except DSP
6	RW	0x0	dsp_tcm_sel1 1'b0: DSP dtcm sram(192KB)is divided into three 64KB slice, the first slice is used as system sram, can not be accessed by DSP 1'b1: DSP dtcm sram(192KB)is divided into three 64KB slice, the first slice is used by DSP sram, can not be accessed by other master except DSP
5	RW	0x0	dsp_tcm_sel0 1'b0: DSP itcm sram(32KB) is used as system sram, can not be accessed by DSP 1'b1: DSP itcm sram(32KB) is used by DSP sram, can not be accessed by other master except DSP
4	RW	0x0	grf_con_hyperbus_fspi1_sel select hyperbus or FSPI1 to be accessed by cpu 1'b0: hyperbus can be access by cpu, and FSPI1 cannot be accessed 1'b1: FSPI1 can be access by cpu, and hyperbus cannot be accessed
3	RW	0x1	grf_con_dsp_jtagtck_gating DSP JTAG tck gating 1'b0: DSP JTAG tck is not gating 1'b1: DSP JTAG tck is gating
2	RW	0x1	grf_con_m0_jtagtck_gating M0 JTAG tck gating 1'b0: M0 JTAG tck is not gating 1'b1: M0 JTAG tck is gating
1	RW	0x1	grf_con_m4f_jtagtck_gating M4F JTAG tck gating 1'b0: M4F JTAG tck is not gating 1'b1: M4F JTAG tck is gating
0	RW	0x0	remap 1'b0: not remap 1'b1: remap

GRF SOC CON1

Address: Operational Base + offset (0x0204)

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

Bit	Attr	Reset Value	Description
15	RW	0x0	uart2_rts_inv_sel 1'b0: normal RTS connection 1'b1: invert RTS connection
14	RW	0x0	uart2_cts_inv_sel 1'b0: normal CTS connection 1'b1: invert CTS connection
13	RO	0x0	reserved
12	RW	0x0	grf_i2c2_multi_iofunc_src_sel I2C2 multiplex IO function selection. 1'b0: I2C2 select multiplex IO function from pad group 0(m0) 1'b1: I2C2 select multiplex IO function from pad group 1(m1) When SOC_CON16[12] is 1'b1, means MCU want to use i2c2 interface to transfer data, otherwise codec use i2c2 interface to transfer data.
11	RW	0x0	grf_con_vip_neg_pos_sel 1'b0: use posedge of clock to latch VICAP data 1'b0: use negedge of clock to latch VICAP data
10	RW	0x0	grf_pdm_multi_iofunc_src_sel PDM multiplex IO function selection. 1'b0: PDM select multiplex IO function from pad group 0(m0) 1'b1: PDM select multiplex IO function from pad group 1(m1)
9:8	RW	0x0	grf_i2c1_multi_iofunc_src_sel I2C1 multiplex IO function selection. 2'b00: I2C1 select multiplex IO function from pad group 0(m0) 2'b01: I2C1 select multiplex IO function from pad group 1(m1) 2'b10: I2C1 select multiplex IO function from pad group 2(m2) 2'b11: reserved
7:6	RW	0x0	grf_i2c0_multi_iofunc_src_sel I2C0 multiplex IO function selection. 2'b00: I2C0 select multiplex IO function from pad group 0(m0) 2'b01: I2C0 select multiplex IO function from pad group 1(m1) 2'b10: I2C0 select multiplex IO function from pad group 2(m2) 2'b11: reserved
5	RW	0x0	grf_pwm1_multi_iofunc_src_sel PWM1 multiplex IO function selection. 1'b0: PWM1 select multiplex IO function from pad group 0(m0) 1'b1: PWM1 select multiplex IO function from pad group 1(m1)
4	RW	0x0	grf_pwm0_multi_iofunc_src_sel PWM0 multiplex IO function selection. 1'b0: PWM0 select multiplex IO function from pad group 0(m0) 1'b1: PWM0 select multiplex IO function from pad group 1(m1)

Bit	Attr	Reset Value	Description
3:2	RW	0x0	grf_spi1_multi_iofunc_src_sel SPI1 multiplex IO function selection. 2'b00: SPI1 select multiplex IO function from pad group 0(m0) 2'b01: SPI1 select multiplex IO function from pad group 1(m1) 2'b10: SPI1 select multiplex IO function from pad group 2(m2) 2'b11: reserved
1	RW	0x0	grf_spi0_multi_iofunc_src_sel SPI0 multiplex IO function selection. 1'b0: SPI0 select multiplex IO function from pad group 0(m0) 1'b1: SPI0 select multiplex IO function from pad group 1(m1)
0	RW	0x0	grf_mclkout_i2s0_ioe I2S0 mclk input/output control signal 1'b0: mclk output enable 1'b1: mclk input enable

GRF SOC CON2

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	cahce1_flush_req M0 cahce flush request 1'b0: no operation 1'b1: request cache to flush data in cache to memory space
13	RW	0x0	dcache0_flush_req M4F data cahce flush request 1'b0: no operation 1'b1: request cache to flush data in cache to memory space
12	RW	0x0	icache0_flush_req M4F code cache flush request 1'b0: no operation 1'b1: request cache to flush data in cache to memory space
11	RW	0x1	intmem1_vad_memauto_gating_en memory auto gating control 1'b1: disable auto gating 1'b0: enable auto gating
10	RW	0x1	intmem1_memauto_gating_en memory auto gating control 1'b1: disable auto gating 1'b0: enable auto gating

Bit	Attr	Reset Value	Description
9	RW	0x1	intmem0_memauto_gating_en memory auto gating control 1'b1: disable auto gating 1'b0: enable auto gating
8	RO	0x0	reserved
7	RW	0x1	ahbbuffer_memauto_gating_en memory auto gating control 1'b1: disable auto gating 1'b0: enable auto gating
6	RW	0x1	bootrom_memauto_gating_en memory auto gating control 1'b1: disable auto gating 1'b0: enable auto gating
5	RO	0x0	reserved
4	RW	0x0	grf_pdm_clk_s_g pdm_clk_s clock gating signal 1'b0: PDM_CLK_S_M0/PDM_CLK_S_M1 is gating 1'b1: PDM_CLK_S_M0/PDM_CLK_S_M1 is not gating
3	RW	0x0	uart1_rts_inv_sel 1'b0: normal RTS connection 1'b1: invert RTS connection
2	RW	0x0	uart0_rts_inv_sel 1'b0: normal RTS connection 1'b1: invert RTS 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 CON3

Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x0	reserved
9	RW	0x0	grf_rpc_initial_state initial state
8	RW	0x0	grf_rds_clk_smp_sel hyperbus sample clk select signal
7:0	RW	0x00	grf_con_rds_delay_adj delayline number for hyperbus

GRF SOC CON4

Address: Operational Base + offset (0x0210)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	dcache0_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
14	RW	0x0	dcache0_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
13	RW	0x0	dcache0_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
12	RW	0x0	dcache0_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
11:8	RW	0x0	dcache0_spra_ms memory read/write margin setting
7	RW	0x0	icache0_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
6	RW	0x0	icache0_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
5	RW	0x0	icache0_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
4	RW	0x0	icache0_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
3:0	RW	0x0	icache0_spra_ms memory read/write margin setting

GRF SOC CONS

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	vop_dprf_pd power shut down enable input 1'b1: power down 1'b0: power on
12:11	RW	0x1	vop_sprf_tsel Timing adjustment for debug purpose 2'b00: Fast 2'b01: Normal Use 2'b10: Slow1 2'b11: Slow2
10	RW	0x1	vop_sprf_turbo Reading timing setting for debug purpose 1'b0: Debug 1'b1: Normal Use
9	RW	0x0	vop_sprf_rtsel Reading timing setting for debug purpose 1'b0: Normal Use 1'b1: Debug
8	RW	0x0	vop_sprf_pd power shut down enable input 1'b1: power down 1'b0: power on
7	RW	0x0	cache1_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
6	RW	0x0	cache1_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
5	RW	0x0	cache1_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
4	RW	0x0	cache1_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
3:0	RW	0x0	cache1_spra_ms memory read/write margin setting

GRF SOC CON6

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13	RW	0x0	bootrom_rom_pd power shut down enable input 1'b1: power down 1'b0: power on
12	RW	0x0	dsp_core_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
11:8	RW	0x0	dsp_core_spra_ms memory read/write margin setting
7	RW	0x0	dsp_core_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
6	RW	0x0	dsp_core_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
5	RW	0x0	dsp_core_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
4:3	RW	0x1	dsp_sprf_tsel Timing adjustment for debug purpose 2'b00: Fast 2'b01: Normal Use 2'b10: Slow1 2'b11: Slow2
2	RW	0x1	dsp_sprf_turbo Reading timing setting for debug purpose 1'b0: Debug 1'b1: Normal Use
1	RW	0x0	dsp_sprf_rtsel Reading timing setting for debug purpose 1'b0: Normal Use 1'b1: Debug

Bit	Attr	Reset Value	Description
0	RW	0x0	dsp_sprf_pd power shut down enable input 1'b1: power down 1'b0: power on

GRF SOC CON7

Address: Operational Base + offset (0x021c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	crypto_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
14	RW	0x0	crypto_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
13	RW	0x0	crypto_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
12	RW	0x0	crypto_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
11:8	RW	0x0	crypto_spra_ms memory read/write margin setting
7	RW	0x0	usb_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
6	RW	0x0	usb_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
5	RW	0x0	usb_spra_pd power shut down enable input 1'b1: power down 1'b0: power on

Bit	Attr	Reset Value	Description
4	RW	0x0	usb_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
3:0	RW	0x0	usb_spra_ms memory read/write margin setting

GRF SOC CON8

Address: Operational Base + offset (0x0220)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x0	ahbbuffer_spra_ret Retention mode enable input 1'b1: retention 1'b0: not retention
14	RW	0x0	ahbbuffer_spra_nap Nap mode enable input 1'b1: enable 1'b0: disable
13	RW	0x0	ahbbuffer_spra_pd power shut down enable input 1'b1: power down 1'b0: power on
12	RW	0x0	ahbbuffer_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
11:8	RW	0x0	ahbbuffer_spra_ms memory read/write margin setting
7:3	RO	0x0	reserved
2	RW	0x0	vip_dprf_pd power shut down enable input 1'b1: power down 1'b0: power on
1	RW	0x0	sdmmc_dprf_pd power shut down enable input 1'b1: power down 1'b0: power on
0	RW	0x0	crypto_dprf_pd power shut down enable input 1'b1: power down 1'b0: power on

GRF SOC CON9

Address: Operational Base + offset (0x0224)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:13	RO	0x0	reserved
12	RW	0x0	intmem1_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
11:8	RW	0x0	intmem1_spra_ms memory read/write margin setting
7:5	RO	0x0	reserved
4	RW	0x0	intmem0_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
3:0	RW	0x0	intmem0_spra_ms memory read/write margin setting

GRF SOC CON10

Address: Operational Base + offset (0x0228)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14:13	RW	0x1	codec_sprf_tsel Timing adjustment for debug purpose 2'b00: Fast 2'b01: Normal Use 2'b10: Slow1 2'b11: Slow2
12	RW	0x1	codec_sprf_turbo Reading timing setting for debug purpose 1'b0: Debug 1'b1: Normal Use
11	RW	0x0	codec_sprf_rtsel Reading timing setting for debug purpose 1'b0: Normal Use 1'b1: Debug

Bit	Attr	Reset Value	Description
10	RW	0x0	codec_sprf_pd power shut down enable input 1'b1: power down 1'b0: power on
9:8	RW	0x1	pdm_sprf_tsel Timing adjustment for debug purpose 2'b00: Fast 2'b01: Normal Use 2'b10: Slow1 2'b11: Slow2
7	RW	0x1	pdm_sprf_turbo Reading timing setting for debug purpose 1'b0: Debug 1'b1: Normal Use
6	RW	0x0	pdm_sprf_rtsel Reading timing setting for debug purpose 1'b0: Normal Use 1'b1: Debug
5	RW	0x0	pdm_sprf_pd power shut down enable input 1'b1: power down 1'b0: power on
4:3	RW	0x1	vad_sprf_tsel Timing adjustment for debug purpose 2'b00: Fast 2'b01: Normal Use 2'b10: Slow1 2'b11: Slow2
2	RW	0x1	vad_sprf_turbo Reading timing setting for debug purpose 1'b0: Debug 1'b1: Normal Use
1	RW	0x0	vad_sprf_rtsel Reading timing setting for debug purpose 1'b0: Normal Use 1'b1: Debug
0	RW	0x0	vad_sprf_pd power shut down enable input 1'b1: power down 1'b0: power on

GRF SOC CON11

Address: Operational Base + offset (0x022c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RO	0x0	reserved
4	RW	0x0	dsp_spra_mse memory read/write margin enable 1'b1: enable 1'b0: disable
3:0	RW	0x0	dsp_spra_ms memory read/write margin setting

GRF SOC CON12

Address: Operational Base + offset (0x0230)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RW	0x000	dsp_dtcm_spra_nap Control nap mode of DSP DTCM memory slice0 to slice15, dsp_dtcm_spra_nap_low[0] corresponding to slice0,dsp_dtcm_spra_nap_low[1] corresponding to slice1,... 1'b1: Nap Mode enable 1'b0: Nap Mode disable
3:2	RO	0x0	reserved
1:0	RW	0x0	dsp_itcm_spra_nap Control nap mode of DSP ITCM memory slice0 to slice7, dsp_itcm_spra_nap[0] corresponding to slice0,dsp_itcm_spra_nap[1] corresponding to slice1,... 1'b1: Nap Mode enable 1'b0: Nap Mode disable

GRF SOC CON13

Address: Operational Base + offset (0x0234)

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

Bit	Attr	Reset Value	Description
15:4	RW	0x000	dsp_dtcm_spra_ret Control retention mode of DSP DTCM memory slice0 to slice15, dsp_dtcm_spra_ret_low[0] corresponding to slice0,dsp_dtcm_spra_ret_low[1] corresponding to slice1,... 1'b1: Retention 1'b0: not retention
3:2	RO	0x0	reserved
1:0	RW	0x0	dsp_itcm_spra_ret Control retention mode of DSP ITCM memory slice0 to slice7, dsp_itcm_spra_ret[0] corresponding to slice0,sp_itcm_spra_ret[1] corresponding to slice1 ... 1'b1: retention 1'b0: not retention

GRF SOC CON14

Address: Operational Base + offset (0x0238)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RW	0x000	dsp_dtcm_spra_pd Control power shut down of DSP DTCM memory slice0 to slice7, dsp_itcm_spra_pd[0] corresponding to slice0,dsp_itcm_spra_pd[1] corresponding to slice1,... 1'b1: power down 1'b0: power on
3:2	RO	0x0	reserved
1:0	RW	0x0	dsp_itcm_spra_pd Control power shut down of DSP ITCM memory slice0 to slice7, dsp_itcm_spra_pd[0] corresponding to slice0,dsp_itcm_spra_pd[1] corresponding to slice1,... 1'b1: power down 1'b0: power on

GRF SOC CON15

Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	RW	0xff	<p>grf_saradc_ien control SARADC pad input enable signal 1'b0: input enable 1'b1: input disable Note: refer to GPIO0C_IOMUX_L/GPIO0C_IOMUX_H</p>

GRF SOC CON16

Address: Operational Base + offset (0x0240)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	<p>write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable</p>
15	RO	0x0	reserved
14	RW	0x0	<p>grf_force_sel_i2c2 1'b1: MCU force to use i2c2 interface to transfer data 1'b0: i2c2 interface used by MCU or codec depends on soc_status[19]</p>
13	RW	0x0	<p>grf_codec_pmic_multi_iofunc_src_sel codec multiplex IO function selection. 1'b0: codec select multiplex IO function from pad group 0(m0) 1'b1: codec select multiplex IO function from pad group 1(m1)</p>
12	RW	0x0	<p>grf_i2c_trans_req 1'b1: MCU want to use I2C2 interface to transfer data 1'b0: codec use I2C2 interface to transfer data</p>
11	RW	0x0	<p>grf_touchkey6_multi_iofunc_src_sel touch key6 multiplex IO function selection. 1'b0: touch key6 select multiplex IO function from pad group 0(m0) 1'b1: touch key6 select multiplex IO function from pad group 1(m1)</p>
10	RW	0x0	<p>grf_touchkey5_multi_iofunc_src_sel touch key5 multiplex IO function selection. 1'b0: touch key5 select multiplex IO function from pad group 0(m0) 1'b1: touch key5 select multiplex IO function from pad group 1(m1)</p>
9	RW	0x0	<p>grf_touchkey4_multi_iofunc_src_sel touch key4 multiplex IO function selection. 1'b0: touch key4 select multiplex IO function from pad group 0(m0) 1'b1: touch key4 select multiplex IO function from pad group 1(m1)</p>

Bit	Attr	Reset Value	Description
8	RW	0x0	grf_touchkey3_multi_iofunc_src_sel touch key3 multiplex IO function selection. 1'b0: touch key3 select multiplex IO function from pad group 0(m0) 1'b1: touch key3 select multiplex IO function from pad group 1(m1)
7	RW	0x0	grf_touchkey2_multi_iofunc_src_sel touch key2 multiplex IO function selection. 1'b0: touch key2 select multiplex IO function from pad group 0(m0) 1'b1: touch key2 select multiplex IO function from pad group 1(m1)
6	RW	0x0	grf_touchkey1_multi_iofunc_src_sel touch key1 multiplex IO function selection. 1'b0: touch key1 select multiplex IO function from pad group 0(m0) 1'b1: touch key1 select multiplex IO function from pad group 1(m1)
5	RW	0x0	grf_touchkey0_multi_iofunc_src_sel touch key0 multiplex IO function selection. 1'b0: touch key0 select multiplex IO function from pad group 0(m0) 1'b1: touch key0 select multiplex IO function from pad group 1(m1)
4	RW	0x0	grf_i2s_multi_iofunc_src_sel I2S multiplex IO function selection. 1'b0: I2S select multiplex IO function from pad group 0(m0) 1'b1: I2S select multiplex IO function from pad group 1(m1)
3	RW	0x0	grf_uart2_multi_iofunc_src_sel UART2 multiplex IO function selection. 1'b0: UART2 select multiplex IO function from pad group 0(m0) 1'b1: UART2 select multiplex IO function from pad group 1(m1)
2:1	RW	0x0	grf_uart1_multi_iofunc_src_sel UART1 multiplex IO function selection. 2'b00: UART1 select multiplex IO function from pad group 0(m0) 2'b01: UART1 select multiplex IO function from pad group 1(m1) 2'b10: UART1 select multiplex IO function from pad group 2(m2)
0	RW	0x0	grf_uart0_multi_iofunc_src_sel UART0 multiplex IO function selection. 1'b0: UART0 select multiplex IO function from pad group 0(m0) 1'b1: UART0 select multiplex IO function from pad group 1(m1)

GRF SOC CON17

Address: Operational Base + offset (0x0244)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:10	RO	0x0	reserved
9:8	RW	0x0	grf_con_ch8_req_ack_sel DMAC handshake8 select signal. 2'b00: PWM0 use handshake8 2'b01: PWM1 use handshake8 2'b10: PWM2 use handshake8 2'b11: PWM3 use handshake8
7:6	RW	0x0	grf_con_ch3_req_ack_sel DMAC handshake3 select signal. 2'b00: UART0 RX use handshake3 2'b01: UART1 RX use handshake3 2'b10: UART2 RX use handshake3 Others: Reserved
5:4	RW	0x0	grf_con_ch2_req_ack_sel DMAC handshake2 select signal. 2'b00: UART0 RX use handshake2 2'b01: UART1 RX use handshake2 2'b10: UART2 RX use handshake2 Others: Reserved
3:2	RW	0x0	grf_con_ch1_req_ack_sel DMAC handshake1 select signal. 2'b00: UART0 RX use handshake1 2'b01: UART1 RX use handshake1 2'b10: UART2 RX use handshake1 Others: Reserved
1:0	RW	0x0	grf_con_ch0_req_ack_sel DMAC handshake0 select signal. 2'b00: UART0 TX use handshake0 2'b01: UART1 TX use handshake0 2'b10: UART2 TX use handshake0 Others: Reserved

GRF SOC CON18

Address: Operational Base + offset (0x0248)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_rd_mask0 eFuse read mask control, each bit indicates its corresponding word can be read or not. Total 32 words eFuse memory. 1'b0: corresponding word can be read. 1'b1: corresponding word can't be read.

GRF SOC CON19

Address: Operational Base + offset (0x024c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_rd_mask1 eFuse read mask control, each bit indicates its corresponding word can be read or not. Total 32 words eFuse memory. 1'b0: corresponding word can be read. 1'b1: corresponding word can't be read.

GRF SOC CON20

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_rd_mask2 eFuse read mask control, each bit indicates its corresponding word can be read or not. Total 32 words eFuse memory. 1'b0: corresponding word can be read. 1'b1: corresponding word can't be read.

GRF SOC CON21

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_rd_mask3 eFuse read mask control, each bit indicates its corresponding word can be read or not. Total 32 words eFuse memory. 1'b0: corresponding word can be read. 1'b1: corresponding word can't be read.

GRF SOC CON22

Address: Operational Base + offset (0x0258)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_pg_mask0 program mask control, each bit indicates its corresponding word can be program or not. Total 32 words eFuse memory. 1'b0: the whole corresponding word can be programmed. 1'b1: the whole corresponding word can't be programmed.

GRF SOC CON23

Address: Operational Base + offset (0x025c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_pg_mask1 program mask control, each bit indicates its corresponding word can be program or not. Total 32 words eFuse memory. 1'b0: the whole corresponding word can be programmed. 1'b1: the whole corresponding word can't be programmed.

GRF SOC CON24

Address: Operational Base + offset (0x0260)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_pg_mask2 program mask control, each bit indicates its corresponding word can be program or not. Total 32 words eFuse memory. 1'b0: the whole corresponding word can be programmed. 1'b1: the whole corresponding word can't be programmed.

GRF SOC CON25

Address: Operational Base + offset (0x0264)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	efuse_pg_mask3 program mask control, each bit indicates its corresponding word can be program or not. Total 32 words eFuse memory. 1'b0: the whole corresponding word can be programmed. 1'b1: the whole corresponding word can't be programmed.

GRF SOC CON26

Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:0	RW	0x2003f000	spi2apb_addr_range_low spi2apb master address control signal. memory address between 0x80000 and spi2apb_addr_range_low can be access by spi2apb; spi2apb_addr_range_low should be increased by n*2KB.

GRF SOC CON27

Address: Operational Base + offset (0x026c)

Bit	Attr	Reset Value	Description
31:0	RW	0x20040000	spi2apb_addr_range_high spi2apb master address control signal. memory address big than spi2apb_addr_range_high can be access by spi2apb; spi2apb_addr_range_low should be increased by n*2KB.

GRF SOC CON28

Address: Operational Base + offset (0x0270)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5	RW	0x0	PeriMstSwLinkFwd_pwrSall 1'b0: bus return error response 1'b1: bus return ok response
4	RW	0x0	WlanSramSwLinkFwd_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
3	RW	0x0	VadMstSwFwd_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
2	RW	0x0	DmaSSwFwd_WifiSlvSwFwd_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
1	RW	0x0	DmaSSwFwd_PmuSlvSwFwd_pwrStall 1'b0: bus return error response 1'b1: bus return ok response
0	RW	0x0	DmaSSwFwd_PerislvSwFwd_pwrStall 1'b0: bus return error response 1'b1: bus return ok response

GRF SOC CON29

Address: Operational Base + offset (0x0274)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:5	RW	0x000	grf_saradc_ana_reg_high Reserved for test mode
4	RW	0x1	grf_saradc_vol_sel saradc voltage select signal 1'b1: select internal reference voltage 1'b0: select AVDD
3:0	RW	0x0	grf_saradc_ana_reg_low Reserved for test mode

GRF SOC CON30

Address: Operational Base + offset (0x0278)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x1	grf_tsadc_tsen_pd The power down signal of temperature sensor 1'b0: enable sensor 1'b1: power down
1	RW	0x0	grf_tsadc_dig_bypass The enable signal of the digital bypass function 1'b0: output directly 1'b1: synchronize output
0	RW	0x0	grf_tsadc_clk_sel The enable signal of the clock inverter for the analog to digital interface 1'b0: invert 1'b1: not invert

GRF SOC CON31

Address: Operational Base + offset (0x027c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x00	grf_tsadc_ana_reg Reserved
7	RW	0x0	grf_tsadc_ana_reg7 The common mode selection signal 1'b0: enable 1'b1: disable
6	RW	0x0	grf_tsadc_ana_reg6 The chopper clk delay selection signal 1'b0: enable 1'b1: disable
5	RW	0x0	grf_tsadc_ana_reg5 The vbe filter selection signal of temperature sensor 1'b0: enable 1'b1: disable
4	RW	0x0	grf_tsadc_ana_reg4 The vbe selection signal of temperature sensor 1'b0: enable 1'b1: disable
3	RW	0x0	grf_tsadc_ana_reg3 The bandgap filter selection signal of temperature sensor 1'b0: enable 1'b1: disable

Bit	Attr	Reset Value	Description
2	RW	0x0	grf_tsadc_ana_reg2 The power down signal of internal buffer in temperature sensor 1'b0: enable 1'b1: disable
1	RW	0x0	grf_tsadc_ana_reg1 The power down signal of vtemp buffer in temperature sensor 1'b0: enable 1'b1: disable
0	RW	0x0	grf_tsadc_ana_reg0 The power down signal of internal bandgap buffer in temperature sensor 1'b0: enable 1'b1: disable

GRF SOC STATUS

Address: Operational Base + offset (0x0280)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25	RO	0x0	PeriMstSwLinkFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
24	RO	0x0	WlanSramSwLinkFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
23	RO	0x0	VadMstSwFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
22	RO	0x0	DmaSSwFwd_WifiSlvSwFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
21	RO	0x0	DmaSSwFwd_PmuSlvSwFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
20	RO	0x0	DmaSSwFwd_PerIsvSwFwd_pwrActive 1'b0: bus return error response 1'b1: bus return ok response
19	RO	0x0	grf_codec_i2c_trans_ack 1'b1: MCU can used I2C2 iomux to trans data now 1'b0: codec is using I2C2 iomux to trans current, MCU cannot use i2c2 iomux
18	RO	0x0	grf_st_timer_1ch_en timer1 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable

Bit	Attr	Reset Value	Description
17	RO	0x0	cache1_flush_ack 1'b1: m0 cache flush done 1'b0: m0 cache flush not done yet
16	RO	0x0	usb2otg_utmi_vbusvalid usb vbusvalid signal
15:14	RO	0x0	usb2otg_utmi_linestate usb linestate signal
13	RO	0x0	usb2otg_utmi_bvalid usb bvalid signal
12	RO	0x0	usb2otg_utmi_iddig usb iddig signal
11	RO	0x0	icache0_flush_ack 1'b1: m4f icache flush done 1'b0: m4f icache flush not done yet
10	RO	0x0	dcache0_flush_ack 1'b1: M4F dcache flush done 1'b0: M4F dcache flush not done yet
9	RO	0x0	reserved
8	RO	0x0	grf_st_npor_bypassn NPOR bypass state 1'b1: NPOR source is from IO 1'b0: NPOR source is from internal NPOR module
7	RO	0x0	grf_st_vpll_lock VPLL lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state
6	RO	0x0	grf_st_timer5_en timer channel 5 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable
5	RO	0x0	grf_st_timer4_en timer channel 4 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable
4	RO	0x0	grf_st_timer3_en timer channel 3 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable
3	RO	0x0	grf_st_timer2_en timer channel 2 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable

Bit	Attr	Reset Value	Description
2	RO	0x0	grf_st_timer1_en timer channel 1 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable
1	RO	0x0	grf_st_timer0_en timer channel 0 enable signal 1'b1: timer channel is enable 1'b0: timer channel is disable
0	RO	0x0	grf_st_gpll_lock GPLL lock status. 1'b0: PLL is not in lock state 1'b1: PLL is in lock state

GRF MCU0 CON0

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	m4_skew Configure this bit LOW when m4 FCLK can guarantee an exact multiple of 10ms, that is no skew. Otherwise, configure this bit HIGH.
23:0	RW	0x0000000	m4_tenms Provides an integer value to compute a 10ms (100Hz) delay from the reference clock (m4 FCLK). For example, 10MHz * 10ms - 1 = 10MHz / 100Hz - 1 = 0x1869F

GRF MCU0 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.
15:10	RO	0x0	reserved
9	RW	0x1	grf_con_m4_dap_dcache 1'b0: Memory is not cacheable when DAP access Dcache 1'b1: Memory is cacheable when DAP access Dcache
8	RW	0x1	grf_con_m4_dap_fixmaster 1'b0: DAP can not access Dcache by SBus 1'b1: DAP can access Dcache by SBus

Bit	Attr	Reset Value	Description
7	RW	0x0	grf_con_m4f_fpu_disable Disable m4 FPU unit 1'b0: Enable 1'b1: Disable
6	RW	0x0	grf_con_m4f_mpu_disable Disable m4 MPU unit 1'b1: Disable 1'b0: Enable
5	RW	0x0	grf_con_m4f_pmu_enable Enable m4 power management unit to carry out powerdown. 1'b1: enable 1'b0: disable
4	RW	0x0	grf_con_m4f_dbgen External debug enable. 1'b1: enable 1'b0: disable
3	RW	0x0	grf_con_m4f_dbgrestart External debug restart request. 1'b1: assert 1'b0: de-assert
2	RW	0x0	grf_con_m4f_edbgrp External debug request. 1'b1: assert 1'b0: de-assert
1	RW	0x0	grf_con_m4f_nmi Non-maskable interrupt. 1'b1: assert 1'b0: de-assert
0	RW	0x0	grf_con_m4f_rxev Send an event to wake up m4 from WFE. 1'b1: assert 1'b0: de-assert

GRF MCU1 CON0

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:0	RW	0x00000000	grf_con_m0_stalib m0 system time calibration

GRF MCU1 CON1

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:14	RO	0x0	reserved
13:6	RW	0x00	grf_con_m0_mpu_disable Disable m4 MPU unit 1'b1: Disable 1'b0: Enable
5	RW	0x0	grf_con_m0_pmu_enable Enable m4 power management unit to carry out powerdown. 1'b1: enable 1'b0: disable
4	RW	0x0	grf_con_m0_dbgen External debug enable. 1'b1: enable 1'b0: disable
3	RW	0x0	grf_con_m0_dbgrestart External debug restart request. 1'b1: assert 1'b0: de-assert
2	RW	0x0	grf_con_m0_edbgrp External debug request. 1'b1: assert 1'b0: de-assert
1	RW	0x0	grf_con_m0_nmi Non-maskable interrupt. 1'b1: assert 1'b0: de-assert
0	RW	0x0	grf_con_m0_rxev Send an event to wake up m4 from WFE. 1'b1: assert 1'b0: de-assert

GRF_DSP_CON0

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:6	RO	0x0	reserved
5	RW	0x0	runstall dsp runstall

Bit	Attr	Reset Value	Description
4	RW	0x0	statvectorsel dsp statvectorsel
3	RO	0x0	reserved
2	RW	0x0	breakoutack Acknowledges that BreakOut has been received
1	RW	0x0	breakin External debug interrupt
0	RW	0x0	ocdhaltonreset Enters OCDHaltMode if this signal is sampled asserted on reset.

GRF DSP CON1

Address: Operational Base + offset (0x0324)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	altresetvec Alternate reset vector address

GRF DSP CON2

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x0	reserved
6	RW	0x0	itcm_mem_auto_gating_en DSP ITCM memory auto clock gating enable 1'b0: Disable 1'b1: Enable
5	RW	0x0	dtag_mem_auto_gating_en DSP DTCM memory auto clock gating enable 1'b0: Disable 1'b1: Enable
4	RW	0x0	prefetch_ram_auto_gating_en Prefetch Memory auto clock gating enable 1'b0: Disable 1'b1: Enable
3	RW	0x0	dtag_mem_auto_gating_en DTag Memory auto clock gating enable 1'b0: Disable 1'b1: Enable
2	RW	0x0	dcache_mem_auto_gating_en DCache Memory auto clock gating enable 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	itag_mem_auto_gating_en ITag Memory auto clock gating enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	icache_mem_auto_gating_en ICache Memory auto clock gating enable 1'b0: Disable 1'b1: Enable

GRF SOC UOC0

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RW	0x1	otgphy_txbitstuffenh High-Byte transmit Bit-Stuffing enable 1'b1: Enable 1'b0: Disable
14	RW	0x1	otgphy_txbitstuffen Low-Byte transmit Bit-Stuffing enable 1'b1: Enable 1'b0: Disable
13	RW	0x0	otgphy_siddq IDDDQ test enable
12	RW	0x0	otgphy_port_reset Per-Port reset
11:10	RW	0x2	otgphy_refclkSEL Reference clock select
9:8	RW	0x1	otgphy_refclkdiv Reference clock division
7:5	RW	0x3	otgphy_tune Vbus valid threshold adjustment.
4	RW	0x0	otgphy_disable OTG block disable 1'b1: The OTG block is powered down 1'b0: The OTG block is powered up
3:1	RW	0x1	otgphy_compdistune Disconnect threshold adjustment.
0	RW	0x1	otgphy_common_on_n Common block power-down control.

GRF SOC UOC1

Address: Operational Base + offset (0x0344)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x0	otgphy_txrisetune HS transmitter rise/fall time adjustment.
13:12	RW	0x1	otgphy_txhsxvtune Transmitter high-speed crossover adjustment.
11:8	RW	0x6	togphy_txvreftune HS DC voltage level adjustment.
7:4	RW	0xf	otgphy_txfslstune FS/LS source impedance adjustment.
3	RW	0x1	otgphy_txpreemphasistune HS transmitter Pre-Emphasis enable.
2:0	RW	0x3	otgphy_sqrxtune Squelch threshold adjustment.

GRF SOC UOC2

Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15	RO	0x0	reserved
14	RW	0x1	grf_con_otg_utmi_iddig_pmu Mini A/B plug indicator
13	RW	0x0	grf_con_otg_dbnce_fltr_bypass debounce filter bypass
12:11	RW	0x0	grf_con_otg_scaledown_mode scale down mode
10	RW	0x1	otgphy_sleepm Sleep assertion
9	RW	0x0	otgphy_vregtune 2.5V voltage regulator HS boost adjustment
8	RW	0x0	grf_con_otg_utmi_termselect USB termination select
7:6	RW	0x0	grf_con_otg_utmi_xcvrselect Transceiver select
5:4	RW	0x0	grf_con_otg_utmi_opmode UTMI+ operational mode
3	RW	0x1	grf_con_otg_utmi_suspend_n Suspend Assertion

Bit	Attr	Reset Value	Description
2	RW	0x0	otgphy_soft_con_sel software controllable select
1	RW	0x0	otgphy_vbusvldextsel External Vbus valid select
0	RW	0x0	otgphy_vbusvldext External Vbus valid indicator

GRF MCU0 STATUS

Address: Operational Base + offset (0x0380)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21	RO	0x0	m4_fpidc Input denormal of FPU exception
20	RO	0x0	m4_fpdc Divide-by-zero of FPU exception
19	RO	0x0	m4_fpioc Invalid operation of FPU exception
18	RO	0x0	m4_fpufc Underflow of FPU exception
17	RO	0x0	m4_fpofc Overflow of FPU exception
16	RO	0x0	m4_fpixc Inexact of FPU exception
15:8	RO	0x00	grf_st_m4f_currpri Indicates what interrupt priority is being used now. Only the upper 3 bits of interrupt priority is significant for 8 supported levels.
7:6	RO	0x0	reserved
5	RO	0x0	grf_st_m4f_lockup Indicates that m4 is locked up. 1'b1: m4 is locked up 1'b0: m4 isn't locked up
4	RO	0x0	grf_st_m4f_gatehclk Indicates that HCLK can be gated because m4 is asleep without the debugger being active. 1'b1: m4 hclk can be safely gated 1'b0: m4 hclk can't be safely gated
3	RO	0x0	grf_st_m4f_degestarted Handshake for m4_dbgrestart. Active High.
2	RO	0x0	grf_st_m4f_halted Indicates that m4 is in halting mode debug. 1'b1: m4 is in debug state 1'b0: m4 isn't in debug state

Bit	Attr	Reset Value	Description
1	RO	0x0	grf_st_m4f_sleepdeep Active only when M4F sleeping is high. Indicates that the sleepdeep bit in the NVIC is set to high and m4 is in deep sleep mode.
0	RO	0x0	grf_st_m4f_sleeping Indicates that m4 is in sleep mode. 1'b1: m4 is idle, waiting for an interrupt 1'b0: m4 isn't idle

GRF MCU1 STATUS

Address: Operational Base + offset (0x0384)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	grf_st_m0_wicenack Indicates M0 wicen acknowledge.
8	RO	0x0	grf_st_m0_sleepholdackn Indicates M0 sleep hold acknowledge.
7	RO	0x0	grf_st_m0_sleeping Indicates M0 sleeping.
6	RO	0x0	grf_st_m0_sleepdeep Indicates M0 sleepdeep.
5	RO	0x0	grf_st_m0_wakeup Indicates M0 wakeup.
4	RO	0x0	grf_st_m0_hclk_gate Indicates that HCLK can be gated because m4 is asleep without the debugger being active. 1'b1: m0 hclk can be safely gated 1'b0: m0 hclk can't be safely gated
3	RO	0x0	grf_st_m0_lockup Indicates that m0 is locked up. 1'b1: m0 is locked up 1'b0: m0 isn't locked up
2	RO	0x0	grf_st_m0_txev M0 TXEV
1	RO	0x0	grf_st_m0_halted Indicates that m0 is in halting mode debug. 1'b1: m0 is in debug state 1'b0: m0 isn't in debug state
0	RO	0x0	grf_st_m0_dbgrestarted Handshake for m0 dbgrestart. Active High.

GRF DSP STAT0

Address: Operational Base + offset (0x0388)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	iram0ladstore This bit differentiates between an instruction doing a data load or store to instruction RAM, and the instruction fetch logic using the instruction RAM to fetch instructions.
7	RO	0x0	pwaitmode Indicates that the processor is in sleep mode.
6	RO	0x0	pfaultinfovalid Strobe signal that is asserted for one cycle every time PFaultInfo signal changes its value.
5	RO	0x0	pfatalerror Sticky fatal error notification signal that is asserted when a fatal error condition occurs.
4	RO	0x0	doubleexceptionerror Single cycle assertion for every time double exception faults occur.
3	RO	0x0	breakout Indication from the processor that it has entered OCD halt mode
2	RO	0x0	breakinack Acknowledges that BreakIn has been received
1	RO	0x0	debugmode Same as XOCDMode but not maskable by software
0	RO	0x0	xocdmode Indicates that the processor is in OCD halt mode.

GRF_DSP_STAT1

Address: Operational Base + offset (0x038c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pfaultinfo Fault information signal. This signal mirrors the internal Fault Information Register and provides the source and severity of the fault.

GRF_GRF_FAST_BOOT

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0	reserved
0	RW	0x1	grf_fast_boot fast boot enable signal

GRF GRF FAST BOOT ADDR

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	grf_fast_boot_addr fast boot address

GRF WLAN CON

Address: Operational Base + offset (0x0500)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:8	RW	0x00	wlan_rev wlan revision
7	RW	0x0	wlan_ext_irq wlan external interrupt
6	RW	0x0	rpu_ready_irq_en RPU ready interrupt enable signal
5	RW	0x0	rpu_wakeup_irq_en RPU wakeup interrupt enable signal
4	RW	0x0	rpu_sleep_irq_en RPU sleep interrupt enable signal
3	RW	0x0	wlan_write_int_en wlan write buffer interrupt enable signal
2	RW	0x1	xtalmod_2 1'b0: not output 1'b1: output
1	RW	0x1	xtalmod_1 1'b0: not output 1'b1: output
0	RW	0x1	xtalmod_0 1'b0: not output 1'b1: output

GRF WLANCLK CON

Address: Operational Base + offset (0x0504)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x0	reserved
6	RW	0x0	bt_pti1 bt_pti1

Bit	Attr	Reset Value	Description
5	RW	0x0	wlan_aon_wakeup wlan aon wakeup control 1'b1: wlan aon wakeup now 1'b0: wlan keep idle
4	RW	0x0	mcu_clk_force Overrides the internal wlan clock gating 1'b0: clock gating is not overridden 1'b1: clock gating is overridden and will take the value of rpu_clk_flevel
3	RW	0x0	mcu_clk_flevel When forced, overrides the internal wlan clock gating 1'b0: the gate is closed and no clock will propagate to the output 1'b1: the gate is open and the clock will propagate to the output
2	RW	0x0	rpu_clk_force Overrides the internal wlan clock gating 1'b0: clock gating is not overridden 1'b1: clock gating is overridden and will take the value of rpu_clk_flevel
1	RW	0x0	rpu_clk_flevel When forced, overrides the internal wlan clock gating 1'b0: the gate is closed and no clock will propagate to the output 1'b1: the gate is open and the clock will propagate to the output
0	RW	0x0	wlan_sys_clk_gate 1'b0: enable, not gating 1'b1: disable, gating

GRF WLAN GPIO IN

Address: Operational Base + offset (0x0508)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wlan_gpio_in WLAN GPIO in

GRF WLAN GPIO OUT

Address: Operational Base + offset (0x050c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	wlan_gpio_out WLAN GPIO out

GRF WLAN STATUS

Address: Operational Base + offset (0x0580)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:4	W1C	0x000	wlan_mac_phy_debug_bus Debug output signals
3	W1C	0x0	wlan_write_data_int_status when wlan write data through interconnect, generate an interrupt.
2	W1C	0x0	rpu_ready_status WLAN ready status
1	W1C	0x0	rpu_wakeup WLAN wakeup state
0	W1C	0x0	rpu_sleep WLAN sleep state

GRF_USB2_DISCONNECT_CON

Address: Operational Base + offset (0x0680)

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 GRF pclk.

GRF_USB2_LINESTATE_CON

Address: Operational Base + offset (0x0684)

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 GRF pclk.

GRF_USB2_BVALID_CON

Address: Operational Base + offset (0x0688)

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 GRF pclk.

GRF_USB2_ID_CON

Address: Operational Base + offset (0x068c)

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 GRF pclk.

GRF_USB2_DETECT_IRQ_ENABLE

Address: Operational Base + offset (0x0690)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x0	reserved
6	RW	0x0	otg0_disconnect_neg_irq_en Enable OTG 0 disconnect negedge detect interrupt. 1'b0: disable 1'b1: enable
5	RW	0x0	otg0_disconnect_pos_irq_en Enable OTG 0 disconnect posedge detect interrupt. 1'b0: disable 1'b1: enable
4	RW	0x0	otg0_id_neg_irq_en Enable OTG 0 id negedge detect interrupt. 1'b0: disable 1'b1: enable
3	RW	0x0	otg0_id_pos_irq_en Enable OTG 0 id posedge detect interrupt. 1'b0: disable 1'b1: enable
2	RW	0x0	otg0_bvalid_neg_irq_en Enable OTG 0 bvalid negedge detect interrupt. 1'b0: disable 1'b1: enable
1	RW	0x0	otg0_bvalid_pos_irq_en Enable OTG 0 bvalid posedge detect interrupt. 1'b0: disable 1'b1: enable
0	RW	0x1	otg0_linesate_irq_en Enable OTG 0 linesate detect interrupt. 1'b0: disable 1'b1: enable

GRF USB2 DETECT IRQ STATUS

Address: Operational Base + offset (0x0694)

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.

Bit	Attr	Reset Value	Description
15:7	RO	0x0	reserved
6	RW	0x0	otg0_disconnect_neg_irq OTG 0 disconnect negedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
5	RW	0x0	otg0_disconnect_pos_irq OTG 0 disconnect posedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
4	RW	0x0	otg0_id_neg_irq OTG 0 id negedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
3	RW	0x0	otg0_id_pos_irq OTG 0 id posedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
2	RW	0x0	otg0_bvalid_neg_irq OTG 0 bvalid negedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
1	RW	0x0	otg0_bvalid_pos_irq OTG 0 bvalid posedge interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted
0	RW	0x1	otg0_linestate_irq OTG 0 linestate change interrupt status. 1'b0: no IRQ 1'b1: IRQ asserted

GRF USB2 DETECT IRQ STATUS CLR

Address: Operational Base + offset (0x0698)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x0	reserved
6	RW	0x0	otg0_disconnect_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status

Bit	Attr	Reset Value	Description
5	RW	0x0	otg0_disconnect_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
4	RW	0x0	otg0_id_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
3	RW	0x0	otg0_id_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
2	RW	0x0	otg0_bvalid_neg_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
1	RW	0x0	otg0_bvalid_pos_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status
0	RW	0x1	otg0_linestate_irq_clr Interrupt status clear. 1'b0: hold status, no action 1'b1: clear interrupt status

GRF SPINLOCK REG x

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	spinlock_reg_x spinlock register When all bits are 0, it can be written with new value, otherwise it cannot be written. It is cleared to 0 when writing 0x0.

GRF OS REG0

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register0 OS register 0

GRF OS REG1

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register1 OS register 1

GRF OS REG2

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register2 OS register 2

GRF OS REG3

Address: Operational Base + offset (0x080c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register3 OS register 0

GRF OS REG4

Address: Operational Base + offset (0x0810)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register4 OS register 4

GRF OS REG5

Address: Operational Base + offset (0x0814)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register5 OS register 5

GRF OS REG6

Address: Operational Base + offset (0x0818)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register6 OS register 6

GRF OS REG7

Address: Operational Base + offset (0x081c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	os_register7 OS register 7

GRF SOC VERSION

Address: Operational Base + offset (0x0820)

Bit	Attr	Reset Value	Description
31:0	RO	0x00002206	soc_version SOC version

3.5 Interface Description

NA

3.6 Application Notes

NA

Rockchip Confidential

Chapter 4 Cortex-M4 with Cache

4.1 Overview

The Cortex-M4 processor is a low-power processor that features low gate count, low interrupt latency, and low-cost debug. It is intended for deeply embedded applications that require fast interrupt response features. And the Level 1 Instruction/Data Cache designed for Cortex-M4 can improve the memory access performance significantly.

4.1.1 Cortex-M4

The following features may or may not be present in the actual product. Please contact Rockchip for the actual feature configurations and the third-party licensing requirements. The Cortex-M4 processor incorporates:

- A low gate count processor core
 - A subset of the Thumb instruction set, defined in the *ARMv7-M Architecture Reference Manual*
 - The ARMv7E-M architecture profile, including the DSP extension
 - Banked Stack Pointer (SP)
 - Hardware integer divide instructions, SDIV and UDIV
 - Handler and Thread modes
 - Thumb and Debug states
 - Support for interruptible-continued instructions LDM, STM, PUSH, and POP for low interrupt latency
 - Automatic processor state saving and restoration for low latency Interrupt Service Routine (ISR) entry and exit
 - Support for ARMv6 little-endian accesses and unaligned accesses
- A Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low-latency interrupt processing
 - 80 external interrupts
 - 8 levels (3 bits) of interrupt priority
 - Dynamic reprioritization of interrupts
 - Priority grouping for selection of preempting interrupt levels and non-preempting interrupt levels
 - Support for tail-chaining and late arrival of interrupts, which enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts
 - Processor state automatically saved on interrupt entry, and restored on interrupt exit, with no instruction overhead
 - An external Non Maskable Interrupt (NMI)
- Multiple high-performance bus interfaces
 - Three Advanced High-performance Bus-Lite (AHB-Lite) interfaces: ICode, DCode, and System bus interfaces
 - Private Peripheral Bus (PPB) based on Advanced Peripheral Bus (APB) interface
 - Memory access alignment
 - Write buffer for buffering of write data
- A low-cost debug solution
 - Debug access to all memory and registers in the system, including access to memory-mapped devices, access to internal core registers when the core is halted, and access to debug control registers even while SYSRESETn is asserted
 - Serial Wire Debug Port (SW-DP) and Advanced High-performance Bus Access Port (AHB-AP) interface for debug access
 - Flash Patch and Breakpoint (FPB) unit for implementing 8 hardware breakpoints and code patches
 - Data Watchpoint and Trace (DWT) unit for implementing 4 watchpoints, data tracing, and system profiling
- A Memory Protection Unit (MPU)
 - 8 memory regions

- Sub Region Disable (SRD), enabling efficient use of memory regions
- The ability to enable a background region that implements the default memory map attributes
- A Floating Point Unit (FPU)
 - 32-bit instructions for single-precision (C float) data-processing operations
 - Combined Multiply and Accumulate instructions for increased precision (Fused MAC)
 - Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square-root
 - Hardware support for denormals and all IEEE rounding modes
 - 32 dedicated 32-bit single-precision registers, also addressable as 16 double-word registers
 - Decoupled three stage pipeline

4.1.2 Cache0

The Cache0 is acted as the Level 1 Cache of Cortex-M4 and used to improve the data access performance from MCU to memory system.

The Cache Controller supports the following features:

- Support 16KB I-Cache and 16KB D-Cache
- Support 2-way set-associative with 32Byte cache line size for each cache
- Support four 32bit AHB-Lite bus slave interface
 - I-Cache slave for the Code (ICode/DCode Mux) bus of Cortex-M4
 - D-Cache slave for the System bus of Cortex-M4
 - I-Cache register configuration
 - D-Cache register configuration
- Support two 128bit AHB-Lite bus master interface to SoC interconnect
 - I-Cache master
 - D-Cache master
- Support memory cacheable access and peripheral bypass access
- Support Write-Through with No-Write-Allocate and Read-Allocate
- Support Write-Back with Write-Allocate and Read-Allocate
- Support cache maintenance operations (clean/invalidate/clean & invalidate) by address
- Support cache invalidate/clean & invalidate operations for all cache lines
- Support cache initialization by software configuration
- Support RAM debug mode for tag RAM and data RAM
- Support MPU mode for Cortex-M4
- Support two interrupt source for I-Cache and D-Cache respectively

4.2 Block Diagram

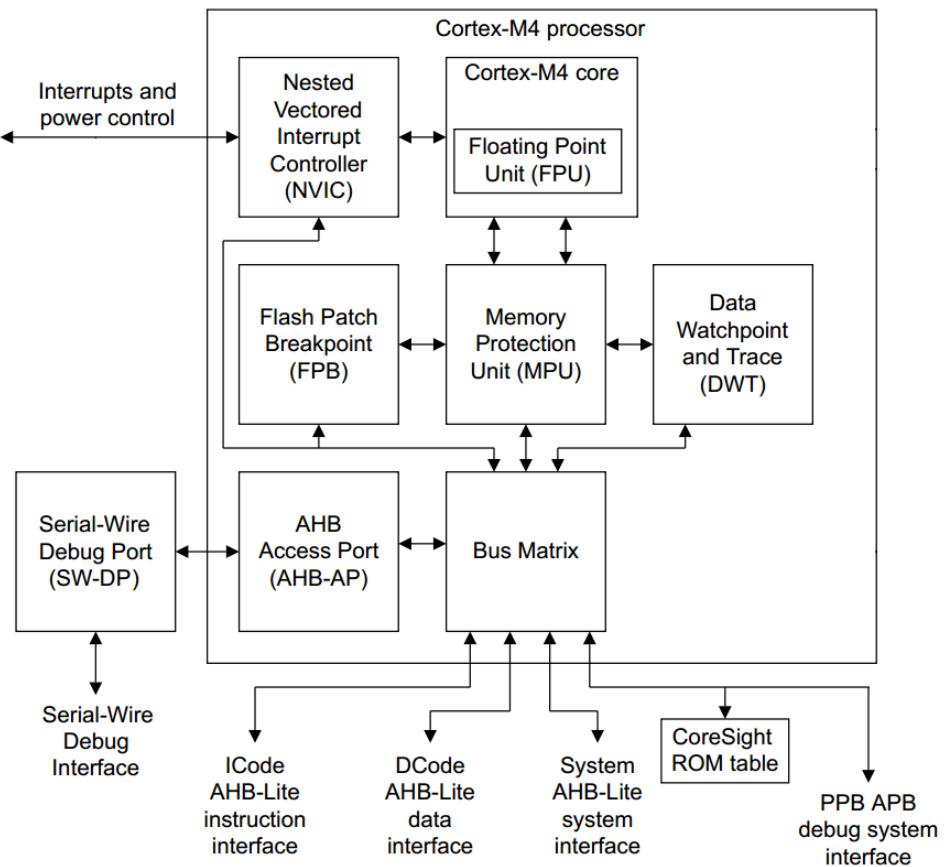


Fig. 4-1 Block Diagram of Cortex-M4

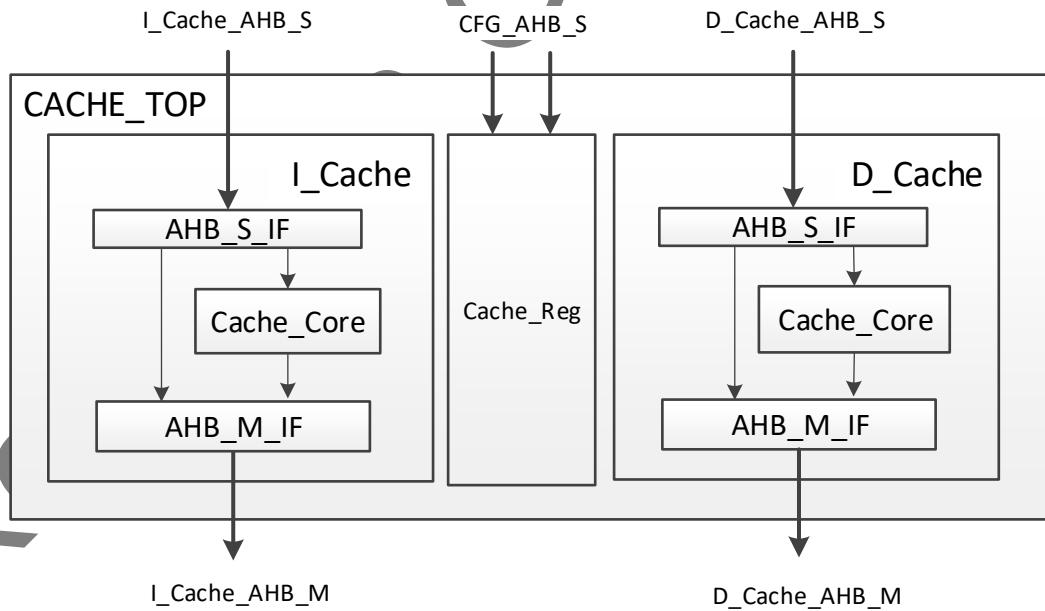


Fig. 4-2 Block Diagram of Cache

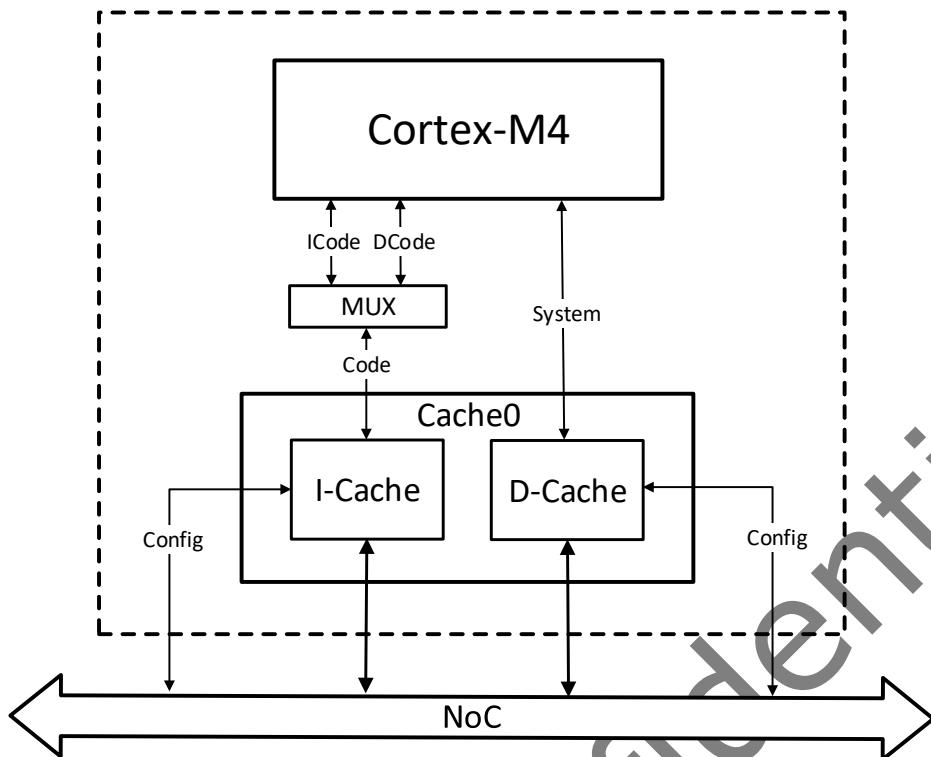


Fig. 4-3 Block Diagram of Cortex-M4 with Cache

4.3 Function Description

The Cortex-M4 processor contains a bus matrix that arbitrates accesses to both the external memory system and to the internal System Control Space (SCS) and debug components. A traditional Harvard memory architecture separates instruction fetches and operand data

references onto specific bus interfaces regardless of access address. The Cortex-M4 of this chip provides a modified Harvard memory architecture, which supports both instruction fetches and data accesses over the Code bus or the System bus by address range.

ICode memory interface

Instruction fetches from Code memory space, 0x00000000 to 0x1FFFFFFC, are performed over the 32-bit AHB-Lite bus. The Debugger cannot access this interface. All fetches are word-wide. The number of instructions fetched per word depends on the code running and the alignment of the code in memory.

DCode memory interface

Data and debug accesses to Code memory space, 0x00000000 to 0x1FFFFFFF, are performed over the 32-bit AHB-Lite bus. Core data accesses have a higher priority than debug accesses on this bus. This means that debug accesses are waited until core accesses have completed when there are simultaneous core and debug access to this bus.

Control logic in this interface converts unaligned data and debug accesses into two or three aligned accesses, depending on the size and alignment of the unaligned access. This stalls any subsequent data or debug access until the unaligned access has completed.

Code memory interface

We combine the ICode and DCode buses into a single unified Code bus for the I-Cache in the implementation. There is bus multiplexer which can arbitrate between simultaneous accesses initiated by the ICode and DCode masters. DCode has a higher priority than ICode.

System interface

Instruction fetches and data and debug accesses to address ranges 0x20000000 to 0xDFFFFFFF and 0xE0100000 to 0xFFFFFFFF are performed over the 32-bit AHB-Lite bus. For simultaneous accesses to the 32-bit AHB-Lite bus, the arbitration order in decreasing

priority is: data accesses, instruction and vector fetches, debug.

The system bus interface contains control logic to handle unaligned accesses, FPB remapped accesses, and pipelined instruction fetches. But bit-band accesses are not supported in this chip.

Pipelined instruction fetches

To provide a clean timing interface on the System bus, instruction and vector fetch requests to this bus are registered. This results in an extra cycle of latency because instructions fetched from the system bus take two cycles. This also means that back-to-back instruction fetches from the system bus are not possible.

Instruction fetch requests to the Code bus are not registered. It is recommended that performance critical code must be located at the address below 0x20000000 and run from the ICode interface.

The muxed Code bus connected to the I-Cache slave is typically used for instruction fetches and data accesses of PC-relative data, while the System bus connected to D-Cache slave is typically used for operand data references to the on-chip/off-chip memories and peripheral accesses.

Private Peripheral Bus

The Private Peripheral Bus (PPB) memory region provides access to internal and external processor resources. These important modules can only be accessed by the core.

The internal PPB space, 0xE0000000 to 0xE003FFFF, provides access to:

- The Data Watchpoint and Trace (DWT)
- The Flashpatch and Breakpoint (FPB)
- The System Control Space (SCS), including the Nested Vectored Interrupt Controller (NVIC) and the Memory Protection Unit (MPU).

The external PPB (EPPB) space, 0xE0040000 to 0xE00FFFFF, provides access to:

- The ROM table for debug
- The other debug and trace components are not implemented in this chip

The following figure show the system address map of the Cortex-M4.

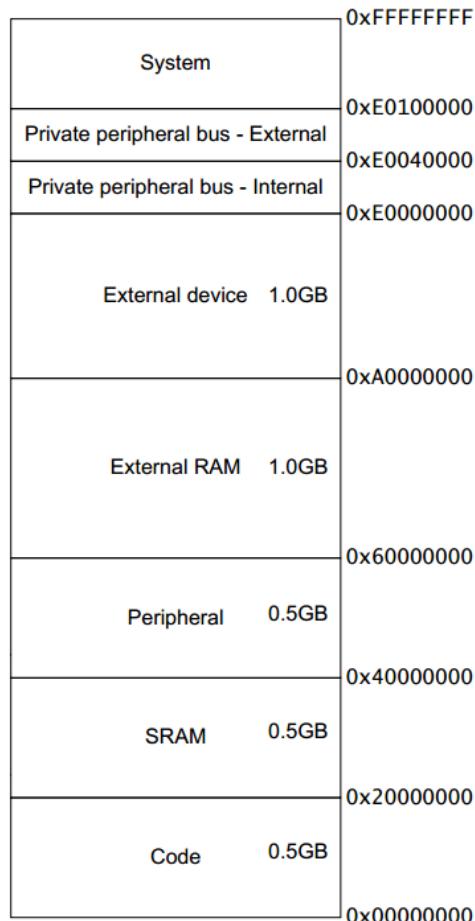


Fig. 4-4 System Address Map of the Cortex-M4

For more information about the system address map and the detail register description of the Cortex-M4, please refer to the following documents:

- The [ARMv7-M Architecture Reference Manual](#)
- The [Cortex-M4 Devices Generic User Guide](#)
- The [ARM Cortex-M4 Processor Technical Reference Manual](#)

4.4 Register Description

This section describes the control/status registers of the Cache0. Software should read and write these registers using 32-bits accesses. There are two Cache Controllers (I-Cache and D-Cache), and each of them has same register group. Therefore, two Cache Controllers' register groups have two different base addresses.

4.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
CACHE_CTRL	0x0000	W	0x0000006cc	Cache Control Register
CACHE_Maintain0	0x0004	W	0x000000000	Cache Maintain 0 Register
CACHE_Maintain1	0x0008	W	0x000000000	Cache Maintain 1 Register
CACHE_STB_TIMEOUT_CTRL	0x000c	W	0x40000000c	Store Buffer Timeout Control Register
CACHE_RAM_DEBUG	0x0010	W	0x000000000	Cache RAM Debug Register
CACHE_INT_EN	0x0020	W	0x000000000	Cache Interrupt Enable Register
CACHE_INT_ST	0x0024	W	0x000000000	Cache Interrupt Status Register
CACHE_ERR_HADDR	0x0028	W	0x000000000	Cache Error Address Register
CACHE_STATUS	0x0030	W	0x000000000	Cache Status Register
CACHE_PMU_RD_NUM_CNT	0x0040	W	0x000000000	PMU Read Number Count Register
CACHE_PMU_WR_NUM_CNT	0x0044	W	0x000000000	PMU Write Number Count Register
CACHE_PMU_RAM_RD_HIT_CNT	0x0048	W	0x000000000	PMU RAM Read Hit Count Register
CACHE_PMU_HB_RD_HIT_CNT	0x004c	W	0x000000000	PMU Hot Buffer Read Hit Count Register
CACHE_PMU_STB_RD_HIT_CNT	0x0050	W	0x000000000	PMU Store Buffer Read Hit Count Register
CACHE_PMU_RD_HIT_CNT	0x0054	W	0x000000000	PMU Read Hit Count Register
CACHE_PMU_WR_HIT_CNT	0x0058	W	0x000000000	PMU Write Hit Count Register
CACHE_PMU_RD_MISS_PENALTY_CNT	0x005c	W	0x000000000	PMU Read Miss Penalty Count Register
CACHE_PMU_WR_MISS_PENALTY_CNT	0x0060	W	0x000000000	PMU Write Miss Penalty Count Register
CACHE_PMU_RD_LAT_CNT	0x0064	W	0x000000000	PMU Read Latency Count Register
CACHE_PMU_WR_LAT_CNT	0x0068	W	0x000000000	PMU Write Latency Count Register

Name	Offset	Size	Reset Value	Description
CACHE_REVISION	0x00f0	W	0x00000100	Cache Design Revision Register

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

4.4.2 Detail Register Description

CACHE_CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	cache_pf_en Cache prefetch enable register. 1'b0: Disable 1'b1: Enable
12	RW	0x0	cache_mpu_mode Cache MPU mode enable register. When this bit is set to 1, the cacheability policy is determined by MPU of Cortex-M4. 1'b0: Enable 1'b1: Disable
11	RO	0x0	reserved
10:8	RW	0x6	stb_entry_thresh Store buffer entry threshold control register. The depth of the store buffer entry is 8. When the number of the used data entries is greater than or equal to threshold value, the write data will be written to Cache RAM.
7	RW	0x1	stb_timeout_en Store buffer timeout enable register. When this bit is set to 1, the data in the store buffer must be flushed to Cache RAM if the counter value is equal to the timeout value. 1'b0: Disable 1'b1: Enable
6	RW	0x1	cache_bypass Cache bypass mode enable register. When this bit is set to 1, data will bypass the Cache. 1'b0: Disable 1'b1: Enable
5	RW	0x0	cache_pmu_en Cache Performance Monitor Unit enable register. 1'b0: Disable 1'b1: Enable
4	RW	0x0	cache_flush Cache flush enable register. When this bit is set to 1, the dirty data is flushed to external memory and the cache line are invalidated. 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x1	cache_stb_en Cache store buffer enable register. This bit must be set to 0 when Write-Through mode is selected and must be set to 1 when Write-Back mode is selected. 1'b0: Disable 1'b1: Enable
2	RW	0x1	cache_hb_en Cache hot buffer enable register. 1'b0: Disable 1'b1: Enable
1	RW	0x0	cache_wt_en Cache mode control register. 1'b0: Write-Back 1'b1: Write-Through
0	RW	0x0	cache_en Cache initialization enable register. 1'b0: Disable 1'b1: Enable

CACHE MAINTAIN0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:5	RW	0x00000000	cache_m_addr Cache maintain start address. This address is 32Byte cache line aligned, that is, the bits[4:0] are always 0.
4:3	RO	0x0	reserved
2:1	RW	0x0	cache_m_cmd Cache maintain command register. 2'b00: Clean by address 2'b01: Invalidate by address 2'b10: Clean and Invalidate by address 2'b11: Invalidate all
0	R/W SC	0x0	cache_m_valid Cache maintain valid register. The maintenance operation is valid only when this bit is set to 1.

CACHE MAINTAIN1

Address: Operational Base + offset (0x0008)

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

Bit	Attr	Reset Value	Description
26:0	RW	0x00000000	cache_m_offset Cache maintain offset. This bit field indicates the end offset of cache line, that is, the value plus 1 determines the number of cache line to be maintained. The bit field of cache_m_addr is treated as the start offset of cache line, so the maintain address range is from (cache_m_addr * 32) to (cache_m_addr * 32 + 31 + cache_m_offset * 32).

CACHE STB TIMEOUT CTRL

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x4000000c	stb_timeout_value Store buffer timeout value.

CACHE RAM DEBUG

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ram_debug_en Cache RAM debug mode enable register. 1'b0: Disable 1'b1: Enable

CACHE INT EN

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	err_record_en AHB master bus error record enable. 1'b0: Disable 1'b1: Enable

CACHE INT ST

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	ahb_error_status Error status bit for AHB master bus. 1'b0: Nothing 1'b1: Bus error

CACHE ERR HADDR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	status_haddr Recent record of AHB bus error address.

CACHE STATUS

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	cache_flush_done Cache flush done status register. 1'b0: Nothing 1'b1: Flush done
1	RO	0x0	cache_m_busy Cache maintain busy status register. 1'b0: Idle 1'b1: Busy
0	RO	0x0	cache_init_finish Cache initialization finish status register. 1'b0: Cache is uninitialized 1'b1: Cache is initialized

CACHE PMU RD NUM CNT

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_num_cnt Total count of read transfers.

CACHE PMU WR NUM CNT

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_num_cnt Total count of write transfers.

CACHE PMU RAM RD HIT CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_ram_rd_hit_cnt Count of read hits on Cache RAM.

CACHE PMU HB RD HIT CNT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_hb_rd_hit_cnt Count of read hits on hot buffer.

CACHE PMU STB RD HIT CNT

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_stb_rd_hit_cnt Count of read hits on store buffer.

CACHE PMU RD HIT CNT

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_hit_cnt Total count of read hits.

CACHE PMU WR HIT CNT

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_hit_cnt Total count of write hits.

CACHE PMU RD MISS PENALTY CNT

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_miss_penalty_cnt Total count of read miss penalty (in clock cycles).

CACHE PMU WR MISS PENALTY CNT

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_miss_penalty_cnt Total count of write miss penalty (in clock cycles).

CACHE PMU RD LAT CNT

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_rd_lat_cnt Total count of read latency (in clock cycles).

CACHE PMU WR LAT CNT

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pmu_wr_lat_cnt Total count of write latency (in clock cycles).

CACHE REVISION

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000100	revision Cache revision number.

4.5 Interface Description

Table 4-1 Serial Wire Debug Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
TCK	I	LCD_D0/CIF_D0/I2C0_SDA_M2/TKEY12 /M4F_WFI/M4F_JTAG_TCK/M0_JTAG_TC K/AONJTAG_TCK/DSPJTAG_TCK/GPIO0 _A0_u	GRF_GPIO0A_IOMUX_L [3:0]=4'b0110
TMS	I/O	LCD_D1/CIF_D1/I2C0_SCL_M2/TKEY13 /M0_WFI/M4F_JTAG_TMS/M0_JTAG_TM S/AONJTAG_TMS/DSPJTAG_TMS/GPIO0 _A1_u	GRF_GPIO0A_IOMUX_L [7:4]=4'b0110
WFI	O	LCD_D0/CIF_D0/I2C0_SDA_M2/TKEY12 /M4F_WFI/M4F_JTAG_TCK/M0_JTAG_TC K/AONJTAG_TCK/DSPJTAG_TCK/GPIO0 _A0_u	GRF_GPIO0A_IOMUX_L [3:0]=4'b0101

Notes: Unused Module Pin is tied to zero! I=input, O=output, I/O=input/output, bidirectional

4.6 Application Notes

4.6.1 Memory Map

Cortex-M4 can access the external memory through two data paths: I-Cache side and D-Cache side. The access address of the I-Cache side (below 0x20000000 on the Code bus) can be regarded as the mirror address for the D-Cache side (above 0x20000000 on the System bus). Therefore, the I-Cache and the D-Cache master can access the same system memory through the NoC.

By placing the various code and data sections in the appropriate locations within the memory map and enabling the Cache0, overall system performance can be optimized. The following table shows the memory address map for the Cortex-M4.

Table 4-2 Memory Address Map for the Cortex-M4

Memory Module	Memory Size	Address Range for I-Cache Side Access	Address Range for D-Cache Side Access
XIP_FSPI1_MEM (GRF_SOC_CON0.grf_con_hyperbus_fspi1_sel==1'b1)	32MB	0x18000000 ~ 0x19FFFFFF	0x38000000 ~ 0x39FFFFFF
XIP HyperBus MEM (GRF_SOC_CON0.grf_con_hyperbus_fspi1_sel==1'b0)	128MB	0x18000000 ~ 0x1FFFFFFF	0x38000000 ~ 0x3FFFFFFF

Memory Module	Memory Size	Address Range for I-Cache Side Access	Address Range for D-Cache Side Access
XIP FSPIO MEM	64MB	0x10000000 ~ 0x13FFFFFF	0x30000000 ~ 0x33FFFFFF
DSP DTCM	192KB	0x00600000 ~ 0x0062FFFF	0x20600000 ~ 0x2062FFFF
DSP ITCM	32KB	0x00400000 ~ 0x00407FFF	0x20400000 ~ 0x20407FFF
System SRAM1	128KB	0x000A0000 ~ 0x000BFFFF	0x20020000 ~ 0x2003FFFF
System SRAM0 (GRF_SOC_CON0. remap==1'b1)	128KB	0x00080000 ~ 0x0009FFFF/ 0x00000000 ~ 0x0001FFFF	0x20000000 ~ 0x2001FFFF
System SRAM0 (GRF_SOC_CON0. remap==1'b0)	128KB	0x00080000 ~ 0x0009FFFF	0x20000000 ~ 0x2001FFFF
BOOTROM (GRF_SOC_CON0. remap==1'b1)	16KB	0x00100000 ~ 0x00101FFF	N/A
BOOTROM (GRF_SOC_CON0. remap==1'b0)	16KB	0x00100000 ~ 0x00101FFF/ 0x00000000 ~ 0x00001FFF	N/A

4.6.2 NVIC

The NVIC supports 66 interrupts in the implementation, each with 8 levels of priority. And all the interrupt request sources are active-high and level-sensitive.

The following table shows the IRQ number of the NVIC.

Table 4-3 IRQ Source ID of the NVIC

IRQ ID	Source	Polarity
0	DMAC_CH0	High level
1	WDT0	High level
2	TOUCH_NEG	High level
3	DMAC_CH1	High level
4	TIMER0_6CH_0	High level
5	TIMER0_6CH_1	High level
6	TIMER0_6CH_2	High level
7	TIMER0_6CH_3	High level
8	TIMER0_6CH_4	High level
9	TIMER0_6CH_5	High level
10	TIMER1_1CH	High level
11	I2C0	High level
12	I2C1	High level
13	I2C2	High level
14	SPI0	High level
15	SPI1	High level
16	UART0	High level

IRQ ID	Source	Polarity
17	UART1	High level
18	UART2	High level
19	PWM0_4CH	High level
20	PWM1_4CH	High level
21	PWM2_4CH	High level
22	SARADC	High level
23	B2A0_0	High level
24	B2A0_1	High level
25	B2A0_2	High level
26	B2A0_3	High level
27	B2A1_0	High level
28	B2A1_1	High level
29	B2A1_2	High level
30	B2A1_3	High level
31	XIPSFC0	High level
32	XIPSFC1	High level
33	VOP	High level
34	CACHE0_I	High level
35	CACHE0_D	High level
36	DSP_ERROR	High level
37	HYPERX8	High level
38	USB2OTG	High level
39	USB2OTG_BVALID	High level
40	USB2OTG_ID	High level
41	USB2OTG_LINESTATE	High level
42	USB2OTG_DISCONNECT	High level
43	SD/MMC	High level
44	AUDPWM	High level
45	PVTM	High level
46	CRYPTO	High level
47	CIF	High level
48	PMU	High level
49	GPIO0	High level
50	GPIO1	High level
51	TRIM	High level
52	I2S0	High level
53	I2S1	High level
54	PDM	High level
55	VAD	High level
56	EFUSE	High level
57	TOUCH_POS	High level
58	PMIC	High level
59	SPI2APB	High level
60	DMAC_CH2	High level

IRQ ID	Source	Polarity
61	DMAC_CH3	High level
62	DMAC_CH4	High level
63	DMAC_CH5	High level
64	TSADC	High level
65	LPW	High level

The NVIC and the processor core interface are closely coupled, to enable low latency interrupt processing and efficient processing of late arriving interrupts. NVIC registers are located within the SCS space. You can access all NVIC registers using byte, halfword, and word accesses unless otherwise stated in the ARM documents.

The NVIC maintains knowledge of the stacked, or nested, interrupts to enable tail-chaining of interrupts. You can only fully access the NVIC from privileged mode, but you can cause interrupts to enter a pending state in user mode if you enable the Configuration and Control Register. Any other user mode access causes a bus fault.

Setting GRF MCU0_CON1.m4_nmi to high will generate a NMI to the NVIC, and you can get the current status of the Cortex-M4 from GRF MCU0_STATUS. Please refer to the Chapter 4 for more information.

In this multi-processor system, the output TXEV of the Cortex-M4 is connected to the input RXEV of the Cortex-M0, and the input RXEV of the Cortex-M4 is connected to the output TXEV of the Cortex-M0. So a SEV instruction executed by the Cortex-M4 can wake up the Cortex-M0 from WFE state. Similarly, a SEV instruction executed by the Cortex-M0 can wake up the Cortex-M4 from WFE state.

4.6.3 Debug

The implemented debug functionality mainly includes: processor halt, single-step, processor core register access, Vector Catch, unlimited software breakpoints, hardware breakpoints, watchpoints, and full system memory access.

In order to enable the external debug access, you should configure

GRF MCU0_CON1.dbgen to high in the bootrom code. Because the IO Pins have been muxed to the SWD function as default, you can connect a debugger to the M4 system directly.

In the original architecture, the DAP accesses on the System bus are always non-cacheable and not under the control of the MPU, which means that you cannot access the D-Cache using a debugger when the MPU mode is enabled. But there is no problem for the I-Cache because the memory attribute on the Code bus is always cacheable. Applying some function switches to resolve this D-Cache issue, you must ensure both GRF MCU0_CON1.m4_dap_fixmaster and GRF MCU0_CON1.m4_dap_dcache are set to high as default.

4.6.4 Cache Initialization

The Cache0 is bypassed after reset and need to be initialized by the Cortex-M4. The basic initialization flow is:

1. Configure to Write-Back or Write-Through mode by setting CACHE_CTRL.cache_wt_en and start initialization by configuring CACHE_CTRL.cache_en to high while keeping the Cache in bypass mode (CACHE_CTRL.cache_bypass is high). The store buffer must and only can be enabled for Write-Back mode, so you must disable the store buffer when Write-Through mode is selected. You can also enable the hot buffer for better performance. It is recommended that configure the I-Cache to Write-Through mode and configure the D-Cache to Write-Back mode.
2. Wait for cache initialization to be finished by polling CACHE_STATUS.cache_init_finish.
3. Disable cache bypass mode by configuring CACHE_CTRL.cache_bypass to low. Now that the Cache is enabled.
4. Simply configure CACHE_CTRL.cache_bypass to bypass or enable the Cache again after the initialization flow.

4.6.5 Cache Maintenance

The Cache Controller support the following cache maintenance operations:

- Clean by address: push the specified cache line data to external memory if it is valid and dirty.
- Invalidate by address: unconditionally clear the valid and dirty data of the specified cache lines.
- Clean and invalidate by address: clean and then invalidate the specified cache lines.
- Invalidate all: unconditionally clear the valid and dirty data for all the cache lines.
- Clean and invalidate all (Flush): clean and then invalidate all the cache lines.

For the operation of cache clean and invalidate all, configure CACHE_CTRL.cache_flush to high first, and then wait for flush to be done by polling CACHE_STATUS.cache_flush_done. Finally, if the flush is done, you should configure CACHE_CTRL.cache_flush to low.

For the other four maintenance operations, you should set the maintenance command and the address range of the cache lines by configuring CACHE_MAINTAIN0 and CACHE_MAINTAIN1, then start by configuring CACHE_MAINTAIN0.cache_m_valid to high. When the operation is finished, CACHE_MAINTAIN0.cache_m_valid will be automatically cleared to low. You can also poll CACHE_STATUS.cache_m_busy to judge whether the operation is done.

4.6.6 Cache MPU Mode

The Memory Protection Unit (MPU) is a hardware block implemented in the Cortex-M4. If a program accesses a memory location that is prohibited by the MPU, the processor generates a Mem_Manage fault. This causes a fault exception and might cause termination of the process in an OS environment. In an OS environment, the kernel can update the MPU region setting dynamically based on the process to be executed. Typically, an embedded OS uses the MPU for memory protection.

The MPU have a default background region, and the memory attributes of the MPU regions can be modified by programming the MPU registers. After enabling the MPU, please use a DSB followed by an ISB instruction or exception return to ensure that the new MPU configuration is used by subsequent instructions.

In this chip, the cache MPU mode compatible with the MPU control is only supported by the D-Cache, and it is not supported by the I-Cache because the cacheability policy of the Code bus is tied off to cacheable.

When the MPU mode is enabled in the D-Cache, the cache attribute is determined by the interface signals of the System bus. Only if HPROTS[3] & (HPROTS[2] | MEMATTRS[0]) is true, the access attribute is cacheable.

When the MPU mode is disable in the D-Cache, the cache attribute is determined by the access address. The peripheral register access must be bypassed, so the address from 0x40000000 to 0x43FFFFFF is non-cacheable, and the other implemented memory address is cacheable.

4.6.7 Cache RAM Debug Mode

When the Cache is enabled and not in the bypass mode, you can enter the cache RAM debug mode by configuring CACHE_RAM_DEBUG.ram_debug_en to high. In the RAM debug mode, the cache RAM can be accessed directly by the specific address, and all the other cacheable memory accesses are bypassed because the cache core is disabled.

The following table shows the address map for the cache RAM debug.

Table 4-4 Address Map for the Cache RAM Debug

RAM Access Type	RAM Access Size	Address Range for I-Cache RAM Debug	Address Range for D-Cache RAM Debug
Reserved	7KB	0x000E6400 ~ 0x000E7FFF	0x000F6400 ~ 0x000F7FFF
TAG RAM1	1KB	0x000E6000 ~ 0x000E63FF	0x000F6000 ~ 0x000F63FF
Reserved	7KB	0x000E4400 ~ 0x000E5FFF	0x000F4400 ~ 0x000F5FFF
TAG RAM0	1KB	0x000E4000 ~ 0x000E43FF	0x000F4000 ~ 0x000F43FF
DATA RAM1	8KB	0x000E2000 ~ 0x000E3FFF	0x000F2000 ~ 0x000F3FFF

RAM Access Type	RAM Access Size	Address Range for I-Cache RAM Debug	Address Range for D-Cache RAM Debug
DATA RAM0	8KB	0x000E0000 ~ 0x000E1FFF	0x000F0000 ~ 0x000F1FFF

As shown in the above table, there are 2 ways for each cache, and the size of one DATA RAM is 8KB. The following explanation may help you use the RAM debug method:

- Using the word-aligned address to access the TAG RAM.
- The bits[9:2] of the TAG RAM address are acted as the bits[12:5] of the corresponding memory address.
- Only the bits[20:0] of the TAG RAM data are available, and the bits[31:21] are always zero.
 - The bit[20] is the valid flag bit.
 - The bit[19] is the dirty flag bit.
 - The bits[18:0] are acted as the bits[31:13] of the corresponding memory address.
- The bits[12:0] of the DATA RAM address are acted as the bits[12:0] of the corresponding memory address.
- In order to get the cache line data at the wanted address, you should read the both of the TAG RAMs to find the matched address, and then read data from the corresponding DATA RAM.
- If write data to the DATA RAM, 8 copies of the 32bit data on the AHB bus will be written to the 32Byte cache line.

4.6.8 Other Hints

- Two interrupt request sources are supported for the I-Cache and the D-Cache, so you can query the interrupt registers for a certain cache in the ISR if the interrupt is enabled. When CACHE_INT_EN.err_record_en is high, the Cache will generate a bus error interrupt if access to a wrong address. You can read CACHE_INT_ST.ahb_error_status for the cache interrupt status and get the recent error address information from CACHE_ERR_HADDR.status_haddr.
- In this chip, asserting SCB.AIRCR.SYSRESETREQ can reset the Cortex-M4 processor core (excluding the debug logic) and the Cache0. If you want to reset the whole Cortex-M4 subsystem consisted of the Cortex-M4 and the Cache0, you can also set CRU_SOFTRST_CON00[3] to high for an auto de-asserted reset.
- There is an external reference clock STCLK used as data for the Systick timer, and its frequency is depending on the input clock from IO. It is necessary that the frequency of the processor core clock FCLK (even if reduced during SLEEP mode) must be more than twice of the constant STCLK frequency, otherwise the cycle counting may miss a cycle and consequently timing accuracy may be lost. You can configure GRF MCU0_CON0 to redefine the calibration value.
- For higher performance with the Cache0, you may access the instruction code and the RO data via the I-Cache and access the RW data via the D-Cache.

Chapter 5 Cortex-M0

5.1 Overview

The Cortex-M0 processor is a very low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications that require an area optimized processor. And the Level 1 Instruction/Data Cache designed for Cortex-M0 can improve the memory access performance significantly.

5.1.1 Cortex-M0

The following features may or may not be present in the actual product. Please contact Rockchip for the actual feature configurations and the third-party licensing requirements. The processor features and benefits are:

- A low gate count processor that features
 - The ARMv6-M Thumb instruction set
 - Thumb-2 technology
 - Load/store-multiples and multicycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
 - Low power sleep-mode entry using Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or the return from interrupt sleep-on-exit feature
- NVIC that features
 - 32 external interrupt inputs, each with four levels of priority
 - Dedicated Non-Maskable Interrupt (NMI) input
 - Optional Wake-up Interrupt Controller (WIC), providing ultra-low power sleep mode support
- Optional debug support
 - Two hardware breakpoints
 - One watch points
 - Support Serial Wire debug connection
 - single 32-bit AMBA-3 AHB-Lite system interface

5.1.2 Cache1

The Cache1 is acted as the Level 1 Cache of Cortex-M0 and used to improve the data access performance from MCU to memory system. The internal function of Cache1 is same with I-Cache or D-Cache of M4.

The Cache Controller supports the following features:

- Support 16KB I/D-Cache
- Support 2-way set-associative with 32Byte cache line size
- Support two 32bit AHB-Lite bus slave interface
 - I/D-Cache slave for the AHB-Lite bus of Cortex-M0
 - I/D-Cache register configuration
- Support one 128bit AHB-Lite bus master interface to SoC interconnect
- Support memory cacheable access and peripheral bypass access
- Support Write-Through with No-Write-Allocate and Read-Allocate
- Support Write-Back with Write-Allocate and Read-Allocate
- Support cache maintenance operations (clean/invalidate/clean & invalidate) by address
- Support cache invalidate/clean & invalidate operations for all cache lines
- Support cache initialization by software configuration
- Support RAM debug mode for tag RAM and data RAM
- Support one interrupt source

5.2 Block Diagram

Cortex-M0 Integration architecture is shown below.

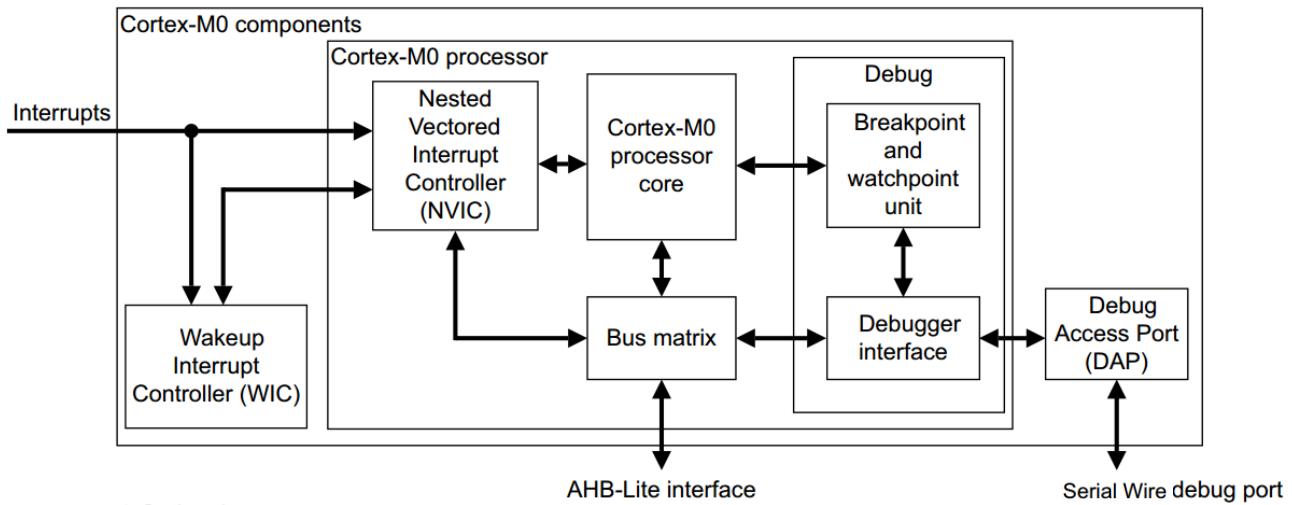


Fig. 5-1 Cortex-M0 Integration Architecture

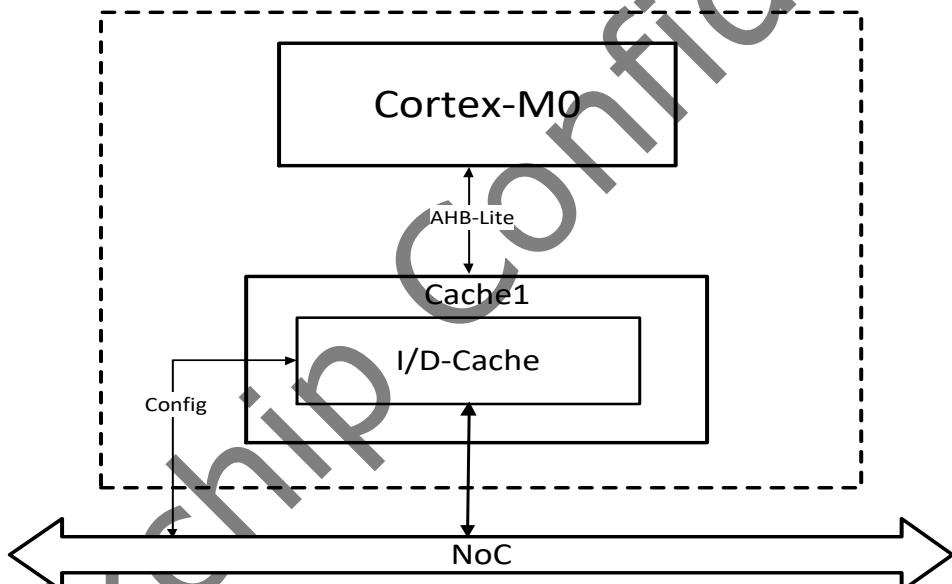


Fig. 5-2 Block Diagram of Cortex-M0 with Cache

5.3 Function Description

Please refer to the document [Cortex-M0_TechnicalReferenceManualRevC_DDI0432_r0p0-02rel0.pdf](#) and [DUI0497A_cortex_m0_r0p0_generic_ug.pdf](#) for the detail description.

5.4 Register Description

Please refer to the document [Cortex-M0_TechnicalReferenceManualRevC_DDI0432_r0p0-02rel0.pdf](#) and [DUI0497A_cortex_m0_r0p0_generic_ug.pdf](#) for the detail description.

5.5 Interface Description

Table 5-1 Cortex-M0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
m0jtag_tck	I	LCD_D0/CIF_D0/I2C0_SDA_M2/ TKEY12/M4F_WFI/M4F_JTAG_TC K/M0_JTAG_TCK/AONJTAG_TCK/ DSPJTAG_TCK/GPIO0_A0_u	GRF_GPIOA_IOMUX_L[3:0] = 4'b0111

Module Pin	Direction	Pad Name	IOMUX Setting
m0jtag_tms	I/O	LCD_D1/CIF_D1/I2C0_SCL_M2/T KEY13/M0_WFI/M4F_JTAG_TMS/ M0_JTAG_TMS/AONJTAG_TMS/D SPJTAG_TMS/GPIO0_A1_u	GRF_GPIOA_IOMUX_L[7:4] = 4'b0111

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

5.6 Application Notes

5.6.1 Clock and Reset Generation

Please refer to "Chapter CRU" for more detailed information.

5.6.2 MemoryMap for M0 BootCode

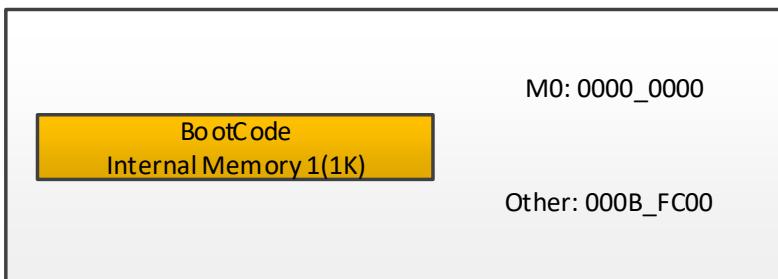


Fig. 5-3 M0 BootCode Map

The boot code of M0 storage in the internal memory1(System SRAM1), the base address for M0 Boot is from the 0x0000_0000 which seen by other master at base physical address at 0x000B_FC00.

5.6.3 Miscellaneous Signals for M0

M0 System Configure Signals

Table 5-2 M0 System Configure Signals

Signal Name	Default	Description
grf_con_m0_irqlatency[7:0]	0	Minimum number of cycles between an interrupt that becomes pending in the NVIC, and the vector fetch for that interrupt being issued
grf_con_m0_nmi	0	Non-maskable interrupt
grf_con_m0_dbgrestart	0	External restart request.
grf_con_m0_edbgqrq	0	External debug request.
grf_con_m0_stcalib[25:0]	0	Systick timer counter
grf_con_m0_rxev	0	A HIGH level on this input causes the architecture defined Event Register to be set in the Cortex-M0 processor. This causes a WFE instruction to complete. It also awakens the processor if it is sleeping as the result of encountering a WFE instruction when the Event Register is clear.
grf_con_m0_dbgen	0	SerialWire Debug enable. 0: disable 1: enable
grf_con_m0_sleepholdreqn	1	Request to extend the processor sleeping state regardless of wake-up events. If the processor acknowledges this request driving SLEEPHOLDACKn Low, this guarantees the processor remains idle even on receipt of a wake-up event

M0 System Status Signals

Table 5-3 M0 System Status Signals

Signal Name	Default	Description
grf_st_m0_dbgrestated	1	Handshake for DBGRESTART
grf_st_m0_halted	0	Indicates that the processor is in debug state. HALTED remains asserted for as long as the processor remains in debug state.
grf_st_m0_txev	0	M0 transform a event

Signal Name	Default	Description
grf_st_m0_lockup	0	Indicates that m0 is locked up
grf_st_m0_hclk_gate	0	Indicates that hclk can be gated.
grf_st_m0_wakeup	0	Indicates M0 wakeup
grf_st_m0_sleepdeep	0	Indicates the processor is idle, waiting for an interrupt on either the IRQ, NMI, or HIGH level on RXEV.
grf_st_m0_sleeping	0	Active only when SLEEPING is HIGH. Indicates that the SLEEPDEEP bit in the NVIC is set to 1.
grf_st_m0_sleepholdack	0	Indicates M0 sleepholdack

5.6.4 Memory Map

The external memory map can be divided into two ranges for Cortex-M0 access. Below 0x20000000 access operation will go through Cache1, and above 0x20000000 access operation will bypass Cache1. About detail usage of Cache1, please refer to Cache0 information in Cortex-M4 chapter.

5.6.5 Interrupt for M0

Table 5-4 M0 Interrupt

IRQ ID	IRQ Source	IRQ ID	IRQ Source
0	DMAC_CH0	16	A2B0_irq2
1	tsadc_irq	17	A2B0_irq3
2	WDT1	18	XIPSFC0
3	NA	19	XIPSFC1
4	TIMER0_6CH_0(0)	20	CACHE1_IRQ
5	TIMER0_6CH_0(1)	21	HYPERX8_IRQ
6	TIMER0_6CH_0(2)	22	PMU_IRQ
7	TIMER0_6CH_0(3)	23	WIFI_SLEEP
8	TIMER0_6CH_0(4)	24	WIFI_WAKEUP
9	TIMER0_6CH_0(5)	25	WIFI_READY
10	TIMER1_1CH	26	LPW_HOST_IRQ
11	UART0	27	DMAC_CH1
12	UART1	28	DMAC_CH2
13	UART2	29	DMAC_CH3
14	A2B0_irq0	30	DMAC_CH4
15	A2B0_irq1	31	DMAC_CH5

Chapter 6 Power Management Unit (PMU)

6.1 Overview

In order to meet low power requirements, a power management unit (PMU) is designed for controlling power resources in RK2206. The RK2206 PMU is dedicated for managing the power of the whole chip.

6.1.1 Feature

- Support 3 power domain PD_MCU, PD_WLAN, PD_PERI
- Support NIU idle interface (idle request, acknowledge and status)
- Support PMU debug through IO or UART interface
- Support wakeup sources
 - M4 interrupt
 - M0 interrupt
 - Timeout

6.2 Block Diagram

6.2.1 Power Partition

Table 6-1 RK2206 Power Domain

Subsystem	Power Domain	Module
MCU subsystem	PD_MCU	M4, M0, DSP and TCM, SFC, Bootrom, Internal Memory Interface, DMAC, Timer, UART, SPI, MailBox, EFUSE, WDT, PWM, I2C, SARADC, INTC
Periphral subsystem	PD_PERI	VICAP, Crypto, SDMMC, USB2OTG, SPI2APB
WLAN subsystem	PD_WLAN	WLAN/MAC, BUFFER
Always on modules		PMU, CRU, GRF, GPIO, TRIM, Acodec, TIMER, VAD, PDM, I2S, Internal memory

6.2.2 PMU Block Diagram

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

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

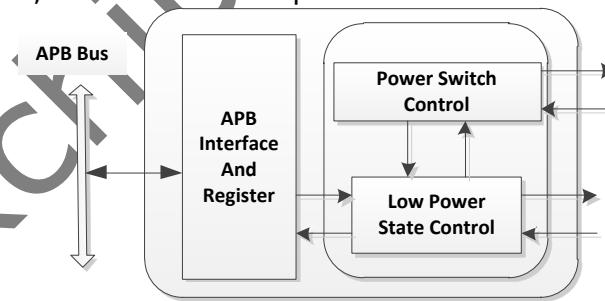


Fig. 6-1 PMU Block Diagram

6.3 Function Description

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

For example, CPU can write PMU_SFT_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 enter low power mode, after setting some power configurations, the

PMU_PWRMODE_CON[0] bit must be set 1 to enable PMU FSM. Then CPU needs to execute a WFI command to perform ready signal. After PMU detects CPU 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 RK2206 chip.

6.4 Register Description

6.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
PMU_WAKEUP_CFG	0x0000	W	0x00000000	Wake up enable for gpio posedge pulse
PMU_PWRDN_ST	0x0004	W	0x00000460	Power down status
PMU_PWRMODE_CON	0x0008	W	0x00000000	Power mode configure Register
PMU_SFT_CON	0x000c	W	0x0001e000	Software control register
PMU_INT_CON	0x0010	W	0x00000000	Pmu interrupt configure register
PMU_INT_ST	0x0014	W	0x000000e0	Pmu interrupt status register
PMU_BUS_IDLE_ST	0x0018	W	0x00000000	Pmu noc idle status register
PMU_POWER_ST	0x001c	W	0x00000000	Pmu power state register
PMU_OSC_CNT	0x0020	W	0x00000000	Pmu OSC counter register
PMU_PLLLOCK_CNT	0x0024	W	0x00000000	Pmu PLL lock counter register
PMU_PLLRST_CNT	0x0028	W	0x00000000	Pmu PLL reset counter
PMU_RET_CON	0x002c	W	0x00000000	Internal memory retention control in the low power mode
PMU_INFO_TX_CON	0x0030	W	0x00000000	Pmu info transfer control register
PMU_SYS_REG0	0x0040	W	0x00000000	Pmu system register 0
PMU_SYS_REG1	0x0044	W	0x00000000	Pmu system register 1
PMU_SYS_REG2	0x0048	W	0x00000000	Pmu system register 2
PMU_SYS_REG3	0x004c	W	0x00000000	Pmu system register 3
PMU_TIMEOUT_CNT	0x0080	W	0x00000000	Timeout wakeup counter value

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

6.4.2 Detail Register Description

PMU_WAKEUP_CFG

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	timeout_en 0: Disable 1: Enable
1	RW	0x0	m0_int_en 0: Disable 1: Enable
0	RW	0x0	m4_int_en 0: Disable 1: Enable

PMU_PWRDN_ST

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	RW	0x0	pd_mcu_dwn_stat Pd_mcu power domain down status from MTCMOS.
8	RW	0x0	pd_peri_dwn_stat Pd_peri power domain down status from MTCMOS.
7	RW	0x0	pd_wlan_dwn_stat Pd_wlan power domain down status from MTCMOS.
6	RW	0x1	wlan_sleep_state 0: Wlan subsystem not sleep 1: Wlan subsystem sleep
5	RW	0x1	wlan_sleep_now 0: Wlan subsystem sleep signal not issue 1: Wlan subsystem sleep signal issue
4	RW	0x0	m0_dbgwrupreq 0: M0 debug powerup not request 1: M0 debug powerup request
3	RW	0x0	m4_dbgwrupreq 0: M4 debug powerup not request 1: M4 debug powerup request
2	RW	0x0	m0_sleep 0: M0 not sleep 1: M0 sleep
1	RW	0x0	m4_sleep 0: M4 not sleep 1: M4 sleep
0	RW	0x0	rpu_sleep_ready 0: Wlan not boot ready 1: Wlan boot ready

PMU_PWRMODE_CON

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	pmu_sleep_pol 0: Active high 1: Active low
22	RW	0x0	vpll_pd_en The bit is valid only when bit20 set to 1 0: VPLL power down disable 1: VPLL power down enable

Bit	Attr	Reset Value	Description
21	RW	0x0	gpll_pd_en The bit is valid only when bit20 set to 1 0: GPLL power down disable 1: GPLL power down enable
20	RW	0x0	pll_pd_en 0: PLL power down disable 1: PLL power down enable
19	RW	0x0	osc_40m_gate_en 0: 40M OSC gating disable 1: 40M OSC gating enable
18	RW	0x0	wait_wakeup_begin 0: Wait wakeup begin disable 1: Wait wakeup begin enable
17	RW	0x0	rf_vdd18_dis 0: RF VDD18 supply close disable 1: RF VDD18 supply close enable Once the bit16-osc_dis set to 1, the bit17 should be set to 1 too.
16	RW	0x0	osc_dis 0: 40M OSC close disable 1: 40M OSC close enable
15	RO	0x0	reserved
14	RW	0x0	rf_reset_en 0: RF reset disable 1: RF reset enable
13	RW	0x0	mcu_pd_en 0: Pd_mcu power down disable 1: Pd_mcu power down enable
12	RW	0x0	global_int_disable 0: Enable interrupt to cpu 1: Disable interrupt to cpu
11	RW	0x0	pmu_use_if 0: PMU working in 40MHz 1: PMU working in 32KHz
10	RW	0x0	clk_top_src_gate 0: Clock top source gating disable 1: Clock top source gating enable
9	RW	0x0	clk_core_src_gate 0: Clock core source gating disable 1: Clock core source gating enable
8	RW	0x0	noc_auto_con_top 0: Clock for noc of top not masked 1: Clock for noc of top masked
7	RW	0x0	noc_auto_con_mcu 0: Clock for noc of pd_mcu not masked 1: Clock for noc of pd_mcu masked

Bit	Attr	Reset Value	Description
6	RW	0x0	noc_auto_con_peri 0: Clock for noc of pd_peri not masked 1: Clock for noc of pd_peri masked
5	RW	0x0	noc_auto_con_wlan 0: Clock for noc of pd_wlan not masked 1: Clock for noc of pd_wlan masked
4	RW	0x0	clr_top Pd_top noc interface idle enable in the low power mode 0: Not idle request 1: Idle request
3	RW	0x0	clr_mcu Pd_mcu noc interface idle enable in the low power mode 0: Not idle request 1: Idle request
2	RW	0x0	clr_peri Pd_peri noc interface idle enable in the low power mode 0: Not idle request 1: Idle request
1	RW	0x0	clr_wlan Pd_wlan noc interface idle enable in the low power mode 0: Not idle request 1: Idle request
0	RW	0x0	power_mode_en 0: Not enter the low power mode 1: Enter the low power mode

PMU_SFT_CON

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	sram_psd_cfg Internal memory power shut down config by software bit[0]: Internal memory 0 blank 0 psd control bit bit[1]: Internal memory 0 blank 1 psd control bit bit[2]: Internal memory 1 blank 0 psd control bit bit[3]: Internal memory 1 blank 1 psd control bit 0: Set psd to 0 1: Set psd to 1
27:24	RW	0x0	sram_ret_cfg Internal memory retention mode config by software bit[0]: Internal memory 0 blank 0 ret control bit bit[1]: Internal memory 0 blank 1 ret control bit bit[2]: Internal memory 1 blank 0 ret control bit bit[3]: Internal memory 1 blank 1 ret control bit 0: Set ret to 0 1: Set ret to 1

Bit	Attr	Reset Value	Description
23:20	RW	0x0	sram_nap_cfg Internal memory nap mode config by software bit[0]: Internal memory 0 blank 0 nap control bit bit[1]: Internal memory 0 blank 1 nap control bit bit[2]: Internal memory 1 blank 0 nap control bit bit[3]: Internal memory 1 blank 1 nap control bit 0: Set nap to 0 1: Set nap to 1
19	RO	0x0	reserved
18	RW	0x0	pll_if_switch_cfg PLL low clock switch control 0: Not switch PLL source clock to low frequency clock 1: Switch PLL source clock
17	RW	0x0	pmu_if_switch_cfg PMU low clock switch control 0: Not switch clk_pmu and other module 1: Switch clk_pmu and other module
16	RO	0x1	Reserved
15	RW	0x1	rf_xo_cfg 0: Disable 1: Enable
14	RO	0x1	Reserved
13	RW	0x1	rf_resetn_cfg 0: RF resetn 1: Not resetn
12	RW	0x0	vpll_pd_cfg Software control VPLL power down 0: Not power down 1: Power down
11	RW	0x0	gpll_pd_cfg Software control GPLL power down 0: Not power down 1: Power down
10	RW	0x0	pd_mcu_pwrdown_cfg Software control mcu power down power down 0: Not power down 1: Power down
9	RW	0x0	pd_peri_pwrdown_cfg Software control peri power down power down 0: Not power down 1: Power down
8	RW	0x0	osc_40_gate_cfg 0: Not gate 1: Gate
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	lf_mode_cfg Low clock switch mode: 0: Switch to 40MHz 1: Switch to 32KHz
5	RW	0x0	lf_ena_cfg 0: Disable 1: Enable
4	RW	0x0	m4_clk_src_gating_cfg 0: Not gate 1: Gate
3	RW	0x0	idle_req_top_cfg 0: NOC for top not idle 1: NOC for top idle request
2	RW	0x0	idle_req_mcu_cfg 0: NOC for pd_mcu not idle 1: NOC for pd_mcu idle request
1	RW	0x0	idle_req_peri_cfg 0: NOC for pd_peri not idle 1: NOC for pd_peri idle request
0	RW	0x0	idle_req_wlan_cfg 0: NOC for pd_wlan not idle 1: NOC for pd_wlan idle request

PMU INT CON

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	mcu_pwr_switch_int_en Pd_mcu power switch interrupt enable 0: Disable 1: Enable
6	RW	0x0	peri_pwr_switch_int_en Pd_peri power switch interrupt enable 0: Disable 1: Enable
5	RW	0x0	wlan_pwr_switch_int_en Pd_wlan power switch interrupt enable 0: Disable 1: Enable
4	RW	0x0	wakeup_timeout_int_en Timeout wakeup interrupt enable 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
3	RW	0x0	wakeup_m0_int_en M0 wakeup interrupt enable 0: Disable 1: Enable
2	RW	0x0	wakeup_m4_int_en M4 wakeup interrupt enable 0: Disable 1: Enable
1	RW	0x0	pwrmode_wakeup_int_en Power mode wakeup interrupt enable 0: Disable 1: Enable
0	RW	0x0	pmu_int_en Global interrupt enable 0: Disable 1: Enable

PMU INT ST

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x1	mcu_power_switch_status 0: Pd_mcu not power on/off 1: Pd_mcu power on/off
6	RW	0x1	peri_power_switch_status 0: Pd_peri not power on/off 1: Pd_peri power on/off
5	RW	0x1	wlan_power_switch_status 0: Pd_wlan not power on/off 1: Pd_wlan power on/off
4	RW	0x0	wakeup_timeout_status Timeout interrupt wakeup status 0: Not wakeup by interrupt timeout 1: Wakeup by interrupt timeout
3	RW	0x0	wakeup_int_m0_status M0 interrupt wakeup status 0: Not wakeup by interrupt gpio 1: Wakeup by interrupt gpio
2	RW	0x0	wakeup_m4_status M4 wakeup status 0: Not wakeup by mcu 1: Wakeup by mcu

Bit	Attr	Reset Value	Description
1	RW	0x0	pwrmode_wakeup_status Power mode wakeup status 0: Not wakeup from power mode 1: Wakeup from power mode
0	RO	0x0	reserved

PMU BUS IDLE ST

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RW	0x0	idle_top Idle status of top niu 0: Idle status of top_niu is 0 1: Idle status of top_niu is 1
18	RW	0x0	idle_mcu Idle status of pd_mcu niu 0: Idle status of pd_mcu niu is 0 1: Idle status of pd_mcu niu is 1
17	RW	0x0	idle_peri Idle status of pd_peri niu 0: Idle status of pd_peri niu is 0 1: Idle status of pd_peri niu is 1
16	RW	0x0	idle_wlan Idle status of pd_wlan niu 0: Idle status of pd_wlan niu is 0 1: Idle status of pd_wlan niu is 1
15:4	RO	0x0	reserved
3	RW	0x0	idle_ack_top Idle acknowledge status from top niu 0: Idle acknowledge status of top niu is 0 1: Idle acknowledge status of top niu is 1
2	RW	0x0	idle_ack_mcu Idle acknowledge status from pd_mcu niu 0: Idle acknowledge status of pd_mcu niu is 0 1: Idle acknowledge status of pd_mcu niu is 1
1	RW	0x0	idle_ack_peri Idle acknowledge status from pd_peri niu 0: Idle acknowledge status of pd_peri niu is 0 1: Idle acknowledge status of pd_peri niu is 1
0	RW	0x0	idle_ack_wlan Idle acknowledge status from pd_wlan niu 0: Idle acknowledge status of pd_wlan niu is 0 1: Idle acknowledge status of pd_wlan niu is 1

PMU POWER ST

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	power_state Power state of pmu FSM

PMU OSC CNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_osc_cnt PMU OSC stable counter value

PMU_PLLLOCK_CNT

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_plllock_cnt PMU PLL stable counter value

PMU_PLLRST_CNT

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	pmu_pllrst_cnt PMU PLL reset stable counter value

PMU RET CON

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x0	sram_nap_en Internal memory power nap mode config in low power mode bit[0]: Internal memory 0 blank 0 nap control bit bit[1]: Internal memory 0 blank 1 nap control bit bit[2]: Internal memory 1 blank 0 nap control bit bit[3]: Internal memory 1 blank 1 nap control bit 0: Set nap to 0 1: Set nap to 1

Bit	Attr	Reset Value	Description
7:4	RW	0x0	sram_ret_en Internal memory retention mode config in low power mode bit[0]: Internal memory 0 blank 0 ret control bit bit[1]: Internal memory 0 blank 1 ret control bit bit[2]: Internal memory 1 blank 0 ret control bit bit[3]: Internal memory 1 blank 1 ret control bit 0: Set ret to 0 1: Set ret to 1
3:0	RW	0x0	sram_psd_en Internal memory power shut down config in low power mode bit[0]: Internal memory 0 blank 0 psd control bit bit[1]: Internal memory 0 blank 1 psd control bit bit[2]: Internal memory 1 blank 0 psd control bit bit[3]: Internal memory 1 blank 1 psd control bit 0: Set psd to 0 1: Set psd to 1

PMU INFO TX CON

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	info_tx_intv_time Information transfer interval time
7	RO	0x0	reserved
6:4	RW	0x0	info_tx_mode PMU information transfer mode select 3'b0: Power state other: Nouse
3:1	RO	0x0	reserved
0	RW	0x0	info_tx_en Pmu information transfer enable 0: Disable 1: Enable

PMU SYS REG0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	system_registers0 System register0, the register only can be reset by the npor.

PMU SYS REG1

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	system_registers1 System register1, the register only can be reset by the npor.

PMU SYS REG2

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	system_registers2 System register2, the register only can be reset by the npor.

PMU SYS REG3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	system_registers3 System register3, the register only can be reset by the npor.

PMU TIMEOUT CNT

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	timeout_count Timeout wakeup counter value

6.5 Timing Diagram

6.5.1 Power switch timing

The following figure shows the power down and power up timing for all power domain.

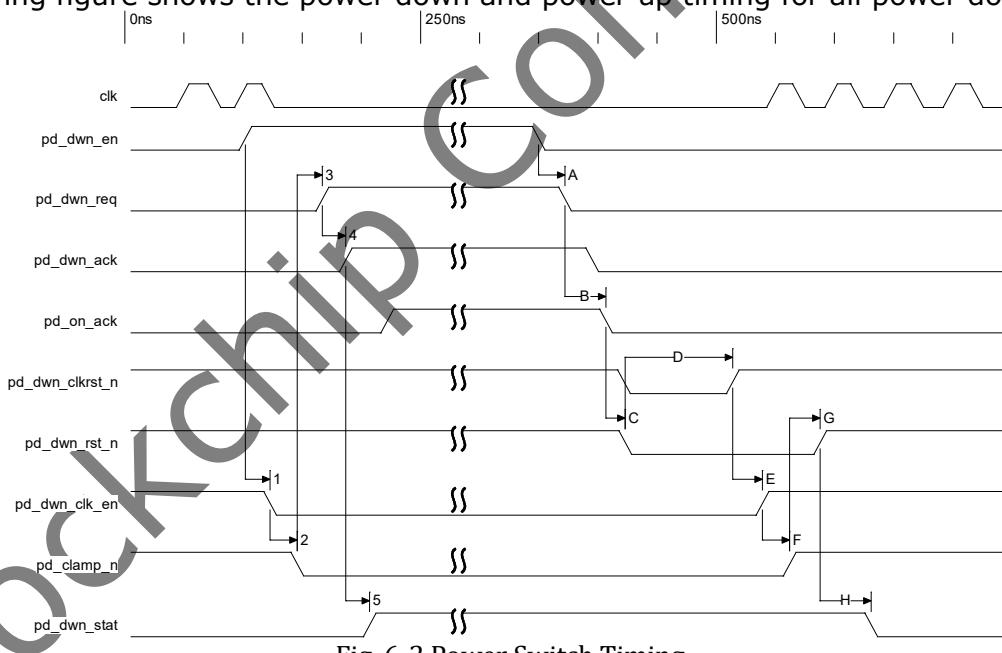


Fig. 6-2 Power Switch Timing

6.6 Application Notes

6.6.1 Low Power Mode

PMU can work in the Low power mode by setting bit0 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 6-2 Low Power State

Num	State	Description
0	NORMAL	in normal active
1	MCU_STANDBY	wait M4 into standby mode (wfi/wfe)
2	TRANS_NO_FIN	wait the corresponding noc interface into idle
3	MCU_CLK_DIS	close M4 clock source
4	SRAM_RET	system sram in retention
5	MCU_PWRDN	pd_mcu power down
6	PMU_LF	pmu clock switch to 32KHz
7	PLL_PWRDN	pll power down
8	OSC_GATE	gate osc40 into cru
9	RF_VDD18_DIS	disable RF VDD18 supply
10	OSC_DIS	close the osc
11	WAIT_WAKEUP	wait wakeup source for M0
12	OSC_EN	enable the OSC
13	RF_VDD18_EN	enable RF VDD18 supply
14	RF_RESET	Assert RF Resetn
15	AON_RESET	N/A
16	RF_RESET_CLR	De-assert RF Resetn
17	AON_RESET_CLR	N/A
18	OSC_STABLE	Wait OSC stable >=2ms
19	OSC_PASS	resume the osc40 out from cru
20	PLL_PWRUP	pll power up
21	PMU_HF	switch pd_pmu back to high speed clock
22	SRAM_UP	system sram exit retention
23	MCU_PWRUP	pd_mcu power up
24	MCU_CLK_EN	resume the mcu clock source
25	TRANS_RESTORE	resume the noc interface

6.6.2 Debug IO

RK2206 provide PMU Debug IO for FSM observation. Each IO maps to the correspond bit of the PMU_POWER_ST register.

Table 6-3 Debug IO MUX

Module Pin	Direction	Pin Name	IOMUX Setting
power_state[0]	O	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_DRIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPJTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_SEL_L[11:8] = 4'b0110
power_state[1]	O	LCD_CSn/CIF_VSYNC/I2C1_SCL_M2/TKEY15/PMU_DEBUG/PMU_STATE1/AONJTAG_TDO/DSPJTAG_TDO/GPIO0_A3_u	GRF_GPIO0A_IOMUX_SEL_L[15:12] = 4'b0110
power_state[2]	O	LCD_RDN/CIF_CLKOUT/UART1_CTSN_M1/TKEY16/PMU_SL_EEP/PMU_STATE2/CODECCLK_M1/AONJTAG_TRSTn/DSPJTAG_TRSTn/GPIO0_A4_u	GRF_GPIO0A_IOMUX_SEL_H[3:0] = 4'b0110

Module Pin	Direction	Pin Name	IOMUX Setting
power_state [3]	O	LCD_WRN/CIF_CLKIN/UART1_RTSN_M1/PDM_CLK_M0/TK_EY17/PMU_STATE3/CODEC_SYNC_M1/GPIO0_A5_d	GRF_GPIO0A_IOMUX_SEL_H[7:4] = 4'b0110
power_state [4]	O	LCD_D2/CIF_D2/UART1_RX_M1/PDM_SDI_M0/TKEY18/PMU_STATE4/CODEC_ADC_D_M1/GPIO0_A6_d	GRF_GPIO0A_IOMUX_SEL_H[11:8] = 4'b0110
debug_Sout	O	LCD_CSn/CIF_VSYNC/I2C1_SCL_M2/TKEY15/PMU_DEBUG/PMU_STATE1/AONJTAG_TDO/DSPJTAG_TDO/GPIO0_A3_u	GRF_GPIO0A_IOMUX_SEL_L[15:12] = 4'b0101

Another way, RK2206 provided the serial output by the debug_sout for FSM observation by setting the bit0 of PMU_INFO_TX_CON register. Once the bit was set to "1", pmu will sent the serial signal as the following timing diagram.

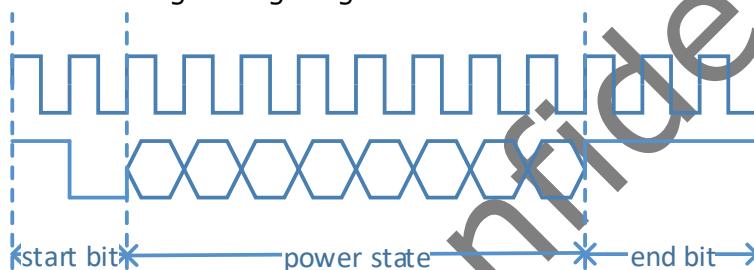


Fig. 6-3 PMU Info Timing

Start_bit: always is "10";

Power_state: shown the current power state of pmu FSM

End_bit: pmu_info_tx_intv_time + 2

6.6.3 Clock Switch

When the system work in the low power mode, some modules (including GPIO, GRF, 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 (40MHz or 32KHz) by the PMU FSM.

There are two bits for switching to 40MHz or 32KHz in the PMU_PWRMODE_CON register.

If_ena_cfg: bit5, clock switch to low frequency enable;

If_mode_cfg: bit6, clock switch to 40MHz or 32KHz, 0-40Mhz, 1-32KHz;

There are another two bits for which switching to the low frequency clock:

pmu_if_switch_cfg : the module clock switch to the low frequency clock

pll_if_switch_cfg : the pll switch to the low frequency clock

There is only one bit can been set in the low power mode.

Note: the high speed frequency of clk_pmu is 40MHz, so it only can be switched to 32KHz.

6.6.4 Power Mode Control Registers and Soft Control Registers

There are two types registers: PMU_PWRMODE_CON_x and PMU_SFT_CON_x to control the same function, for example pll on/off. The PMU_PWRMODE_CON_X register is valid only in the pmu low power mode, and the PMU_SFT_CON_x register will take effect as soon as the register is set.

The following figure shows the structure of the two types registers:

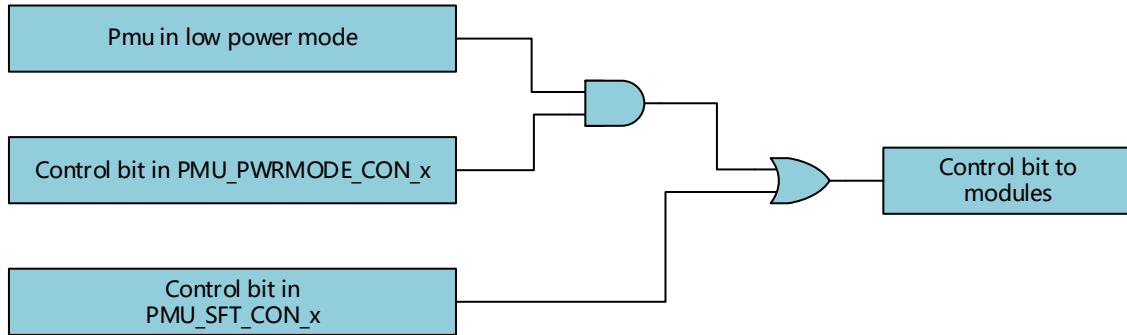


Fig. 6-4 PMU Control Register Relationship

Rockchip Confidential

Chapter 7 I2S

7.1 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.

7.1.1 Features

There are 2 I2S/PCM controllers embed in the system, one is I2S0, the other is I2S1. The I2S/PCM controller supports I2S and PCM mode stereo audio output and input.

- Support eight internal 32-bit wide and 32-location deep FIFOs, four for transmitting and the others 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 4-channel audio transmitting in I2S mode and 2-channel in PCM mode for I2S0
- Support 2-channel audio receiving in I2S mode and 2 channel in PCM mode for I2S0
- Support 2-channel audio transmitting in I2S mode and 2-channel in PCM mode for I2S1
- Support 4-channel audio receiving in I2S mode and 2 channel in PCM mode for I2S1
- 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. Single LRCK can be used for transmitting and receiving data if the sample rate are the same
- Support configurable SCLK and LRCK polarity

7.2 Block Diagram

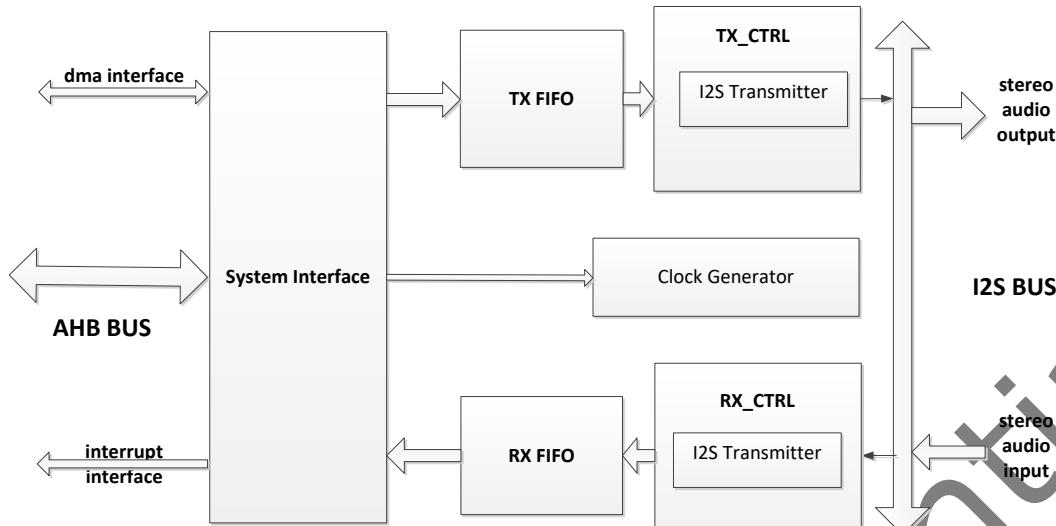


Fig.7-1 I2S/PCM controller (8 channel) Block Diagram

System Interface

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

Clock Generator

The Clock Generator implements clock generation function. By the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either a master or a slave, with I2S or PCM mode surround 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 one FIFO is 32bits x 32.

Receive FIFO

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

7.3 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.

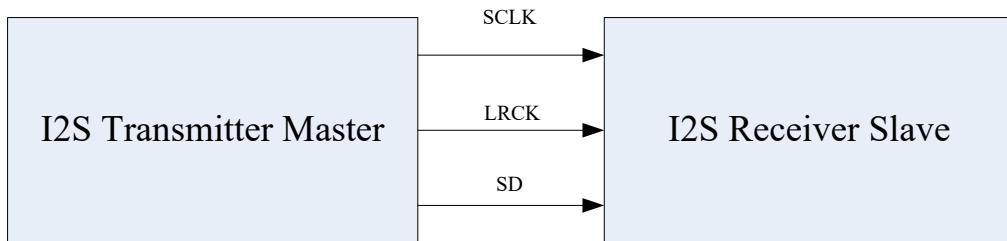


Fig.7-2 I2S transmitter-master & receiver-slave condition

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.

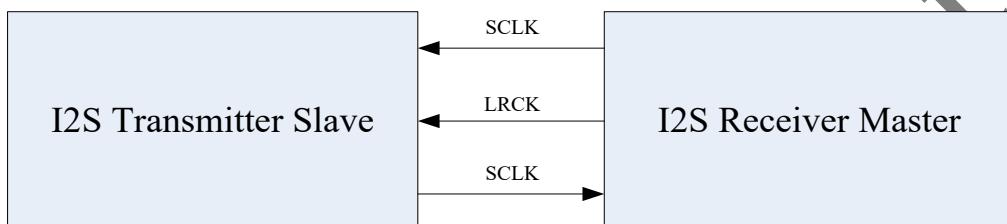


Fig.7-3 I2S transmitter-slave & receiver-master condition

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.

7.3.1 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.

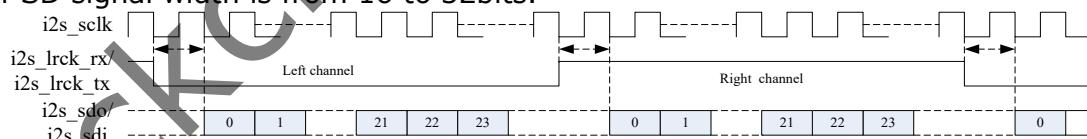


Fig.7-4 I2S normal mode timing format

7.3.2 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.

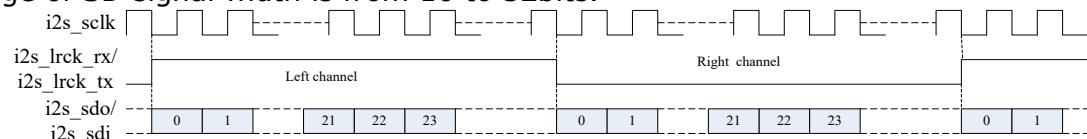


Fig.7-5 I2S left justified mode timing format

7.3.3 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.

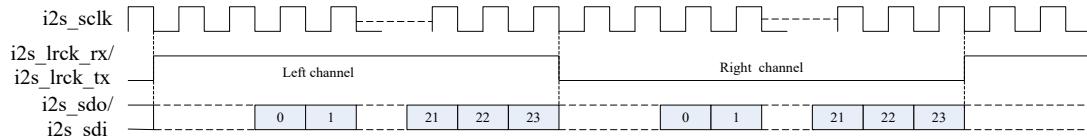


Fig.7-6 I2S right justified mode timing format

7.3.4 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.

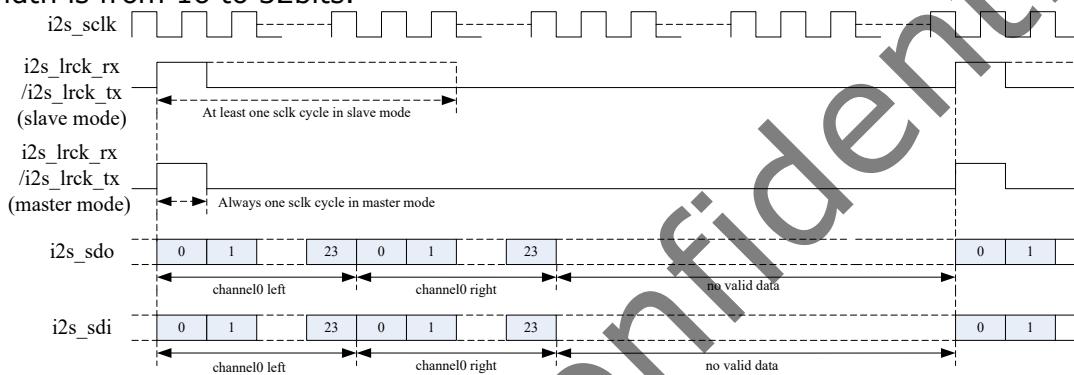


Fig.7-7 PCM early mode timing format

7.3.5 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.

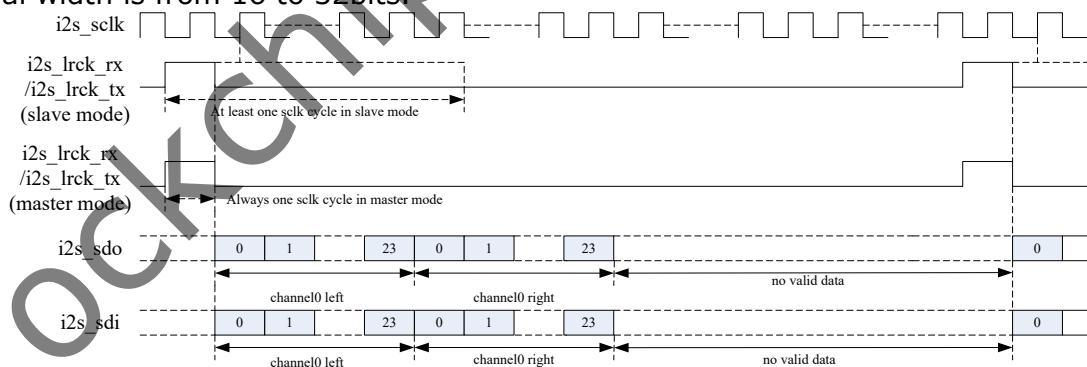


Fig.7-8 PCM late1 mode timing format

7.3.6 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.

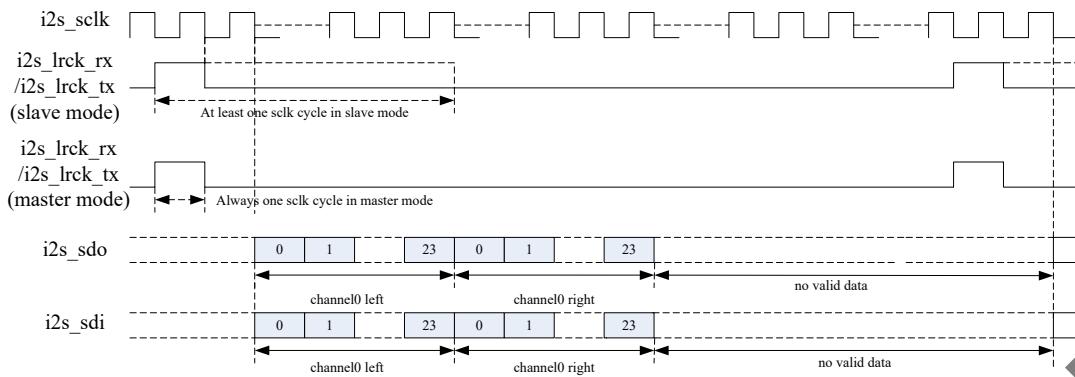


Fig.7-9 PCM late2 mode timing format

7.3.7 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.

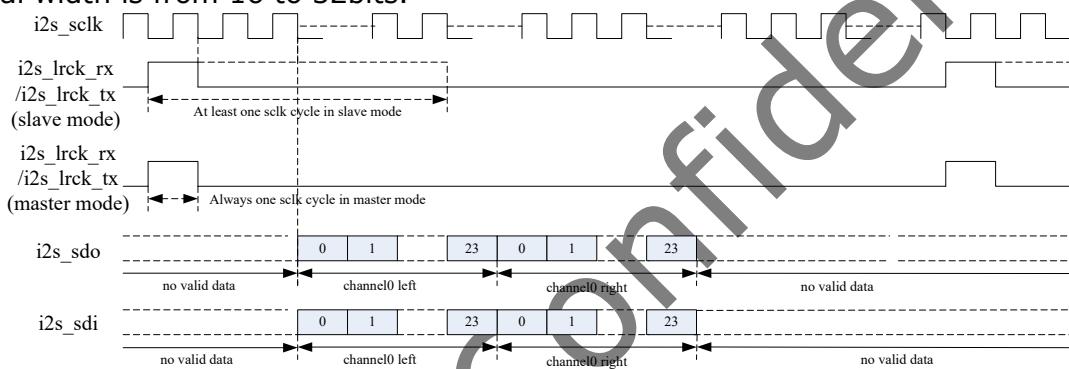


Fig.7-10 PCM late3 mode timing format

7.4 Register description

7.4.1 Internal Address Mapping

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

7.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
I2S_TXCR	0x0000	W	0x00000000	Transmit Operation Control Register
I2S_RXCR	0x0004	W	0x00480000	Receive Operation Control Register
I2S_CKR	0x0008	W	0x00000000	Clock Generation Register
I2S_TXFIFOLR	0x000c	W	0x00000000	TX FIFO Level Register
I2S_DMACR	0x0010	W	0x00000000	DMA Control Register
I2S_INTCR	0x0014	W	0x01f00000	Interrupt Control Register
I2S_INTSR	0x0018	W	0x00000000	Interrupt Status Register
I2S_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S_CLR	0x0020	W	0x00000000	Sclk Domain Logic Clear Register
I2S_TXDR	0x0024	W	0x00000000	Transmit FIFO Data Register
I2S_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register

Name	Offset	Size	Reset Value	Description
I2S_RXFIFOLR	0x002c	W	0x00000000	RX FIFO Level Register
I2S_CLKDIV	0x0038	W	0x00000000	Clock Divider Register
I2S_VERSION	0x003c	W	0x00000000	Version Register

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

7.4.2 Detail Register Description

I2S TXCR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:29	RW	0x0	tx_path_select3 TX Path Select 2'b00: Sd03 output data from path0 2'b01: Sd03 output data from path1 2'b10: Sd03 output data from path2 2'b11: Sd03 output data from path3
28:27	RW	0x0	tx_path_select2 TX Path Select 2'b00: Sd02 output data from path0 2'b01: Sd02 output data from path1 2'b10: Sd02 output data from path2 2'b11: Sd02 output data from path3
26:25	RW	0x0	tx_path_select1 TX Path Select 2'b00: Sd01 output data from path0 2'b01: Sd01 output data from path1 2'b10: Sd01 output data from path2 2'b11: Sd01 output data from path3
24:23	RW	0x0	tx_path_select0 TX Path Select 2'b00: Sd00 output data from path0 2'b01: Sd00 output data from path1 2'b10: Sd00 output data from path2 2'b11: Sd00 output data from path3
22:17	RW	0x00	RCNT (Can be written only when XFER[0] bit is 0.) Only valid in I2S Right justified format and slave TX mode is selected. Start to transmit data RCNT sclk cycles after left channel valid. Note: Only function when TX TFS[1]=0.
16:15	RW	0x0	TCSR Transmit Channel Select Register 2'b00: Two channel 2'b01: Four channel 2'b10: Six channel 2'b11: Eight channel

Bit	Attr	Reset Value	Description
14	RW	0x0	HWT Halfword Word Transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid from AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store Justified Mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width FIFO. If VDW select 16bit data, this bit is valid only when HWT select 0. Because if HWT is 1, every FIFO unit contains two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 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 (Can be written only when XFER[0] bit is 0.) 2'b00: PCM no delay mode 2'b01: PCM delay 1 mode 2'b10: PCM delay 2 mode 2'b11: PCM delay 3 mode Note: Function when TX TFS[1:0] is 1.
6:5	RW	0x0	TFS (Can be written only when XFER[0] bit is 0.) 2'b00: I2S format 2'b01: PCM format 2'b10~2'b11: Reserved

Bit	Attr	Reset Value	Description
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
31:25	RO	0x0	reserved
24:23	RW	0x0	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
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
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
18:17	RW	0x0	rx_path_select0 Rx Path Select 2'b00: Path0 data from sdi0 2'b01: Path0 data from sdi1 2'b10: Path0 data from sdi2 2'b11: Path0 data from sdi3

Bit	Attr	Reset Value	Description
16:15	RW	0x0	RCSR Receive Channel Select Register 2'b00: Two channel 2'b01: Four channel 2'b10: Six channel 2'b11: Eight channel
14	RW	0x0	HWT Halfword Word Transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 1'b0: 32 bit data valid to AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel. 1'b1: Low 16bit data valid to AHB/APB bus, high 16 bit data invalid.
13	RO	0x0	reserved
12	RW	0x0	SJM Store Justified Mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width FIFO. If VDW select 16bit data, this bit is valid only when HWT select 0. Because if HWT is 1, every FIFO unit contains two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 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

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'b10~2'b11: Reserved
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	LRCK_COMMON TX_and_RX Common Use 2'b00/2'b11: Tx_Irck/rx_Irck are used as synchronous signal for TX /RX respectively. 2'b01: Only tx_Irck is used as synchronous signal for TX and RX. 2'b10: Only rx_Irck is used as synchronous signal for TX and RX.
27	RW	0x0	MSS Master/Slave Mode Select (Can be written only when XFER[1] or XFER[0] bit is 0.) 1'b0: Master mode(sclk output) 1'b1: Slave mode(sclk input)
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	<p>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)</p>
24	RW	0x0	<p>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)</p>
23:16	RO	0x0	reserved
15:8	RW	0x00	<p>RSD Receive Sclk Divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (receive sclk divider/2)*2*frequency of rx_lrck Note: Function when RX TFS[1:0] is 2'b00 or 2'b01.</p>
7:0	RW	0x00	<p>TSD Transmit Sclk Divider (Can be written only when XFER[1] or XFER[0] bit is 0.) 8'h00~8'h1e: Reserved 8'h1f~8'hff: Frequency of sclk = (Transmit sclk divider/2)*2*frequency of tx_lrck Note: Function when TX TFS[1:0] is 2'b00 or 2'b01.</p>

I2S 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 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	0x00	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
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 TX FIFO(TX FIFO0 if CSR=00;TX FIFO1 if CSR=01,TX FIFO2 if CSR=10,TX FIFO3 if CSR=11)is equal to or below this field value.

I2S_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 entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	RW	0x0	TXUIC TX Underrun Interrupt Clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX Underrun Interrupt Enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	TXEIE TX empty Interrupt Enable 1'b0: Disable 1'b1: Enable

I2S_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_XFER

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Start Bit 1'b0: Stop RX transfer 1'b1: Start RX transfer
0	RW	0x0	TXS TX Transfer Start Bit 1'b0: Stop TX transfer 1'b1: Start TX transfer

I2S_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, 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.

I2S_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_CLKDIV

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	RX_MDIV RX Mclk Divider (Can be written only when XFER[1] bit is 0.) mclk_rx divider = (mclk_rx/sclk_rx)-1. For example, if mclk_rx divider is 5, then the frequency of sclk_rx is mclk_rx/6.

Bit	Attr	Reset Value	Description
7:0	RW	0x00	TX_MDIV TX Mclk Divider (Can be written only when XFER[0] bit is 0.) mclk_tx divider = (mclk_tx/sclk_tx)-1. For example, if mclk_tx divider is 5, then the frequency of sclk_tx is mclk_tx/6.

I2S VERSION

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	I2S_VERSION I2S Version

7.5 Interface Description

Two I2S controllers embed in the chip are I2S0 and I2S1. The I2S0 is connected to two groups of IO interfaces while the I2S1 communicates with the Digital Audio Codec inside the chip.

The following table shows the I2S0 group 0 interface description.

Table 7-1 I2S0 Group 0 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_mclk_tx or i2s0_mclk_rx	I/O	CODEC_ADC_D_M0/PWM6_M1/ TKEY2_M1/UART2_CTSN_M0/ SPI1_CLK_M2/I2S_MCLK_M1/GPIO0_D2_u	GRF_GPIO0D_IOMUX_L[11:8]=4'b0110
i2s0_sclk_tx or i2s0_sclk_rx	I/O	CODEC_DAC_DL_M0/PWM8/TKEY3_M1/UAR T2_RTSN_M0/SPI1_MOSI_M2/I2S_SCLK_TX _M1/GPIO0_D3_u	GRF_GPIO0D_IOMUX_L[15:12]=4'b0110
i2s0_lrck_tx or i2s0_lrck_rx	I/O	PMIC_INT_M0/PWM9/TKEY4_M1/UART2_RX _M0/SPI1_MISO_M2/I2S_LRCK_TX_M1/GPI O0_D4_u	GRF_GPIO0D_IOMUX_H[3:0]=4'b0110
i2s0_sdi0	I	I2C2_SCL_M0/PWM11/TKEY6_M1/TKEY_DRI VE_M5/I2C1_SCL_M3/I2S_SDI_M1/PWM_A UDIO_R_M3/GPIO0_D6_u	GRF_GPIO0D_IOMUX_H[11:8]=4'b0110
i2s0_sdo0	O	I2C2_SDA_M0/PWM10/TKEY5_M1/UART2_T X_M0/I2C1_SDA_M3/I2S_SDO0_M1/PWM_A UDIO_L_M3/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[7:4]=4'b0110

Whether the i2s0_mclk_tx or i2s0_mclk_rx is connected to I2S_MCLK_M1 depends on the value of CRU_CLKSEL_CON26[15:14]. If CRU_CLKSEL_CON26[15:14] is equal to 2'b00, then i2s0_mclk_tx is connected to I2S_MCLK_M1, otherwise the i2s0_mclk_rx is connected to I2S_MCLK_M1.

The following table shows the I2S0 group 1 interface description.

Table 7-2 I2S0 Group 1 Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_mclk_tx or i2s0_mclk_rx	I/O	SRADC3/PWM3_M0/UART1_TX_M0/SPI0_MISO_M1 /PDM_CLK_S_M1/I2S_MCLK_M0/TKEY4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GRF_GPIO0C_IOMUX_L[15:12]=4'b0110
i2s0_sclk_tx	I/O	SRADC4/PWM4_M0/UART0_CTSN_M1/SPI1_CS0n_M0/PDM_CLK_M1/I2S_SCLK_TX_M0/TKEY5_M0/GPIO0_C4_u	GRF_GPIO0C_IOMUX_H[3:0]=4'b0110

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_sclk_rx	I/O	SRADC1/PWM1_M0/UART1_RTSN_M0/SPI0_CLK_M1/I2C1_SDA_M1/I2S_SCLK_RX_M0/SDMMC_CMD/TKEY10/GPIO0_C1_u	GRF_GPIO0C_IOMUX_L[7:4] =4'b0110
i2s0_lrck_tx	I/O	SRADC5/PWM5_M0/UART0_RTSN_M1/SPI1_CLK_M0/PDM_SD _I M1/I2S_LRCK_TX_M0/TKEY6_M0/GPO0_C5_u	GRF_GPIO0C_IOMUX_H[7:4] =4'b0110
i2s0_lrck_rx	I/O	SRADC2/PWM2_M0/UART1_RX_M0/SPI0_MOSI_M1/I2C1_SCL_M1/I2S_LRCK_RX_M0/SDMMC_D0/TKEY11/GPIO0_C2_u	GRF_GPIO0C_IOMUX_L[11:8] =4'b0110
i2s0_sdi0	I	SRADC7/PWM7_M0/UART0_TX_M1/SPI1_MISO_M0/I2C0_SCL_M1/I2S_SDI_M0/TKEY8_PWM_AUDIO_R_M1/PMIC_INT_M1/GPIO0_C7_d	GRF_GPIO0C_IOMUX_H[15:12] =4'b0110
i2s0_sdo0	O	SRADC6/PWM6_M0/UART0_RX_M1/SPI1_MOSI_M0/I2C0_SDA_M1/I2S_SDO_M0/TKEY7_PWM_AUDIO_L_M1/GPIO0_C6_u	GRF_GPIO0C_IOMUX_H[11:8] =4'b0110
i2s0_sdo1	O	SRADC0/PWM0_M0/UART1_CTSN_M0/SPI0_CS0n_M1/I2S_SDO1_M0/SDMMC_CLKOUT/TKEY9/GPIO0_C0_u	GRF_GPIO0C_IOMUX_L[3:0] =4'b0101

GRF_SOC_CON16[4] determines which group of input signals are connected to the I2S0. If GRF_SOC_CON16[4] is equal to b0, the input signals of I2S0 group 0 interface are connected to the I2S0, otherwise the input signals of I2S0 group 1 interface are connected to the I2S0.

7.6 Application Notes

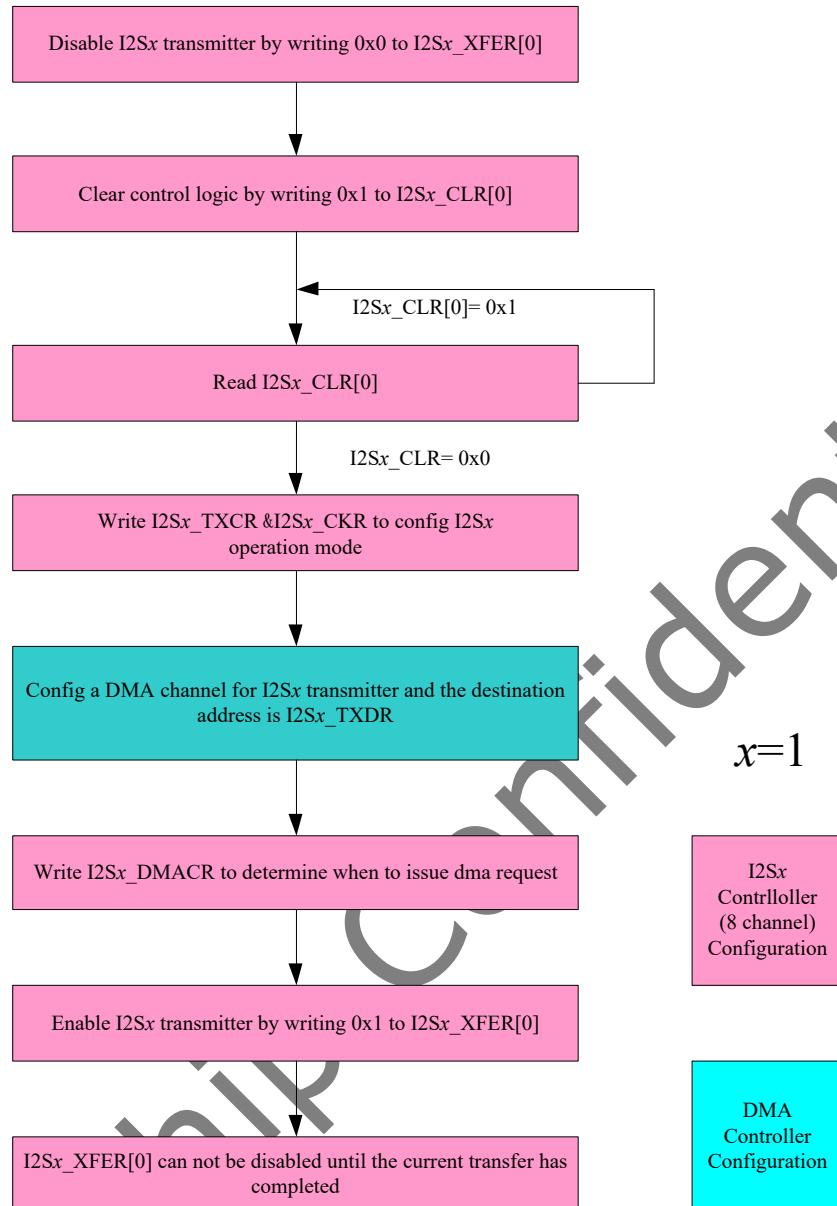


Fig.7-11 I2S/PCM controller transmit operation flow chart

Note: User should clear TX/RX logical by CLR[0]/CLR[1] and wait clear operation done before configure the other registers.

Chapter 8 I2C Interface

8.1 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:

- Supports 4 independent I2C: I2C0, I2C1, I2C2
- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- Filter out glitch on SCL and SDA

8.2 Block Diagram

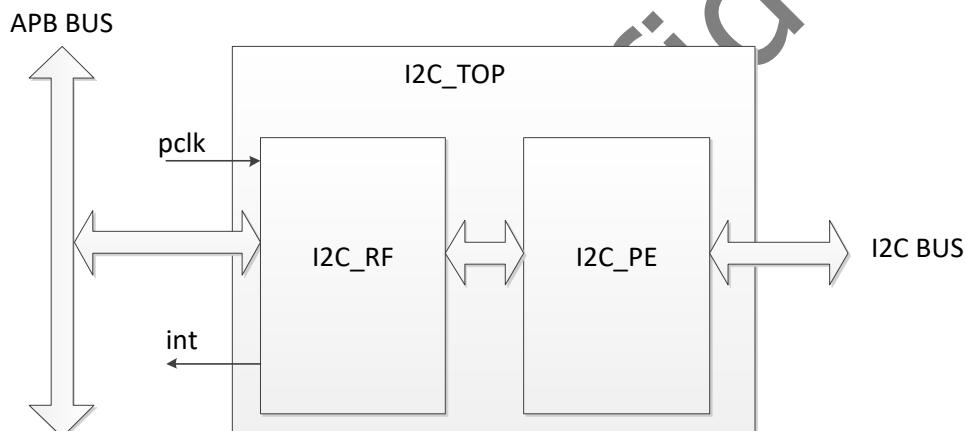


Fig. 8-1 I2C architecture

8.2.1 I2C_RF

I2C_RF module is used to control the I2C controller operation by the host with APB interface. It implements the register settings and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

8.2.2 I2C_PE

I2C_PE module implements the I2C master operation for transmitting/receiving data from other I2C devices. The I2C master controller operates synchronously with the pclk.

8.2.3 I2C_TOP

I2C_TOP module is the top module of the I2C controller.

8.3 Function Description

This chapter provides a description about the functions and behavior under various conditions.

The I2C controller only supports master function. It supports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller are divided to 2 parts and described separately: initialization and master mode programming.

8.3.1 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 to 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.

8.3.2 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:

$$\text{SCL Divisor} = 8 * (\text{CLKDIVL} + 1 + \text{CLKDIVH} + 1)$$

$$\text{SCL} = \text{PCLK} / \text{SCLK Divisor}$$

- 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 transmits 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 enters 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 enters 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.

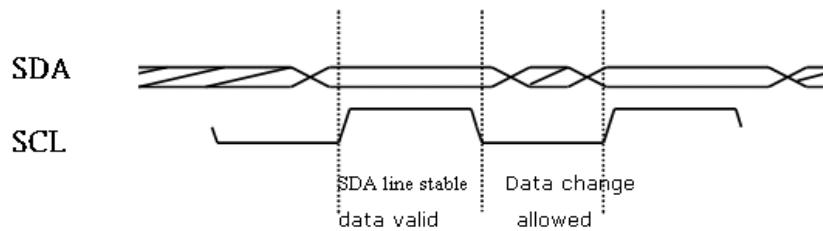


Fig. 8-2 I2C DATA Validity

◆ 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.

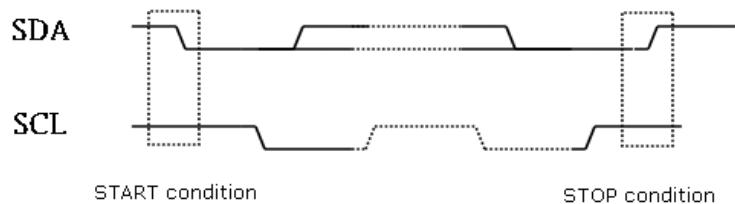


Fig. 8-3 I2C Start and stop conditions

◆ 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".

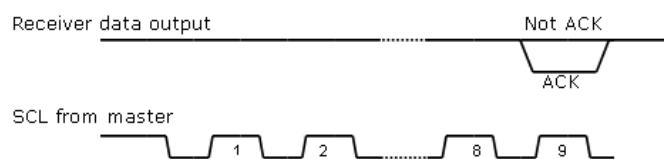


Fig. 8-4 I2C Acknowledge

➤ Byte transfer

The master owned 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.

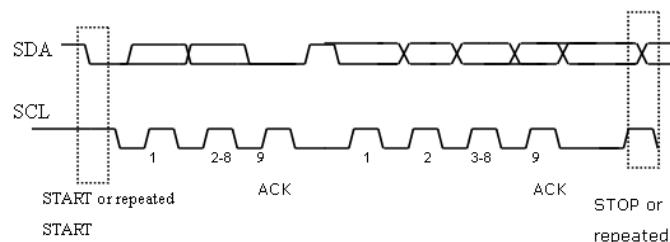


Fig. 8-5 I2C byte transfer

8.4 Register Description

8.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
RKI2C_CON	0x0000	W	0x00030000	Control register
RKI2C_CLKDIV	0x0004	W	0x00000001	Clock divider register, I2C CLK = PCLK / (16*CLKDIV)
RKI2C_MRXADDR	0x0008	W	0x00000000	The slave address accessed for master RX mode
RKI2C_MRXRADDR	0x000c	W	0x00000000	The slave register address accessed for master RX mode
RKI2C_MTXCNT	0x0010	W	0x00000000	Master transmit count. Specify the total bytes to be transmit (0~32)
RKI2C_MRXCNT	0x0014	W	0x00000000	Master RX count. Specify the total bytes to be received(0~32)
RKI2C_IEN	0x0018	W	0x00000000	Interrupt enable register
RKI2C_IPD	0x001c	W	0x00000000	Interrupt pending register
RKI2C_FCNT	0x0020	W	0x00000000	Finished count: the count of data which has been transmitted or received for debug purpose
RKI2C_SCL_OE_DB	0x0024	W	0x00000020	Slave hold debounce configure register
RKI2C_TXDATA0	0x0100	W	0x00000000	I2C TX data register 0
RKI2C_TXDATA1	0x0104	W	0x00000000	I2C TX data register 1
RKI2C_TXDATA2	0x0108	W	0x00000000	I2C TX data register 2
RKI2C_TXDATA3	0x010c	W	0x00000000	I2C TX data register 3
RKI2C_TXDATA4	0x0110	W	0x00000000	I2C TX data register 4
RKI2C_TXDATA5	0x0114	W	0x00000000	I2C TX data register 5
RKI2C_TXDATA6	0x0118	W	0x00000000	I2C TX data register 6
RKI2C_TXDATA7	0x011c	W	0x00000000	I2C TX data register 7
RKI2C_RXDATA0	0x0200	W	0x00000000	I2C RX data register 0
RKI2C_RXDATA1	0x0204	W	0x00000000	I2C RX data register 1
RKI2C_RXDATA2	0x0208	W	0x00000000	I2C RX data register 2
RKI2C_RXDATA3	0x020c	W	0x00000000	I2C RX data register 3
RKI2C_RXDATA4	0x0210	W	0x00000000	I2C RX data register 4
RKI2C_RXDATA5	0x0214	W	0x00000000	I2C RX data register 5
RKI2C_RXDATA6	0x0218	W	0x00000000	I2C RX data register 6
RKI2C_RXDATA7	0x021c	W	0x00000000	I2C RX data register 7
RKI2C_ST	0x0220	W	0x00000000	Status debug register
RKI2C_DBGCTRL	0x0224	W	0x00000000	Debug config register

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

8.4.2 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: $TSU;sto = (\text{stop_setup} + 1) * T(\text{SCL_HIGH}) + \text{Tclk_i2c}$
13:12	RW	0x0	start_setup Start setup config: $TSU;sta = (\text{start_setup} + 1) * T(\text{SCL_HIGH}) + \text{Tclk_i2c}$ $THD;sta = (\text{start_setup} + 2) * T(\text{SCL_HIGH}) - \text{Tclk_i2c}$
11	RO	0x0	reserved
10:8	RW	0x0	data_upd_st SDA update point config: Used to config sda change state when scl is low, used to adjust setup/hold time 4'b n : $\text{Thold} = (n + 1) * \text{Tclk_i2c}$ Note: $0 \leq n \leq 5$
7	RO	0x0	reserved
6	RW	0x0	act2nak Operation when NAK handshake is received: 1'b0: Ignored 1'b1: Stop transaction
5	RW	0x0	ack Last byte acknowledge control in master receive mode: 1'b0: ACK 1'b1: NAK
4	RW	0x0	stop Stop enable, when this bit is written to 1, I2C will generate stop signal.
3	RW	0x0	start Start enable, when this bit is written to 1, I2C will generate start signal.
2:1	RW	0x0	i2c_mode I2c mode select: 2'b00: transmit only 2'b01: transmit address (device + register address) --> restart --> transmit address -> receive only 2'b10: receive only 2'b11: transmit address (device + register address, write/read bit is 1) --> restart --> transmit address (device address) --> receive data
0	RW	0x0	i2c_en i2c module enable: 1'b0: Disable 1'b1: Enable

RKI2C_CLKDIV

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH Scl high level clock count: $T(SCL_HIGH) = Tclk_i2c * (CLKDIVH + 1) * 8$
15:0	RW	0x0001	CLKDIVL Scl low level clock count: $T(SCL_LOW) = Tclk_i2c * (CLKDIVL + 1) * 8$

RKI2C_MRADDR

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	0x0000000	saddr Master address register. the lowest bit indicate write or read 24 bits address register

RKI2C_MRRAADDR

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

Bit	Attr	Reset Value	Description
24	RW	0x0	sraddlvlid Address low byte valid: 1'b0: Invalid 1'b1: Valid
23:0	RW	0x000000	saddr 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

Bit	Attr	Reset Value	Description
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	W1 C	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	W1 C	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	W1 C	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	W1 C	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

Bit	Attr	Reset Value	Description
2	W1 C	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	W1 C	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	W1 C	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
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 1'b0: Check if scl been pull down by slave at the whole SCL_HIGH. 1'b1: Check if scl been pull down by slave only at the h0 of SCL_HIGH(SCL_HIGH including h0~h7).
13	RW	0x0	nak_release_scl 1'b0: Hold scl as low when received nack 1'b1: Release scl as high when received nack
12	RW	0x0	flt_en SCL edage glitch filter enable 1'b0: Disable 1'b1: Enable
11:8	RW	0x0	slv_hold_scl_th Slave hold scl threshold = slv_hold_scl_db * Tclk_i2c
7:4	RW	0x0	flt_r Filter scl rising edge glitches of width less than flt_r * Tclk_i2c
3:0	RW	0x0	flt_f Filter scl falling edge glitches of width less than flt_f * Tclk_i2c

8.5 Interface Description

Table 8-1 I2C Interface Description

Module pin	Direction	Pin name	IOMUX
IOMUX0 I2C0 Interface			
I2C_SDA0	I/O	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_O_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GRF_GPIO0B_IOMUX_H[3:0] = 4'b0100
I2C_SCL0	I/O	PWM1_M1/UART0_RTSN_M0/SPI0_CLK_M0/I2C0_SCL_M0/TKEY1_M0/PWM_AUDIO_R_M0/I2C2_SCL_M1/GPIO0_B5_d	GRF_GPIO0B_IOMUX_H[6:4] = 3'b100
IOMUX0 I2C1 Interface			
I2C_SDA1	I/O	PWM2_M1/UART0_RX_M0/SPI0_MOSI_M0/I2C1_SDA_M0/TKEY2_M0/GPIO0_B6_d	GRF_GPIO0B_IOMUX_H[10:8] = 3'b100
I2C_SCL1	I/O	PWM3_M1/UART0_TX_M0/SPI0_MISO_M0/I2C1_SCL_M0/TKEY3_M0/TKEY_DRIVE_M2/GPIO0_B7_d	GRF_GPIO0B_IOMUX_H[14:12] = 3'b100
IOMUX0 I2C2 Interface			
I2C_SDA2	I/O	I2C2_SDA_M0/PWM10/TKEY5_M1/UART2_TX_M0/I2C1_SDA_M3/I2S_SDO0_M1/PWM_AUDIO_L_M3/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[6:4] = 3'b001
I2C_SCL2	I/O	I2C2_SCL_M0/PWM11/TKEY6_M1/TKEY_DRIVE_M5/I2C1_SCL_M3/I2S_SDI_M1/PWM_AUDIO_R_M3/GPIO0_D6_u	GRF_GPIO0D_IOMUX_H[10:8] = 3'b001
IOMUX1 I2C0 Interface			
I2C_SDA0	I/O	SRADC6/PWM6_M0/UART0_RX_M1/SPI1_MOSI_M0/I2C0_SDA_M1/I2S_SDO_M0/TKEY7/PWM_AUDIO_L_M1/GPIO0_C6_u	GRF_GPIO0C_IOMUX_H[11:8] = 4'b0101
I2C_SCL0	I/O	SRADC7/PWM7_M0/UART0_TX_M1/SPI1_MISO_M0/I2C0_SCL_M1/I2S_SDI_M0/TKEY8/PWM_AUDIO_R_M1/PMIC_INT_M1/GPIO0_C7_d	GRF_GPIO0C_IOMUX_H[15:12] = 4'b0101
IOMUX1 I2C1 Interface			
I2C_SDA1	I/O	SRADC1/PWM1_M0/UART1_RTSN_M0/SPI0_CLK_M1/I2C1_SDA_M1/I2S_SCLK_RX_M0/SDMMC_CMD/TKEY10/GPIO0_C1_u	GRF_GPIO0C_IOMUX_L[7:4] = 4'b0101
I2C_SCL1	I/O	SRADC2/PWM2_M0/UART1_RX_M0/SPI0_MOSI_M1/I2C1_SCL_M1/I2S_LRCK_RX_M0/SDMMC_D0/TKEY11/GPIO0_C2_u	GRF_GPIO0C_IOMUX_L[11:8] = 4'b0101
IOMUX1 I2C2 Interface			
I2C_SDA2	I/O	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_O_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GRF_GPIO0B_IOMUX_H[3:0] = 4'b1000
I2C_SCL2	I/O	PWM1_M1/UART0_RTSN_M0/SPI0_CLK_M0/I2C0_SCL_M0/TKEY1_M0/PWM_AUDIO_R_M0/I2C2_SCL_M1/GPIO0_B5_d	GRF_GPIO0B_IOMUX_H[6:4] = 3'b111

Module pin	Direction	Pin name	IOMUX
IOMUX2 I2C0 Interface			
I2C_SDA0	I/O	LCD_D0/CIF_D0/I2C0_SDA_M2/TKEY12/M4F_WFI/M4F_JTAG_TCK/M0_JTAG_TCK/AONJTAG_TCK/DSP_JTAG_TCK/GPIO0_A0_u	GRF_GPIO0A_IOMUX_L[3:0] = 4'b0011
I2C_SCL0	I/O	LCD_D1/CIF_D1/I2C0_SCL_M2/TKEY13/M0_WFI/M4F_JTAG_TMS/M0_JTAG_TMS/AONJTAG_TMS/DSPJ_TAG_TMS/GPIO0_A1_u	GRF_GPIO0A_IOMUX_L[7:4] = 4'b0011
IOMUX2 I2C1 Interface			
I2C_SDA1	I/O	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_D_RIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPJTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_L[11:8] = 4'b0011
I2C_SCL1	I/O	LCD_CSn/CIF_VSYNC/I2C1_SCL_M2/TKEY15/PMU_DEBUG/PMU_STATE1/AONJTAG_TDO/DSPJTAG_TD_O/GPIO0_A3_u	GRF_GPIO0A_IOMUX_L[15:12] = 4'b0011
IOMUX3 I2C0 Interface			
I2C_SDA0	I/O	CODEC_CLK_M0/PWM4_M1/TKEY0_M1/UART1_RX_M2/TKEY_DRIVE_M4/I2C0_SDA_M3/PWM_AUDIO_L_M2/GPIO0_D0_u	GRF_GPIO0D_IOMUX_L[2:0] = 3'b110
I2C_SCL0	I/O	CODEC_SYNC_M0/PWM5_M1/TKEY1_M1/UART1_TX_M2/SPI1_CS0n_M2/I2C0_SCL_M3/PWM_AUDIO_R_M2/GPIO0_D1_u	GRF_GPIO0D_IOMUX_L[6:4] = 3'b110
IOMUX3 I2C1 Interface			
I2C_SDA1	I/O	I2C2_SDA_M0/PWM10/TKEY5_M1/UART2_TX_M0/I2C1_SDA_M3/I2S_SD00_M1/PWM_AUDIO_L_M3/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[6:4] = 3'b101
I2C_SCL1	I/O	I2C2_SCL_M0/PWM11/TKEY6_M1/TKEY_DRIVE_M5/I2C1_SCL_M3/I2S_SD1_M1/PWM_AUDIO_R_M3/GPIO0_D6_u	GRF_GPIO0D_IOMUX_H[10:8] = 3'b101

In RK2206, Digital Audio Codec also has an I2C interface, and the I2C2 and Digital Audio Codec shares the same IOs. Digital Audio Codec can use IOs by default, if I2C2 wants to used IOs, processor must set GRF_CON_16[12] to 1'b1, wait GRF_SOC_STATUS[19] equal to 1'b1, then I2C2 can used IOs to transmit. When I2C2 finished communication, processor needs to set GRF_CON_16[12] back to 1'b0.

8.6 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections: transmit only mode, receive only mode, and mix mode. Users are strongly advised to follow

- Transmit only mode (I2C_CON[1:0]=2'b00)

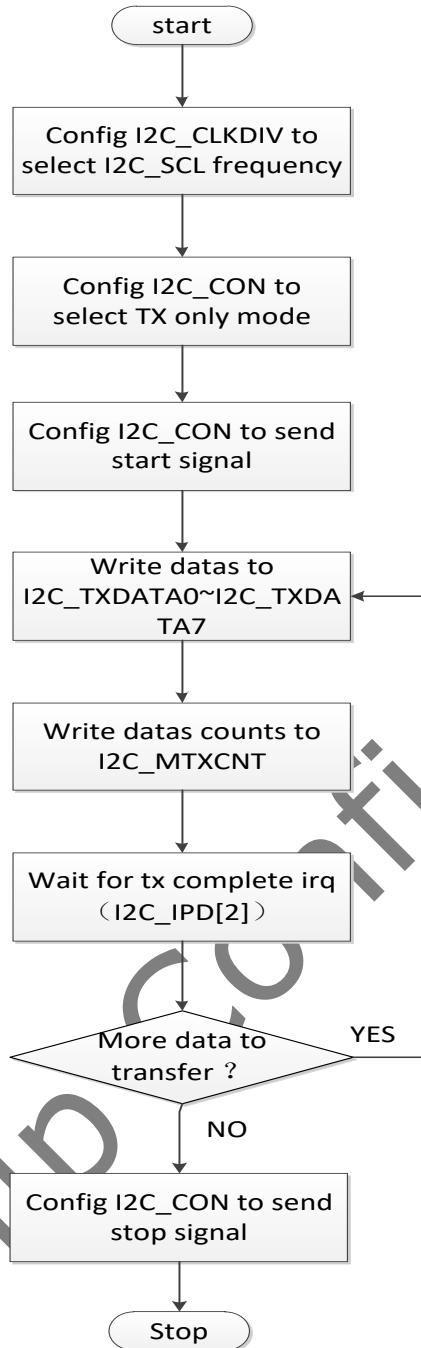


Fig. 8-6 I2C Flow chat for transmit only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

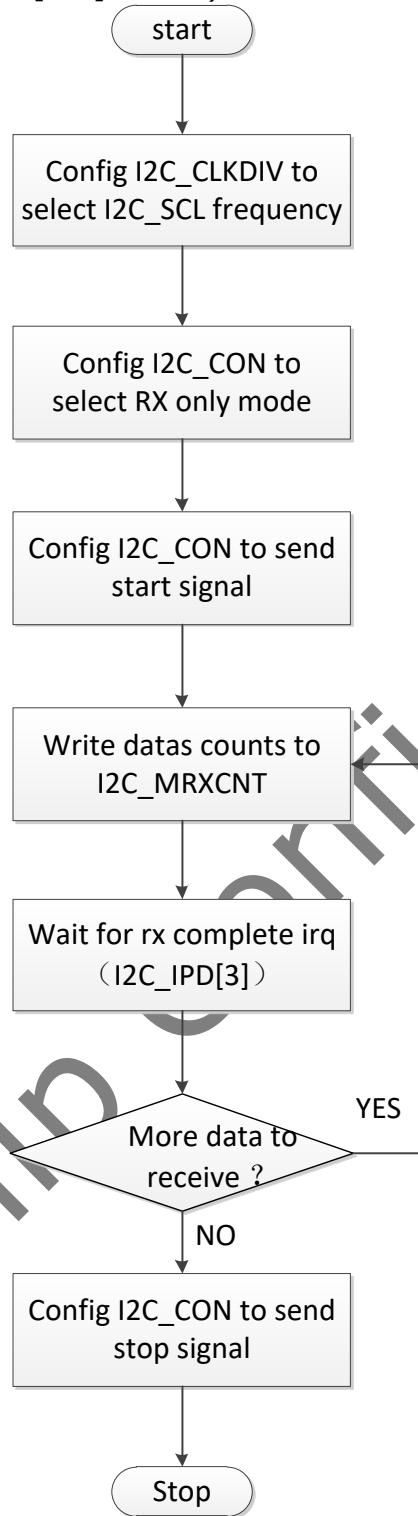


Fig. 8-7 I2C Flow chat for receive only mode

- Mix mode ($\text{I2C_CON}[1:0]=2'b01$ or $\text{I2C_CON}[1:0]=2'b11$)

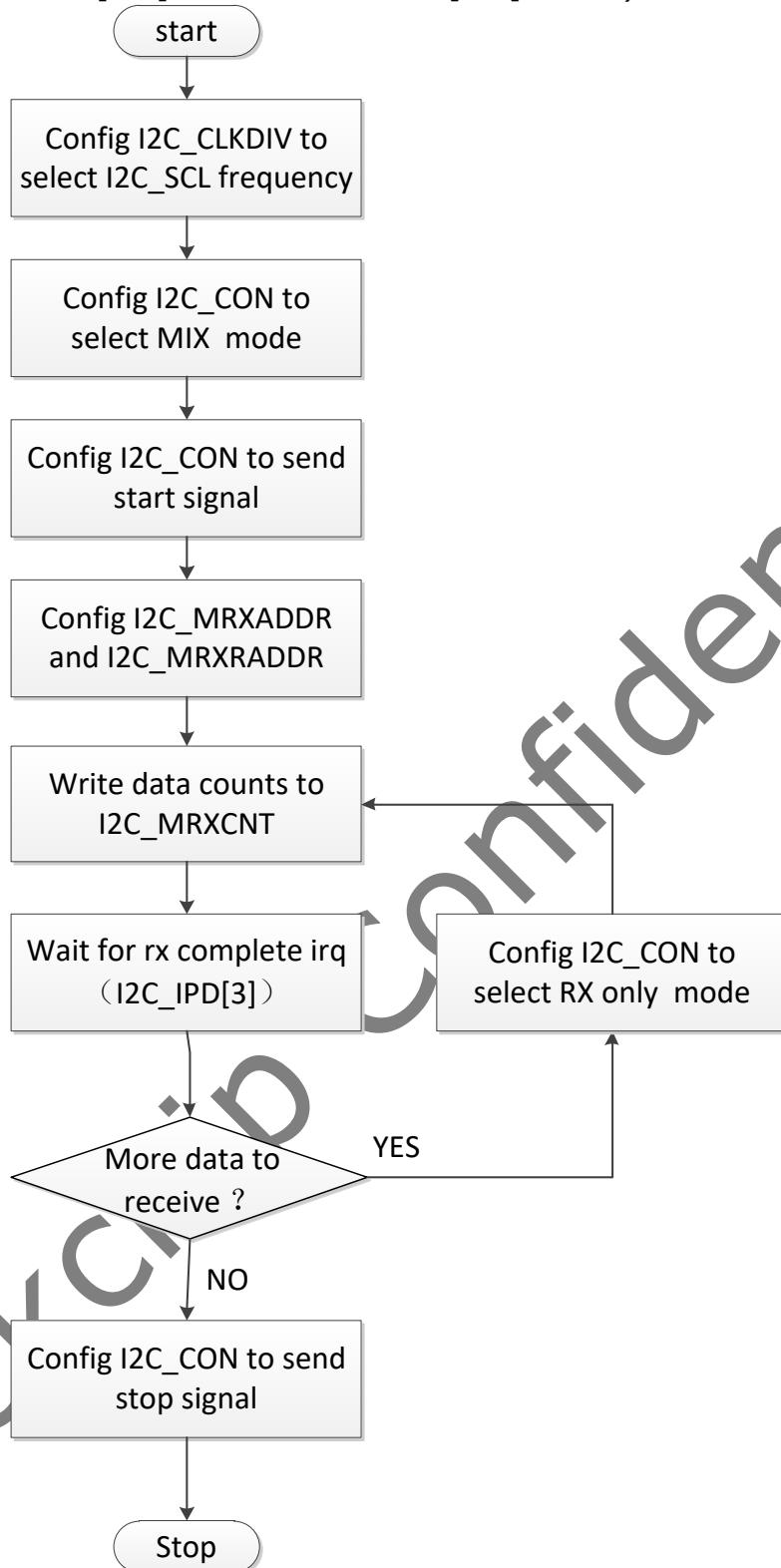


Fig. 8-8 I2C Flow chat for mix mode

Chapter 9 LPO Trim Controller

9.1 Overview

LPO (Low Power Oscillator) Trim Controller is designed to monitor the power up and provide an integrated internal low frequency 32.768KHz clock (RTCCLK) for the system. In order to improve accuracy of the RTCCLK, there is a trim block to calibrate the frequency of RTCCLK.

LPO has below main features:

- Low power and low cost integrated internal RTC clock oscillator
- $\pm 5\%$ accuracy after trim calibration
- Hardware and software trim calibration

9.2 Block Diagram

The diagram of LPO and LPO TRIM Controller are shown below.

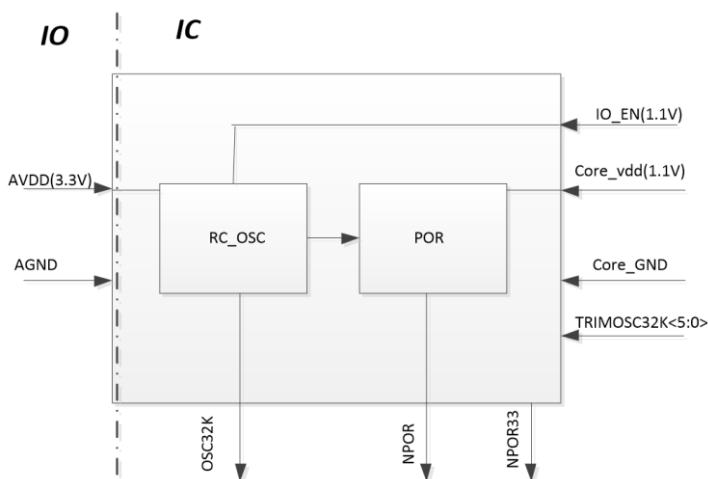


Fig. 9-1 LPO Diagram

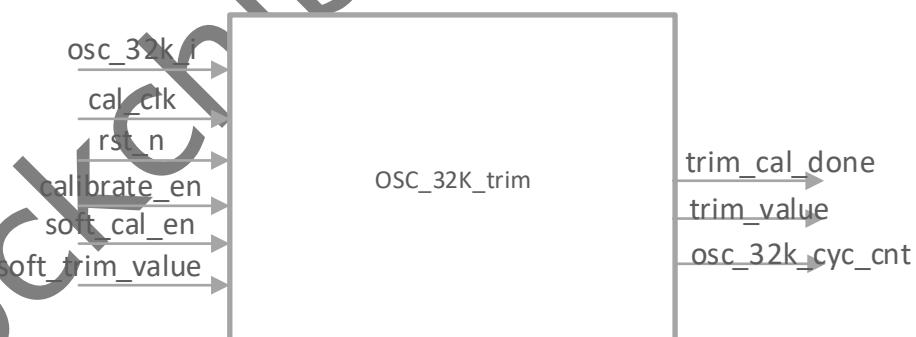


Fig. 9-2 LPO TRIM Diagram

9.3 Function Description

9.3.1 Power Up

The Internal POR circuit monitors the AVDD3.3V and Core VDD1.1V of the whole system. The power supply sequence is AVDD3.3 V and then Core VDD1.1V. After the VDD1.1V power is stable, the POR assert NPOR_RTC until it powers down again. The POROSC also provides a delayed NPOR to the system, the delayed time is 128 cycles of the un-calibrated RTCCLK.

An external NPOR from IO is also provided to reset the whole system. If the external NPOR IO is not used, the default pull-up IO state will ensure the functional integrity of the power

up.

9.3.2 Low Power Clock

There are two RTCCLK source in current system, whether using internal LPO oscillator or internal more precise 32K which is divided from system PLL. The internal RTC oscillator can be used in less power sensitive application when wifi device is in sleep mode than using an external RTC clock source. The internal RTC oscillator begins to work after AVDD3.3V is stable, and its RTC clock output is controlled by IO_EN which is enable by default.

9.3.3 Low Power Oscillator Trim

When Internal RTC clock is used, it has a corresponding trim block to compute the target trim value and feedback to POROSC to adjust the internal RTC precision. After power up, it is auto calibrated, and it asserts calibration done to processor. To considerate the temperature variation, the user can re-calibrate the RTC clock by controlling the toggle of calibrate_en, and then it will also assert done to processor. The calibrated frequency of the RTC clock is equal or greater than the standard RTC frequency. If user wants to get a lower calibrated frequency clock, the software can get the trim_value as the reference value, then switch the trim mode to software trim mode to set the manual trim. No matter the trim mode, the software can acquire the more precise frequency of the RTC clock by reading the osc_32k_cyc_cnt that computed by precise 40MHz clock. Below shows the calibration state machine of the TRIM block.

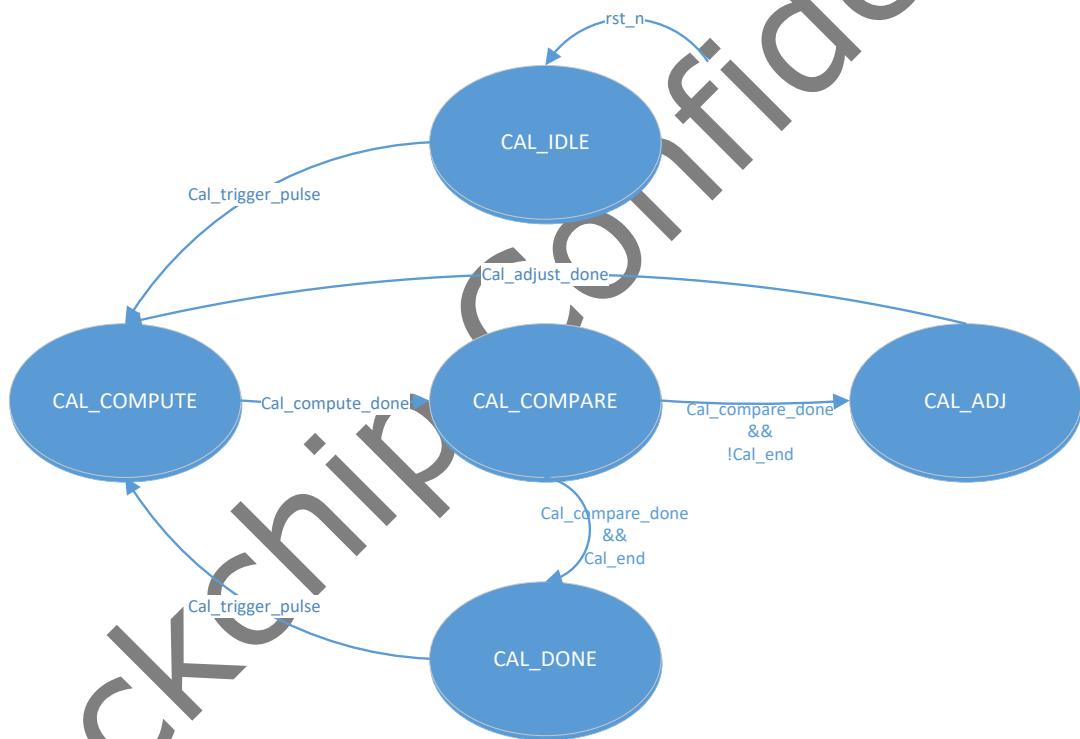


Fig. 9-3 Calibration State Machine

9.4 Register Description

9.4.1 Internal Address Mapping

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

Base Address for LPO TRIM Controller is 0x41070000.

9.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
LPO OSC CTRL	0x0000	W	0x00000001	Oscillator Control Register

Name	Offset	Size	Reset Value	Description
LPO_TRIM_CTRL	0x0004	W	0x00000021	TRIM Control Register
LPO_TRIM_INTE	0x0008	W	0x00000000	TRIM Interrupt Enable Register
LPO_TRIM_INTS	0x000c	W	0x00000000	TRIM Interrupt Status Register
LPO_TRIM_VALUE	0x0010	W	0x00000000	Trim Value Register
LPO_OSC_CYCLE_CNT	0x0014	W	0x00000000	Oscillator Clock Cycle Counter Register
LPO_OSC_CYCLE_CNT_ST_D	0x0018	W	0x000004c4	Standard 32KHz Oscillator Clock Cycle counter

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

9.4.3 Detail Register Description

LPO OSC CTRL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0	reserved
0	RW	0x1	osc_32k_en Oscillator 32K Enable 1'b0: Disable NPOR and Oscillator 1'b1: Enable NPOR and Oscillator

LPO TRIM CTRL

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:7	RO	0x0	reserved
6:4	RW	0x2	adj_wait_cyc_num Adjust wait clock cycle number When trim value change, the trim controller should wait adj_wait_cyc_num cycles to wait oscillator lock.
3	RW	0x0	calibrate_trend_sel Calibration result control 1'b0: Trimed freq is higher than standard 32.768 kHz 1'b1: Trimed freq is lower than standard 32.768 kHz
2	RW	0x0	trim_src_sel Trim source selection 1'b0: Hardware trim value 1'b1: Software trim value

Bit	Attr	Reset Value	Description
1	RW	0x0	soft_calibrate_en Enable internal 32K oscillator software calibration 1'b0: Disable 1'b1: Enable This bit is auto cleared after calibration done
0	R/W SC	0x1	calibrate_en Enable internal 32K oscillator hardware calibration 1'b0: Disable 1'b1: Enable This bit is auto cleared after calibration done

LPO TRIM INTE

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RO	0x0	write_enable Write enable for lower 16 bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0	reserved
0	RW	0x0	calibrate_int_en Calibration Interrupt Enable 1'b0: Disable 1'b1: Enable

LPO TRIM INTS

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	calibrate_int_st Calibration Interrupt Status 1'b1: Calibration done 1'b0: Calibration in progress or interrupt has been cleared Write 1 to clear the interrupt

LPO TRIM VALUE

Address: Operational Base + offset (0x0010)

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

Bit	Attr	Reset Value	Description
13:8	RO	0x00	hardware_trim_value Hardware Trim Value This value is auto generated by hardware calibration after calibration done each time.
7:6	RO	0x0	reserved
5:0	RW	0x00	soft_trim_value Software Trim Value The trim value when trim_src_sel is set to "1".

LPO OSC CYCLE CNT

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	write_enable Write enable for lower 16 bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:0	RO	0x000	osc_32k_cyc_cnt Oscillator 32K Cycle Counter. This value can be used to calculate the precise frequency of 32K. This cycle is counted based on XTAL frequency.

LPO OSC CYCLE CNT STD

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RO	0x000	write_enable Write enable for lower 16 bits, each bit is individual 1'b0: Write access disable 1'b1: Write access enable
15:12	RO	0x0	reserved
11:0	RW	0x4c4	osc_32k_cyc_cnt_std Criteria 32k counter cycles based on XTAL frequency. If the trimed osc frequency reach the most precise value, the trim calibration done.

9.5 Interface Description

N/A

9.6 Application Notes

9.6.1 TRIM Table

Table 9-1 Trim Table

N	<5>	<4>	<3>	<2>	<1>	<0>	value	accuracy of LSB/2
32	1	0	0	0	0	0	58.44%	0.65%
31	1	0	0	0	0	1	59.21%	0.66%
30	1	0	0	0	1	0	60.00%	0.67%
29	1	0	0	0	1	1	60.81%	0.68%
28	1	0	0	1	0	0	61.64%	0.68%
27	1	0	0	1	0	1	62.50%	0.69%
26	1	0	0	1	1	0	63.38%	0.70%
25	1	0	0	1	1	1	64.29%	0.71%
24	1	0	1	0	0	0	65.22%	0.72%
23	1	0	1	0	0	1	66.18%	0.74%
22	1	0	1	0	1	0	67.16%	0.75%
21	1	0	1	0	1	1	68.18%	0.76%
20	1	0	1	1	0	0	69.23%	0.77%
19	1	0	1	1	0	1	70.31%	0.78%
18	1	0	1	1	1	0	71.43%	0.79%
17	1	0	1	1	1	1	72.58%	0.81%
16	1	1	0	0	0	0	73.77%	0.82%
15	1	1	0	0	0	1	75.00%	0.83%
14	1	1	0	0	1	0	76.27%	0.85%
13	1	1	0	0	1	1	77.59%	0.86%
12	1	1	0	1	0	0	78.95%	0.88%
11	1	1	0	1	0	1	80.36%	0.89%
10	1	1	0	1	1	0	81.82%	0.91%
9	1	1	0	1	1	1	83.33%	0.93%
8	1	1	1	0	0	0	84.91%	0.94%
7	1	1	1	0	0	1	86.54%	0.96%
6	1	1	1	0	1	0	88.24%	0.98%
5	1	1	1	0	1	1	90.00%	1.00%
4	1	1	1	1	0	0	91.84%	1.02%
3	1	1	1	1	0	1	93.75%	1.04%
2	1	1	1	1	1	0	95.74%	1.06%
1	1	1	1	1	1	1	97.83%	1.09%
0	0	0	0	0	0	0	100.00%	1.11%
-1	0	0	0	0	0	1	102.27%	1.14%
-2	0	0	0	0	1	0	104.65%	1.16%
-3	0	0	0	0	1	1	107.14%	1.19%
-4	0	0	0	1	0	0	109.76%	1.22%
-5	0	0	0	1	0	1	112.50%	1.25%
-6	0	0	0	1	1	0	115.38%	1.28%
-7	0	0	0	1	1	1	118.42%	1.32%
-8	0	0	1	0	0	0	121.62%	1.35%
-9	0	0	1	0	0	1	125.00%	1.39%
-10	0	0	1	0	1	0	128.57%	1.43%
-11	0	0	1	0	1	1	132.35%	1.47%
-12	0	0	1	1	0	0	136.36%	1.52%
-13	0	0	1	1	0	1	140.63%	1.56%
-14	0	0	1	1	1	0	145.16%	1.61%
-15	0	0	1	1	1	1	150.00%	1.67%
-16	0	1	0	0	0	0	155.17%	1.72%
-17	0	1	0	0	0	1	160.71%	1.79%

N	<5>	<4>	<3>	<2>	<1>	<0>	value	accuracy of LSB/2
-18	0	1	0	0	1	0	166.67%	1.85%
-19	0	1	0	0	1	1	173.08%	1.92%
-20	0	1	0	1	0	0	180.00%	2.00%
-21	0	1	0	1	0	1	187.50%	2.08%
-22	0	1	0	1	1	0	195.65%	2.17%
-23	0	1	0	1	1	1	204.55%	2.27%
-24	0	1	1	0	0	0	214.29%	2.38%
-25	0	1	1	0	0	1	225.00%	2.50%
-26	0	1	1	0	1	0	236.84%	2.63%
-27	0	1	1	0	1	1	250.00%	2.78%
-28	0	1	1	1	0	0	264.71%	2.94%
-29	0	1	1	1	0	1	281.25%	3.13%
-30	0	1	1	1	1	0	300.00%	3.33%
-31	0	1	1	1	1	1	321.43%	3.57%

Chapter 10 DSP

10.1 Overview

The DSP is a highly optimized audio processor geared for efficient execution of audio and voice CODECs, and pre- and post-processing modules. It has four multipliers, three VLIW slots, good support for 32x16-bit and 32x32-bit multiplication, a true 64-bit data path. The support for 32-bit audio, including automatic vectoring, provides much better performance on out-of-the-box C programs and voice algorithms.

Any algorithm written in C/C++ can simply be recompiled to get modest performance improvements. All DSP operations can be used as intrinsic in standard C/C++ applications. The DSP system is composed of one DSP core, Local Memory, and AXI bridge interface. The DSP system supports following features:

- DSP Core
 - Contain a 16-entry, 64-bit register file, Each register can hold one or two, 24 or 32-bit operands, one or four 16-bit operands or one 56- or 64-bit operand
 - Support 3 very long instruction word (VLIW) slots
 - 64bit VLIW bundle instruction
 - Support 2 fixed 32x32 Multiple Accumulator (MACs) per cycle
 - Support 4 fixed 24x24 MACs per cycle
 - Support 4 fixed 32x16 MACs per cycle
 - Support 4 fixed 16x16 MACs per cycle
 - Support 64bit accumulator
 - Integrate 2-way SIMD vector floating point (VFPU)
 - Support one circular buffer
 - Support 32bit integer divider
 - CLAMPS instruction test signal integer ranging from 7 to 22 bit
 - Support normalize shift amount instruction
 - Support max and min operation for signed and unsigned value
 - Support sign Extend instruction (SEXT)
 - Enable Density Instructions
 - Boolean Registers
 - Zero-overhead loop instructions
 - Synchronize instruction
- Bus interface
 - One AXI4 master with 64bit data width
 - One AXI4 slave with 64bit data width
 - AXI interface is synchronized to DSP core
- Local Memory
 - Support 16KB Icache
 - Support 16KB Dcache
 - Support 32KB internal instruction memory
 - Support 192KB internal data memory
- Global features
 - Configurable boot address
 - Support fatal error interrupt
 - Global Clock Gating
 - Functional Unit Clock Gating
 - Asynchronous Reset

10.2 Block Diagram

The DSP consists of Debug Module, DSP core, Local Memory and AXI interface. The DSP core consists of a number of main blocks. The DSP has a number of local-memory interface options that allow fast access to instructions and data over a wide range of different applications. Debug module can be interfaced to a separate JTAG module for chip debug

and test. AXI interface includes one AXI master and one AXI slave. The block diagram for DSP is shown below.

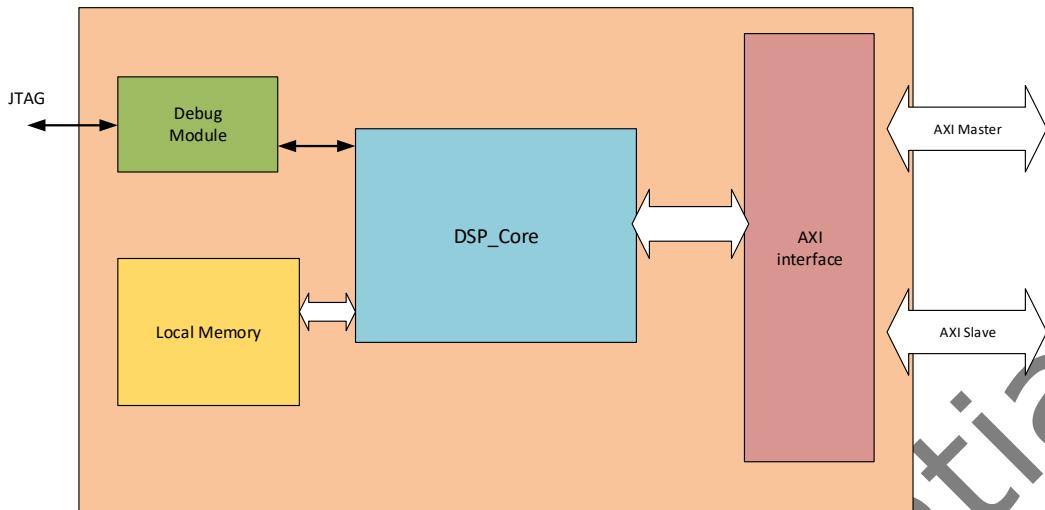


Fig. 10-1 DSP Block Diagram

10.3 Application Notes

10.3.1 Clock and Reset

10.3.1.1 Clock Domains

The DSP is fully synchronized, AXI interface and DSP core has the same clock.

10.3.1.2 Reset and Reset Vector

There are two reset signals in the DSP. Breset is the reset signal to DSP core, AXI interface and Local memory. Dreset is the reset signal to Debug module. All the reset signal is active high, and can be software controlled by write registers in CRU.

Normally, reset vector should be at address 0x30000000 in instruction RAM. It can also be placed at other address: Firstly, set the GRF_DSP_CON0[4](statvectorsel) to 1'b1, and set GRF_DSP_CON1[31:0](altresetvec) to the reset vector address, such as 0x20020000 in share memory. Secondly, remove the reset, then the DSP core will boot from address pointed by altresetvec signal.

10.3.2 Interrupt Application

The Interrupts connect to the input of DSP from interrupt control module (INTC), so DSP just handle 1bit active-high level interrupt from the DSP interface, please refer to the INTC section.

DSP also output 1bit interrupt signal when PFatalError signal is high, which means a fatal error condition occurs. This will be handled by MCU.

10.3.3 Code Initialization

There is no ROM in the DSP, so DSP code should be loaded to internal instruction RAM before DSP works. This could be executed by MCU and DMA.

There is a mechanism to initialize local instruction and data RAMs after a reset. If the RunStall input is asserted during reset and remains asserted after the reset is removed, the processor will be in a stalled state. However requests to local instruction and data RAMs (I/D TCM) can still occur. Data and instruction could be loaded by AXI slave. The RunStall signal can be controlled by GRF_DSP_CON0[5]. After RunStall is dis-assert and reset is removed, DSP will work.

10.3.4 TCM access MUX

I/D TCM can be accessed by DSP and other system masters. The mux can be configured by GRF_SOC_CON0[8:5], and the TCM can be accessed by other system masters default.

Before DSP need to access the TCM, the mux should be switched to DSP firstly.

Chapter 11 DMAC

11.1 Overview

The DMAC supports all kinds of burst type and data size, which defined in AHB protocol. It can support on-the-fly DMA transfer on AHB bus devices and AHB bus memory. There is only one DMAC in the chip.

DMAC supports the following features:

- Provide AHB-to-AHB bus protocol translation
- Support AHB-to-AHB DMA or Single AHB DMA
- Support one AHB slave interface and one AHB master interface
- Supports six channels
- one channel one interrupt, up to six interrupts
- Supports 16 peripheral request
- Support byte, half word and word data transfer sizes
- Support incremental and fixed addressing mode
- Support block and software DMA transfer mode
- Support channel priority arbitration
- All channels support LLP transfer
- All channels support auto-reload
- Support memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral DMA transfers

11.2 Block Diagram

The DMAC consists of the following main functional blocks.

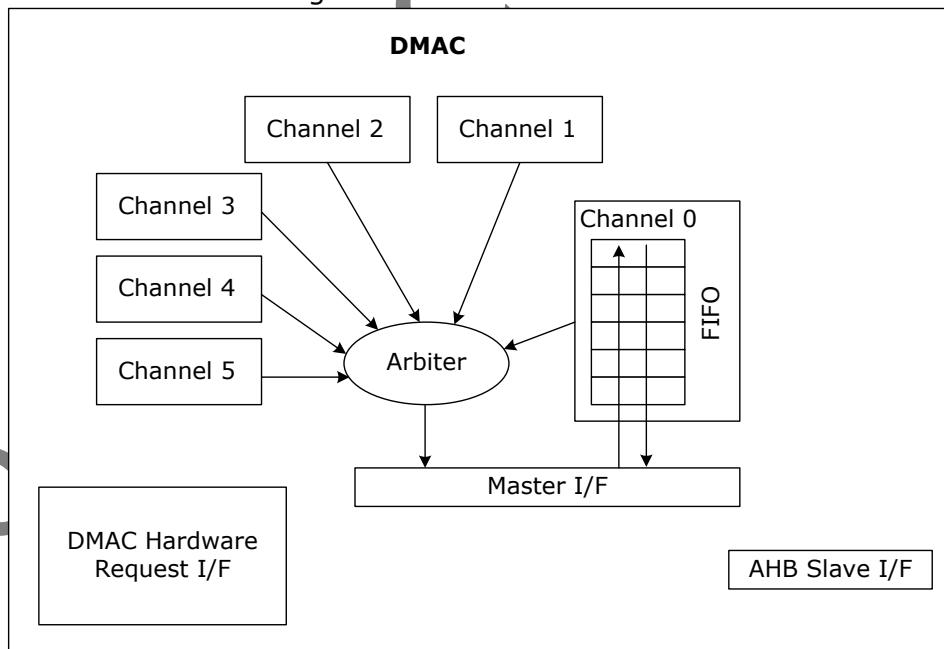


Fig. 11-1 Host Controller Block Diagram

- DMAC hardware request interface

Uses hardware signals to control a single or burst transaction between the DMAC and the peripheral terminal.

- Up to six channels

Channel serves as Read/write data path between a source peripheral and a destination peripheral that occurs through the channel FIFO. If the source peripheral is not a memory, then a source handshaking interface is assigned to the channel. If the

destination peripheral is not a memory, then a destination handshaking interface is assigned to the channel. Source and destination handshaking interfaces can be assigned dynamically by programming the channel registers.

- FIFO per channel for source and destination

Each channel has a 128byte depth FIFO.

- Arbiter

Each channel has to arbitrate for the master interface.

- One AHB mater interface

DMAC is a master on the AHB bus, reading data from the source and writing it to the destination over the AHB bus.

- One AHB slave interface

The AHB interface over which the DMAC is programmed.

11.3 Register Description

11.3.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DMA_SAR0	0x0000	W	0x00000000	Channel 0 Source Address Register
DMA_DAR0	0x0008	W	0x00000000	Channel 0 Destination Address Register
DMA_LLPO	0x0010	W	0x00000000	Channel 0 Linked List Pointer Register
DMA_CTL0	0x0018	W	0x00004801	Channel 0 Control Register
DMA_SSTAT0	0x0020	W	0x00000000	Source Status Register for Channel 0
DMA_DSTAT0	0x0028	W	0x00000000	Destination Status Register for Channel 0
DMA_SSTATAR0	0x0030	W	0x00000000	Source Status Address Register for Channel 0
DMA_DSTATAR0	0x0038	W	0x00000000	Destination Status Address Register for Channel 0
DMA_CFG0	0x0040	W	0x00000c00	Channel 0 Configuration Register
DMA_SGR0	0x0048	W	0x00000000	Channel 0 Source Gather Register
DMA_DSR0	0x0050	W	0x00000000	Channel 0 Destination Scatter Register
DMA_SAR1	0x0058	W	0x00000000	Channel 1 Source Address Register
DMA_DAR1	0x0060	W	0x00000000	Channel 1 Destination Address Register
DMA_LLPI	0x0068	W	0x00000000	Channel 1 Linked List Pointer Register
DMA_CTL1	0x0070	W	0x00004801	Channel 1 Control Register
DMA_SSTAT1	0x0078	W	0x00000000	Source Status Register for Channel 1
DMA_DSTAT1	0x0080	W	0x00000000	Destination Status Register for Channel 1

Name	Offset	Size	Reset Value	Description
DMA_SSTATAR1	0x0088	W	0x00000000	Source Status Address Register for Channel 1
DMA_DSTATAR1	0x0090	W	0x00000000	Destination Status Address Register for Channel 1
DMA_CFG1	0x0098	W	0x00000c00	Channel 1 Configuration Register
DMA_SGR1	0x00a0	W	0x00000000	Channel 1 Source Gather Register
DMA_DSR1	0x00a8	W	0x00000000	Channel 1 Destination Scatter Register
DMA_SAR2	0x00b0	W	0x00000000	Channel 2 Source Address Register
DMA_DAR2	0x00b8	W	0x00000000	Channel 2 Destination Address Register
DMA_LL2	0x00c0	W	0x00000000	Channel 2 Linked List Pointer Register
DMA_CTL2	0x00c8	W	0x00004801	Channel 2 Control Register
DMA_SSTAT2	0x00d0	W	0x00000000	Source Status Register for Channel 2
DMA_DSTAT2	0x00d8	W	0x00000000	Destination Status Register for Channel 2
DMA_SSTATAR2	0x00e0	W	0x00000000	Source Status Address Register for Channel 2
DMA_DSTATAR2	0x00e8	W	0x00000000	Destination Status Address Register for Channel 2
DMA_CFG2	0x00f0	W	0x00000c00	Channel 2 Configuration Register
DMA_SGR2	0x00f8	W	0x00000000	Channel 2 Source Gather Register
DMA_DSR2	0x0100	W	0x00000000	Channel 2 Destination Scatter Register
DMA_SAR3	0x0108	W	0x00000000	Channel 3 Source Address Register
DMA_DAR3	0x0110	W	0x00000000	Channel 3 Destination Address Register
DMA_LL3	0x0118	W	0x00000000	Channel 3 Linked List Pointer Register
DMA_CTL3	0x0120	W	0x00004801	Channel 3 Control Register
DMA_SSTAT3	0x0128	W	0x00000000	Source Status Register for Channel 3
DMA_DSTAT3	0x0130	W	0x00000000	Destination Status Register for Channel 3
DMA_SSTATAR3	0x0138	W	0x00000000	Source Status Address Register for Channel 3
DMA_DSTATAR3	0x0140	W	0x00000000	Destination Status Address Register for Channel 3
DMA_CFG3	0x0148	W	0x00000c00	Channel 3 Configuration Register
DMA_SGR3	0x0150	W	0x00000000	Channel 3 Source Gather Register

Name	Offset	Size	Reset Value	Description
DMA_DSR3	0x0158	W	0x00000000	Channel 3 Destination Scatter Register
DMA_SAR4	0x0160	W	0x00000000	Channel 4 Source Address Register
DMA_DAR4	0x0168	W	0x00000000	Channel 4 Destination Address Register
DMA_LL4	0x0170	W	0x00000000	Channel 4 Linked List Pointer Register
DMA_CTL4	0x0178	W	0x00004801	Channel 4 Control Register
DMA_SSTAT4	0x0180	W	0x00000000	Source Status Register for Channel 4
DMA_DSTAT4	0x0188	W	0x00000000	Destination Status Register for Channel 4
DMA_SSTATAR4	0x0190	W	0x00000000	Source Status Address Register for Channel 4
DMA_DSTATAR4	0x0198	W	0x00000000	Destination Status Address Register for Channel 4
DMA_CFG4	0x01a0	W	0x00000c00	Channel 4 Configuration Register
DMA_SGR4	0x01a8	W	0x00000000	Channel 4 Source Gather Register
DMA_DSR4	0x01b0	W	0x00000000	Channel 4 Destination Scatter Register
DMA_SAR5	0x01b8	W	0x00000000	Channel 5 Source Address Register
DMA_DAR5	0x01c0	W	0x00000000	Channel 5 Destination Address Register
DMA_LL5	0x01c8	W	0x00000000	Channel 5 Linked List Pointer Register
DMA_CTL5	0x01d0	W	0x00004801	Channel 5 Control Register
DMA_SSTAT5	0x01d8	W	0x00000000	Source Status Register for Channel 5
DMA_DSTAT5	0x01e0	W	0x00000000	Destination Status Register for Channel 5
DMA_SSTATAR5	0x01e8	W	0x00000000	Source Status Address Register for Channel 5
DMA_DSTATAR5	0x01f0	W	0x00000000	Destination Status Address Register for Channel 5
DMA_CFG5	0x01f8	W	0x00000c00	Channel 5 Configuration Register
DMA_SGR5	0x0200	W	0x00000000	Channel 5 Source Gather Register
DMA_DSR5	0x0208	W	0x00000000	Channel 5 Destination Scatter Register
DMA_RAWTFR	0x02c0	W	0x00000000	Raw Status for IntTfr Interrupt
DMA_RAWBLOCK	0x02c8	W	0x00000000	Raw Status for IntBlock Interrupt
DMA_RAWSRCTRN	0x02d0	W	0x00000000	Raw Status for IntSrcTrans Interrupt

Name	Offset	Size	Reset Value	Description
DMA_RAWDSTTRAN	0x02d8	W	0x00000000	Raw Status for IntDstTran Interrupt
DMA_RAWERR	0x02e0	W	0x00000000	Raw Status for IntErr Interrupt
DMA_STATUSTFR	0x02e8	W	0x00000000	Status for IntTfr Interrupt
DMA_STATUSBLOCK	0x02f0	W	0x00000000	Status for IntBlock Interrupt
DMA_STATUSSRCTRAN	0x02f8	W	0x00000000	Status for IntSrcTran Interrupt
DMA_STATUSDSTTRAN	0x0300	W	0x00000000	Status for IntDstTran Interrupt
DMA_STATUSERR	0x0308	W	0x00000000	Status for IntErr Interrupt
DMA_MASKTFR	0x0310	W	0x00000000	Mask for IntTfr Interrupt
DMA_MASKBLOCK	0x0318	W	0x00000000	Mask for IntBlock Interrupt
DMA_MASKSRCTRAN	0x0320	W	0x00000000	Mask for IntSrcTran Interrupt
DMA_MASKDSTTRAN	0x0328	W	0x00000000	Mask for IntDstTran Interrupt
DMA_MASKERR	0x0330	W	0x00000000	Mask for IntErr Interrupt
DMA_CLEARTFR	0x0338	W	0x00000000	Clear for IntTfr Interrupt
DMA_CLEARBLOCK	0x0340	W	0x00000000	Clear for IntBlock Interrupt
DMA_CLEARSRCTRAN	0x0348	W	0x00000000	Clear for IntSrcTran Interrupt
DMA_CLEARDSTTRAN	0x0350	W	0x00000000	Clear for IntDstTran Interrupt
DMA_CLEARERR	0x0358	W	0x00000000	Clear for IntErr Interrupt
DMA_STATUSINT	0x0360	W	0x00000000	The contents of each of the five status registers StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran, StatusErr is ORed to produce a single bit for each interrupt type in the Combined Status Register (StatusInt). This register is read-only
DMA_DMACFGREG	0x0398	W	0x00000000	DMA Configuration Register
DMA_CHENREG	0x03a0	W	0x00000000	DMA Channel Enable Register

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

11.3.2 Detail Register Description

DMA_SAR0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DAR0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>dar</p> <p>Current destination address of DMA transfer.</p> <p>Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.</p>

DMA_LLPO

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>loc</p> <p>Starting address.</p> <p>In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.</p>
1:0	RW	0x0	<p>lms</p> <p>List master select.</p> <p>Identifies the AHB layer/interface where the memory device that stores the next linked list item resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p>

DMA_CTL0

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
44	RW	0x0	<p>done</p> <p>Done bit</p> <p>If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.</p>

Bit	Attr	Reset Value	Description
43:32	RW	0x801	<p>block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. $b = \log_2(\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 31$ Bits 43:b+1 do not exist and return 0 on a read.</p>
31:29	RO	0x0	reserved
28	RW	0x0	<p>llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.</p>
27	RW	0x0	<p>llp_dst_en Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms Source master select. Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed. 2'b00: AHB master 1 2'b01: AHB master 2 Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms Destination master select. Identifies the Master Interface layer where the destination device (peripheral or memory) resides. 2'b00: AHB master 1 2'b01: AHB master 2 Reset Value: DMAH_CHx_DMS[1:0]</p>

Bit	Attr	Reset Value	Description
22:20	RW	0x0	<p>tt_fc Transfer type and flow control. The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en Destination scatter enable bit: 1'b0: Scatter disabled 1'b1: Scatter enabled Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field.</p> <p>NOTE: This value is not related to the AHB bus master HBURST bus.</p>

Bit	Attr	Reset Value	Description
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface.</p> <p>NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change." 2'b00: Increment 2'b01: Decrement 2'b1x: No change</p> <p>NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary.</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change." 2'b00: Increment 2'b01: Decrement 2'b1x: No change</p> <p>NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary.</p> <p>Reset Value: 0x0</p>
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_Mx_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>

Bit	Attr	Reset Value	Description
3:1	RW	0x0	<p>dst_tr_width Destination transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width.</p> <p>This value must be less than or equal to DMAH_Mk_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides.</p> <p>Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTATO

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.</p>

DMA_DSTATO

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.</p>

DMA_SSTATAR0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.</p>

DMA_DSTATAR0

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.</p>

DMA_CFG0

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	<p>dest_per</p> <p>Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface.</p> <p>NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.</p>
42:39	RW	0x8	<p>src_per</p> <p>Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface.</p> <p>NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.</p>
38	RW	0x0	<p>ss_upd_en</p> <p>Source status update enable.</p> <p>Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high.</p> <p>NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.</p>
37	RW	0x0	<p>ds_upd_en</p> <p>Destination status update enable.</p> <p>Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.</p>

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.</p> <p>Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width.</p> <p>1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers.</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcemode Flow control mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled.</p> <p>1'b1: Source transaction requests are not serviced until a destination transaction request occurs.</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>

Bit	Attr	Reset Value	Description
30	RW	0x0	reload_src Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal hlock is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L.
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src Source software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst Destination software or hardware handshaking select. This register selects which of the handshaking interfaces hardware or software is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty 1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended 1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1). A programmed value outside this range will cause erroneous behavior.</p> <p>Reset Value: Channel Number. For example: Chan0=0 Chan1=1</p>
4:0	RO	0x0	reserved

DMA_SGR0

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.
19:0	RW	0x00000	sgi Source gather interval.

DMA_DSR0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.
19:0	RW	0x00000	dsi Destination scatter interval.

DMA_SAR1

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DAR1

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dar Current destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

DMA_LL1

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	loc Starting address. In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.
1:0	RW	0x0	lms List master select. Identifies the AHB layer/interface where the memory device that stores the next linked list item resides. 2'b00: AHB master 1 2'b01: AHB master 2

DMA_CTL1

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
44	RW	0x0	done Done bit If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.
43:32	RW	0x801	block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. b = log2(DMAH_CHx_MAX_BLK_SIZE + 1) + 31 Bits 43:b+1 do not exist and return 0 on a read.
31:29	RO	0x0	reserved
28	RW	0x0	llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.

Bit	Attr	Reset Value	Description
27	RW	0x0	<p>llp_dst_en</p> <p>Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms</p> <p>Source master select.</p> <p>Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms</p> <p>Destination master select.</p> <p>Identifies the Master Interface layer where the destination device (peripheral or memory) resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_DMS[1:0]</p>
22:20	RW	0x0	<p>tt_fc</p> <p>Transfer type and flow control.</p> <p>The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent:</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en</p> <p>Destination scatter enable bit:</p> <p>1'b0: Scatter disabled 1'b1: Scatter enabled</p> <p>Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change." 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary.</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary. Reset Value: 0x0</p>

Bit	Attr	Reset Value	Description
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_M_x_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>
3:1	RW	0x0	<p>dst_tr_width Destination Transfer Width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to DMAH_M_k_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides. Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTAT1

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.

DMA_DSTAT1

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.

DMA_SSTATAR1

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

DMA_DSTATAR1

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

DMA_CFG1

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	dest_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	RW	0x8	src_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	RW	0x0	ss_upd_en Source status update enable. Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high. NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.
37	RW	0x0	ds_upd_en Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.</p> <p>Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width 1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcmode Flow control mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled 1'b1: Source transaction requests are not serviced until a destination transaction request occurs</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>
30	RW	0x0	<p>reload_src Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.</p>

Bit	Attr	Reset Value	Description
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L.
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src</p> <p>Source software or hardware handshaking select.</p> <p>This register selects which of the handshaking interfaces hardware or software is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored</p> <p>1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored</p> <p>If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst</p> <p>Destination software or hardware handshaking select.</p> <p>This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored</p> <p>1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored</p> <p>If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty</p> <p>Indicates if there is data left in the channel FIFO.</p> <p>Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty</p> <p>1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp</p> <p>Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended.</p> <p>1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior</p> <p>Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1).</p> <p>A programmed value outside this range will cause erroneous behavior.</p> <p>Reset Value: Channel Number. For example: Chan0=0 Chan1=1</p>
4:0	RO	0x0	reserved

DMA_SGR1

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.
19:0	RW	0x00000	sgi Source gather interval.

DMA_DSR1

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.
19:0	RW	0x00000	dsi Destination scatter interval.

DMA_SAR2

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DAR2

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dar Current destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

DMA_LL2

Address: Operational Base + offset (0x00c0)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	loc Starting address. In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.
1:0	RW	0x0	lms List master select. Identifies the AHB layer/interface where the memory device that stores the next linked list item resides. 2'b00: AHB master 1 2'b01: AHB master 2

DMA_CTL2

Address: Operational Base + offset (0x00c8)

Bit	Attr	Reset Value	Description
44	RW	0x0	done Done bit If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.
43:32	RW	0x801	block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. b = log2(DMAH_CHx_MAX_BLK_SIZE + 1) + 31 Bits 43:b+1 do not exist and return 0 on a read.
31:29	RO	0x0	reserved
28	RW	0x0	llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.

Bit	Attr	Reset Value	Description
27	RW	0x0	<p>llp_dst_en</p> <p>Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms</p> <p>Source master select.</p> <p>Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms</p> <p>Destination master select.</p> <p>Identifies the Master Interface layer where the destination device (peripheral or memory) resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_DMS[1:0]</p>
22:20	RW	0x0	<p>tt_fc</p> <p>Transfer type and flow control.</p> <p>The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent:</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en</p> <p>Destination scatter enable bit:</p> <p>1'b0: Scatter disabled 1'b1: Scatter enabled</p> <p>Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field; NOTE: This value is not related to the AHB bus master HBURST bus.</p>
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary.</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary. Reset Value: 0x0</p>

Bit	Attr	Reset Value	Description
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_Mx_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>
3:1	RW	0x0	<p>dst_tr_width Destination Transfer Width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to DMAH_Mk_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides. Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTAT2

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.

DMA_DSTAT2

Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.

DMA_SSTATAR2

Address: Operational Base + offset (0x00e0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

DMA_DSTATAR2

Address: Operational Base + offset (0x00e8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

DMA_CFG2

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	dest_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	RW	0x8	src_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	RW	0x0	ss_upd_en Source status update enable. Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high. NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.
37	RW	0x0	ds_upd_en Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.</p> <p>Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width.</p> <p>1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers.</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcmode Flow control mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled.</p> <p>1'b1: Source transaction requests are not serviced until a destination transaction request occurs.</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>
30	RW	0x0	<p>reload_src Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.</p>

Bit	Attr	Reset Value	Description
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src Source software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst Destination software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty 1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended. 1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1). A programmed value outside this range will cause erroneous behavior.</p> <p>Reset Value: Channel Number. For example: Chan0=0 Chan1=1</p>
4:0	RO	0x0	reserved

DMA_SGR2

Address: Operational Base + offset (0x00f8)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.
19:0	RW	0x00000	sgi Source gather interval.

DMA_DSR2

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.
19:0	RW	0x00000	dsi Destination scatter interval.

DMA_SAR3

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DAR3

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dar Current destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

DMA_LL3

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	loc Starting address. In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.
1:0	RW	0x0	lms List master select. Identifies the AHB layer/interface where the memory device that stores the next linked list item resides. 2'b00: AHB master 1 2'b01: AHB master 2

DMA_CTL3

Address: Operational Base + offset (0x0120)

Bit	Attr	Reset Value	Description
44	RW	0x0	done Done bit If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.
43:32	RW	0x801	block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. b = log2(DMAH_CHx_MAX_BLK_SIZE + 1) + 31 Bits 43:b+1 do not exist and return 0 on a read.
31:29	RO	0x0	reserved
28	RW	0x0	llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.

Bit	Attr	Reset Value	Description
27	RW	0x0	<p>llp_dst_en</p> <p>Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms</p> <p>Source master select.</p> <p>Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms</p> <p>Destination master select.</p> <p>Identifies the Master Interface layer where the destination device (peripheral or memory) resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_DMS[1:0]</p>
22:20	RW	0x0	<p>tt_fc</p> <p>Transfer type and flow control.</p> <p>The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent:</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en</p> <p>Destination scatter enable bit:</p> <p>1'b0: Scatter disabled 1'b1: Scatter enabled</p> <p>Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary.</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary. Reset Value: 0x0</p>

Bit	Attr	Reset Value	Description
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_Mx_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>
3:1	RW	0x0	<p>dst_tr_width Destination Transfer Width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to DMAH_Mk_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides. Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTAT3

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.

DMA_DSTAT3

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.

DMA_SSTATAR3

Address: Operational Base + offset (0x0138)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

DMA_DSTATAR3

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

DMA_CFG3

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	dest_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	RW	0x8	src_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	RW	0x0	ss_upd_en Source status update enable. Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high. NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.
37	RW	0x0	ds_upd_en Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl</p> <p>Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access.</p> <p>HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals. Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode</p> <p>FIFO mode select.</p> <p>Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width</p> <p>1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcemode</p> <p>Flow control mode.</p> <p>Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled.</p> <p>1'b1: Source transaction requests are not serviced until a destination transaction request occurs.</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst</p> <p>Automatic destination reload.</p> <p>The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>
30	RW	0x0	<p>reload_src</p> <p>Automatic source reload.</p> <p>The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.</p>

Bit	Attr	Reset Value	Description
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src Source software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst Destination software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty 1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended. 1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1). A programmed value outside this range will cause erroneous behavior.</p> <p>Reset Value: Channel Number. for example: Chan0=0 Chan1=1</p>
4:0	RO	0x0	reserved

DMA_SGR3

Address: Operational Base + offset (0x0150)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.
19:0	RW	0x00000	sgi Source gather interval.

DMA_DSR3

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.
19:0	RW	0x00000	dsi Destination scatter interval.

DMA_SAR4

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DAR4

Address: Operational Base + offset (0x0168)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dar Current destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

DMA_LL4

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	loc Starting address. In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.
1:0	RW	0x0	lms List master select. Identifies the AHB layer/interface where the memory device that stores the next linked list item resides. 2'b00: AHB master 1 2'b01: AHB master 2

DMA_CTL4

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
44	RW	0x0	done Done bit If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.
43:32	RW	0x801	block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. b = log2(DMAH_CHx_MAX_BLK_SIZE + 1) + 31 Bits 43:b+1 do not exist and return 0 on a read.
31:29	RO	0x0	reserved
28	RW	0x0	llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.

Bit	Attr	Reset Value	Description
27	RW	0x0	<p>llp_dst_en</p> <p>Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms</p> <p>Source master select.</p> <p>Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms</p> <p>Destination master select.</p> <p>Identifies the Master Interface layer where the destination device (peripheral or memory) resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_DMS[1:0]</p>
22:20	RW	0x0	<p>tt_fc</p> <p>Transfer type and flow control.</p> <p>The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent:</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en</p> <p>Destination scatter enable bit:</p> <p>1'b0: Scatter disabled 1'b1: Scatter enabled</p> <p>Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary. Reset Value: 0x0</p>

Bit	Attr	Reset Value	Description
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_M_x_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>
3:1	RW	0x0	<p>dst_tr_width Destination Transfer Width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to DMAH_M_k_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides. Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTAT4

Address: Operational Base + offset (0x0180)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.

DMA_DSTAT4

Address: Operational Base + offset (0x0188)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.

DMA_SSTATAR4

Address: Operational Base + offset (0x0190)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

DMA_DSTATAR4

Address: Operational Base + offset (0x0198)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

DMA_CFG4

Address: Operational Base + offset (0x01a0)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	dest_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	RW	0x8	src_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	RW	0x0	ss_upd_en Source status update enable. Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high. NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.
37	RW	0x0	ds_upd_en Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.</p> <p>Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width 1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcmode Flow control mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled. 1'b1: Source transaction requests are not serviced until a destination transaction request occurs.</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>
30	RW	0x0	<p>reload_src Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.</p>

Bit	Attr	Reset Value	Description
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L.
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src Source software or hardware handshaking select. This register selects which of the handshaking interfaces hardware or software is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst Destination software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored 1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty 1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended. 1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1). A programmed value outside this range will cause erroneous behavior.</p> <p>Reset Value: Channel Number. For example: Chan0=0 Chan1=1</p>
4:0	RO	0x0	reserved

DMA_SGR4

Address: Operational Base + offset (0x01a8)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.
19:0	RW	0x00000	sgi Source gather interval.

DMA_DSR4

Address: Operational Base + offset (0x01b0)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.
19:0	RW	0x00000	dsi Destination scatter interval.

DMA_SAR5

Address: Operational Base + offset (0x01b8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sar Current source address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

DMA_DARS5

Address: Operational Base + offset (0x01c0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dar Current destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

DMA_LL5

Address: Operational Base + offset (0x01c8)

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	loc Starting address. In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary.
1:0	RW	0x0	lms List master select. Identifies the AHB layer/interface where the memory device that stores the next linked list item resides. 2'b00: AHB master 1 2'b01: AHB master 2

DMA_CTL5

Address: Operational Base + offset (0x01d0)

Bit	Attr	Reset Value	Description
44	RW	0x0	done Done bit. If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel.
43:32	RW	0x801	block_ts Block transfer size. When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of the single transaction is determined by TLx.SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at DMAH_CHx_MAX_BLK_SIZE, but the actual block size can be greater. b = log2(DMAH_CHx_MAX_BLK_SIZE + 1) + 31 Bits 43:b+1 do not exist and return 0 on a read.
31:29	RO	0x0	reserved
28	RW	0x0	llp_src_en Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero.

Bit	Attr	Reset Value	Description
27	RW	0x0	<p>llp_dst_en</p> <p>Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPx.LOC is non-zero.</p>
26:25	RW	0x0	<p>sms</p> <p>Source master select.</p> <p>Identifies the Master Interface layer from which the source device (peripheral or memory) is accessed.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_SMS[1:0]</p>
24:23	RW	0x0	<p>dms</p> <p>Destination master select.</p> <p>Identifies the Master Interface layer where the destination device (peripheral or memory) resides.</p> <p>2'b00: AHB master 1 2'b01: AHB master 2</p> <p>Reset Value: DMAH_CHx_DMS[1:0]</p>
22:20	RW	0x0	<p>tt_fc</p> <p>Transfer type and flow control.</p> <p>The following transfer types are supported.</p> <ul style="list-style-type: none"> a. Memory to Memory b. Memory to Peripheral c. Peripheral to Memory d. Peripheral to Peripheral <p>Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.</p> <p>Table 3 lists the decoding for this field.</p> <p>Reset Value: Configuration dependent:</p> <p>TT_FC[0] = 1'b1 TT_FC[1] = DMAH_CHx_FC[1] (!DMAH_CHx_FC[0]) TT_FC[2] = DMAH_CHx_FC[1] ^ DMAH_CHx_FC[0]</p> <p>Dependencies: If the configuration parameter DMAH_CHx_FC is set to DMA_FC_ONLY, then TT_FC[2] does not exist and TT_FC[2] always reads back 0. If DMAH_CHx_FC is set to SRC_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b10. If DMAH_CHx_FC is set to DST_FC_ONLY, then TT_FC[2:1] does not exist and TT_FC[2:1] always reads back 2'b11</p>
19	RO	0x0	reserved
18	RW	0x0	<p>dst_scatter_en</p> <p>Destination scatter enable bit:</p> <p>1'b0: Scatter disabled 1'b1: Scatter enabled</p> <p>Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control.</p>

Bit	Attr	Reset Value	Description
17	RW	0x0	<p>src_gather_en Source gather enable bit: 1'b0: Gather disabled 1'b1: Gather enabled Gather on the source side is applicable only when the CTLx.SINC bit indicates an incrementing or decrementing address control.</p>
16:14	RW	0x1	<p>src_msize Source burst transaction length. Number of data items, each of width CTLx.SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface. Table 1 lists the decoding for this field; NOTE: This value is not related to the AHB bus master HBURST bus.</p>
13:11	RW	0x1	<p>dest_msize Destination burst transaction length. Number of data items, each of width CTLx.DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. NOTE: This value is not related to the AHB bus master HBURST bus.</p>
10:9	RW	0x0	<p>sinc Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.SRC_TR_WIDTH boundary</p>
8:7	RW	0x0	<p>dinc Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change". 2'b00: Increment 2'b01: Decrement 2'b1x: No change NOTE: Incrementing or decrementing is done for alignment to the next CTLx.DST_TR_WIDTH boundary. Reset Value: 0x0</p>

Bit	Attr	Reset Value	Description
6:4	RW	0x0	<p>src_tr_width Source transfer width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to DMAH_M_x_HDATA_WIDTH, where x is the AHB layer 1 to 2 where the source resides. Reset Value: Encoded value; refer to Table 2.</p>
3:1	RW	0x0	<p>dst_tr_width Destination Transfer Width. Table 2 lists the decoding for this field. Mapped to AHB bus "hsize". For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to DMAH_M_k_HDATA_WIDTH, where k is the AHB layer 1 to 2 where the destination resides. Reset Value: Encoded value; refer to Table 2.</p>
0	RW	0x1	<p>int_en Interrupt enable bit. 1'b1: All interrupt-generating sources are enabled 1'b0: All interrupt-generating sources are disable</p>

DMA_SSTAT5

Address: Operational Base + offset (0x01d8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstat Source status information retrieved by hardware from the address pointed to by the contents of the SSTATARx register.

DMA_DSTAT5

Address: Operational Base + offset (0x01e0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstat Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register.

DMA_SSTATAR5

Address: Operational Base + offset (0x01e8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sstatar Pointer from where hardware can fetch the source status information, which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

DMA_DSTATARS

Address: Operational Base + offset (0x01f0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dstatar Pointer from where hardware can fetch the destination status information, which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

DMA_CFG5

Address: Operational Base + offset (0x01f8)

Bit	Attr	Reset Value	Description
46:43	RW	0x1	dest_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the destination of channel x if the CFGx.HS_SEL_DST field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	RW	0x8	src_per Assigns a hardware handshaking interface (0 - DMAH_NUM_HS_INT-1) to the source of channel x if the CFGx.HS_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	RW	0x0	ss_upd_en Source status update enable. Source status information is fetched only from the location pointed to by the SSTARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI if SS_UPD_EN is high. NOTE: This enable is applicable only if DMAH_CHx_STAT_SRC is set to True.
37	RW	0x0	ds_upd_en Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTARx register, stored in the DSTATx register and written out to the DSTATx location of the LLI if DS_UPD_EN is high.

Bit	Attr	Reset Value	Description
36:34	RW	0x0	<p>protctl Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.</p> <p>Table 4 shows the mapping of bits in this field to the AHB HPROT[3:1] bus.</p>
33	RW	0x0	<p>fifo_mode FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced.</p> <p>1'b0: Space/data available for single AHB transfer of the specified transfer width.</p> <p>1'b1: Space/data available is greater than or equal to half the FIFO depth for destination transfers and less than half the FIFO depth for source transfers.</p> <p>The exceptions are at the end of a burst transaction request or at the end of a block transfer.</p>
32	RW	0x0	<p>fcmode Flow control mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller.</p> <p>1'b0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled</p> <p>1'b1: Source transaction requests are not serviced until a destination transaction request occurs</p> <p>In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.</p>
31	RW	0x0	<p>reload_dst Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs, refer to Table 5.</p>
30	RW	0x0	<p>reload_src Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. For conditions under which this occurs.</p>

Bit	Attr	Reset Value	Description
29:20	RW	0x000	max_abrst Maximum AMBA burst length. Maximum AMBA burst length that is used for DMA transfers on this channel. A value of 0 indicates that software is not limiting the maximum AMBA burst length for DMA transfers on this channel.
19	RW	0x0	src_hs_pol Source handshaking interface polarity. 1'b0: Active high 1'b1: Active low
18	RW	0x0	dst_hs_pol Destination handshaking interface polarity. 1'b0: Active high 1'b1: Active low
17	RW	0x0	lock_b Bus lock bit. When active, the AHB bus master signal block is asserted for the duration specified in CFGx.LOCK_B_L.
16	RW	0x0	lock_ch Channel lock bit. When the channel is granted control of the master bus interface and if the CFGx.LOCK_CH bit is asserted, then no other channels are granted control of the master bus interface for the duration specified in CFGx.LOCK_CH_L. Indicates to the master bus interface arbiter that this channel wants exclusive access to the master bus interface for the duration specified in CFGx.LOCK_CH_L.
15:14	RW	0x0	lock_b_l Bus lock level. Indicates the duration over which CFGx.LOCK_B bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction
13:12	RW	0x0	lock_ch_l Channel lock level. Indicates the duration over which CFGx.LOCK_CH bit applies. 2'b00: Over complete DMA transfer 2'b01: Over complete DMA block transfer 2'b1x: Over complete DMA transaction

Bit	Attr	Reset Value	Description
11	RW	0x1	<p>hs_sel_src Source software or hardware handshaking select. This register selects which of the handshaking interfaces hardware or software is active for source requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1'b1: Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.</p>
10	RW	0x1	<p>hs_sel_dst Destination software or hardware handshaking select. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel.</p> <p>1'b0: Hardware handshaking interface. Software-initiated transaction requests are ignored. 1'b1: Software handshaking interface. Hardware- initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.</p>
9	RO	0x0	<p>fifo_empty Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel.</p> <p>1'b1: Channel FIFO empty 1'b0: Channel FIFO not empty</p>
8	RW	0x0	<p>ch_susp Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data.</p> <p>1'b0: Not suspended. 1'b1: Suspend DMA transfer from the source</p>
7:5	RW	0x0	<p>ch_prior Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0: (DMAH_NUM_CHANNELS -1). A programmed value outside this range will cause erroneous behavior. Reset Value: Channel Number. for example: Chan0=0 Chan1=1.</p>
4:0	RO	0x0	reserved

DMA_SGR5

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	<p>sgc Source gather count. Source contiguous transfer count between successive gather boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits [31:b+1] do not exist and read back as 0.</p>
19:0	RW	0x00000	<p>sgi Source gather interval.</p>

DMA_DSR5

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:20	RW	0x000	<p>dsc Destination scatter count. Destination contiguous transfer count between successive scatter boundaries. $b = \log_2 (\text{DMAH_CHx_MAX_BLK_SIZE} + 1) + 19$ Bits 31:b+1 do not exist and read 0.</p>
19:0	RW	0x00000	<p>dsi Destination scatter interval.</p>

DMA_RAWTFR

Address: Operational Base + offset (0x02c0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>raw Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, RawTfr[2] is the Channel 2 raw transfer complete interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the ClearTfr register. 1'b0: Inactive Raw Interrupt Status 1'b1: active Raw Interrupt Status</p>

DMA_RAWBLOCK

Address: Operational Base + offset (0x02c8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>raw</p> <p>Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, RawBlock[2] is the Channel 2 raw block complete interrupt.</p> <p>Each bit in this register is cleared by writing a 1 to the corresponding location in the ClearBlock register.</p> <p>1'b0: Inactive Raw Interrupt Status 1'b1: active Raw Interrupt Status</p>

DMA RAWSRCTRAN

Address: Operational Base + offset (0x02d0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>raw</p> <p>Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, RawSrcTran[2] is the Channel 2 raw source transaction complete interrupt.</p> <p>Each bit in this register is cleared by writing a 1 to the corresponding location in the ClearSrcTran register.</p> <p>1'b0: Inactive Raw Interrupt Status 1'b1: active Raw Interrupt Status</p>

DMA RAWDSTTRAN

Address: Operational Base + offset (0x02d8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>raw</p> <p>Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, RawDstTran[2] is the Channel 2 raw destination transaction complete interrupt.</p> <p>Each bit in this register is cleared by writing a 1 to the corresponding location in the ClearDstTran register.</p> <p>1'b0: Inactive Raw Interrupt Status 1'b1: active Raw Interrupt Status</p>

DMA RAWERR

Address: Operational Base + offset (0x02e0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>raw Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated per channel; for example, RawErr[2] is the Channel 2 raw error interrupt.</p> <p>Each bit in this register is cleared by writing a 1 to the corresponding location in the ClearErr register.</p> <p>Note: Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.</p> <p>1'b0: Inactive Raw Interrupt Status 1'b1: active Raw Interrupt Status</p>

DMA STATUSTFR

Address: Operational Base + offset (0x02e8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>status Channel DMA Transfer complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, StatusTfr[2] is the Channel 2 source DMA transfer complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMAC.</p> <p>1'b0: Inactive Interrupt Status 1'b1: active Interrupt Status</p>

DMA STATUSBLOCK

Address: Operational Base + offset (0x02f0)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>status Channel block complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, StatusBlock[2] is the Channel 2 block complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMAC.</p> <p>1'b0: Inactive Interrupt Status 1'b1: active Interrupt Status</p>

DMA STATUSRCTRAN

Address: Operational Base + offset (0x02f8)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>status</p> <p>Channel Source Transaction complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, StatusSrcTran[2] is the Channel 2 source transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMAC.</p> <p>1'b0: Inactive Interrupt Status 1'b1: active Interrupt Status</p>

DMA STATUSDSTTRAN

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>status</p> <p>Channel destination transaction complete interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, StatusDstTran[2] is the Channel 2 status destination transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMAC.</p> <p>1'b0: Inactive Interrupt Status 1'b1: active Interrupt Status</p>

DMA STATUSERR

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RO	0x00	<p>status</p> <p>Channel Error interrupt event from all channels is stored in this Interrupt Status register after masking. This register has a bit allocated per channel; for example, StatusErr[2] is the Channel 2 status Error interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMAC.</p> <p>1'b0: Inactive Interrupt Status 1'b1: active Interrupt Status</p>

DMA MASKTFR

Address: Operational Base + offset (0x0310)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	int_mask_we Interrupt mask write enable. 1'b0: write disabled 1'b1: write enabled
7:6	RO	0x0	reserved
5:0	RW	0x00	int_mask The contents of the Raw Status register RawTfr is masked with the contents of the Mask register MaskTfr. Each bit of register is allocated per channel; for example, MaskTfr[2] is the mask bit for the Channel 2 transfer complete interrupt. 1'b0: masked 1'b1: unmasked

DMA MASKBLOCK

Address: Operational Base + offset (0x0318)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	int_mask_we Interrupt mask write enable. 1'b0: write disabled 1'b1: write enabled
7:6	RO	0x0	reserved
5:0	RW	0x00	int_mask The contents of the Raw Status register RawBlock is masked with the contents of the Mask register MaskBlock. Each bit of register is allocated per channel; for example, MaskBlock[2] is the mask bit for the Channel 2 block complete interrupt. 1'b0: masked 1'b1: unmasked

DMA MASKRCTRN

Address: Operational Base + offset (0x0320)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	int_mask_we Interrupt mask write enable. 1'b0: write disabled 1'b1: write enabled
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	<p>int_mask</p> <p>The contents of the Raw Status register RawSrcTran is masked with the contents of the Mask register MaskSrcTran. Each bit of register is allocated per channel; for example, MaskSrcTran[2] is the mask bit for the Channel 2 source transaction complete interrupt.</p> <p>1'b0: masked 1'b1: unmasked</p>

DMA MASKDSTTRAN

Address: Operational Base + offset (0x0328)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	<p>int_mask_we</p> <p>Interrupt mask write enable.</p> <p>1'b0: write disabled 1'b1: write enabled</p>
7:6	RO	0x0	reserved
5:0	RW	0x00	<p>int_mask</p> <p>The contents of the Raw Status register RawDstTran is masked with the contents of the Mask register MaskDstTran. Each bit of register is allocated per channel; for example, MaskDstTran[2] is the mask bit for the Channel 2 destination transaction complete interrupt.</p> <p>1'b0: masked 1'b1: unmasked</p>

DMA MASKERR

Address: Operational Base + offset (0x0330)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	<p>int_mask_we</p> <p>Interrupt mask write enable.</p> <p>1'b0: write disabled 1'b1: write enabled</p>
7:6	RO	0x0	reserved
5:0	RW	0x00	<p>int_mask</p> <p>The contents of the Raw Status register RawErr is masked with the contents of the Mask register MaskErr. Each bit of register is allocated per channel; for example, MaskErr[2] is the mask bit for the Channel 2 error interrupt.</p> <p>1'b0: masked 1'b1: unmasked</p>

DMA CLEARTFR

Address: Operational Base + offset (0x0338)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	WO	0x00	clear Each bit in the RawTfr and StatusTfr is cleared on the same cycle by writing a 1 to the corresponding location in the this registers. Each bit is allocated per channel; for example, ClearTfr[2] is the clear bit for the Channel 2 transfer done interrupt. Writing a 0 has no effect. This registers are not readable. 1'b0: no effect 1'b1: clear interrupt

DMA_CLEARBLOCK

Address: Operational Base + offset (0x0340)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	WO	0x00	clear Each bit in the RawBlock and StatusBlock is cleared on the same cycle by writing a 1 to the corresponding location in the this registers. Each bit is allocated per channel; for example, ClearBlock[2] is the clear bit for the Channel 2 block done interrupt. Writing a 0 has no effect. This registers are not readable. 1'b0: no effect 1'b1: clear interrupt

DMA_CLEARSRCTRAN

Address: Operational Base + offset (0x0348)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	WO	0x00	clear Each bit in the RawSrcTran and StatusSrcTran is cleared on the same cycle by writing a 1 to the corresponding location in the this registers. Each bit is allocated per channel; for example, ClearSrcTran[2] is the clear bit for the Channel 2 source transaction done interrupt. Writing a 0 has no effect. This registers are not readable. 1'b0: no effect 1'b1: clear interrupt

DMA_CLEARDSTTRAN

Address: Operational Base + offset (0x0350)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	WO	0x00	clear Each bit in the RawDstTran and StatusDstTran is cleared on the same cycle by writing a 1 to the corresponding location in the this registers. Each bit is allocated per channel; for example, ClearDstTran[2] is the clear bit for the Channel 2 destination transaction done interrupt. Writing a 0 has no effect. This registers are not readable. 1'b0: no effect 1'b1: clear interrupt

DMA CLEARERR

Address: Operational Base + offset (0x0358)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	WO	0x00	clear Each bit in the RawErr and StatusErr is cleared on the same cycle by writing a 1 to the corresponding location in the this registers. Each bit is allocated per channel; for example, ClearErr[2] is the clear bit for the Channel 2 error interrupt. Writing a 0 has no effect. This registers are not readable. 1'b0: no effect 1'b1: clear interrupt

DMA STATUSINT

Address: Operational Base + offset (0x0360)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	err OR of the contents of StatusErr register.
3	RO	0x0	dstt OR of the contents of StatusDst register.
2	RO	0x0	srct OR of the contents of StatusSrcTran register.
1	RO	0x0	block OR of the contents of StatusBlock register.
0	RO	0x0	tfr OR of the contents of StatusTfr register.

DMA DMACFGREG

Address: Operational Base + offset (0x0398)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	dma_en DMA enable bit. 1'b0: DMA Disabled 1'b1: DMA Enabled

DMA CHENREG

Address: Operational Base + offset (0x03a0)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:8	WO	0x00	ch_en_we Channel enable write enable. 1'b0: disable 1'b1: enable
7:6	RO	0x0	reserved
5:0	RW	0x00	ch_en Enables/Disables the channel. 1'b0: Disable the Channel 1'b1: Enable the Channel The ChEnReg.CH_EN bit is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.

11.4 Application Notes

11.4.1 Illegal Register Access

An illegal access can be any of the following:

- An AHB transfer of hsize greater than 32 is attempted.
- The hsel signal is asserted, but the address does not decode to a valid address.
- A write to the SARx, DARx, LLPx, CTLx, SSTATx, DSTATx, SSTATARx, DSTATARx, SGRx, or DSRx registers occurs when the channel is enabled.
- A read from the ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr registers is attempted.
- A write to the StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr registers is attempted.
- A write to the StatusInt register is attempted.
- A write to either the DmaIdReg or DMA Component ID Register register is attempted.

An error response is returned once there is an illegal access(read/write) occurs.

11.4.2 DMA Transfer Types

A DMA transfer may consist of single transfer or multi-block transfers. On successive blocks of a multi-block transfer, the SARx/DARx register in the DMAC is reprogrammed using either of the following methods:

- Block chaining using linked lists

- Auto-reloading
- Contiguous address between blocks

On successive blocks of a multi-block transfer, the CTLx register in the DMAC is reprogrammed using either of the following methods:

- Block chaining using linked lists
- Auto-reloading

When block chaining, using Linked Lists is the multi-block method of choice. On successive blocks, the LLPx register in the DMAC is reprogrammed using block chaining with linked lists.

A block descriptor consists of six registers: SARx, DARx, LLPx, CTLx, SSTATx, and DSTATx. The first four registers, along with the CFGx register, are used by the DMAC to set up and describe the block transfer.

Block Chaining Using Linked Lists

In this case, the DMAC reprograms the channel registers prior to the start of each block by fetching the block descriptor for that block from system memory. This is known as an LLI update.

DMAC block chaining uses a Linked List Pointer register (LLPx) that stores the address in memory of the next linked list item. Each LLI contains the corresponding block descriptors:

- SARx
- DARx
- LLPx
- CTLx
- SSTATx
- DSTATx

To set up block chaining, you need to program a sequence of Linked Lists in memory.

The SARx, DARx, LLPx, and CTLx registers are fetched from system memory on an LLI update. The updated contents of the CTLx, SSTATx, and DSTATx registers are written back to memory on block completion. Figure 2 shows the usage of chained linked lists in memory to define multi-block transfers using block chaining.

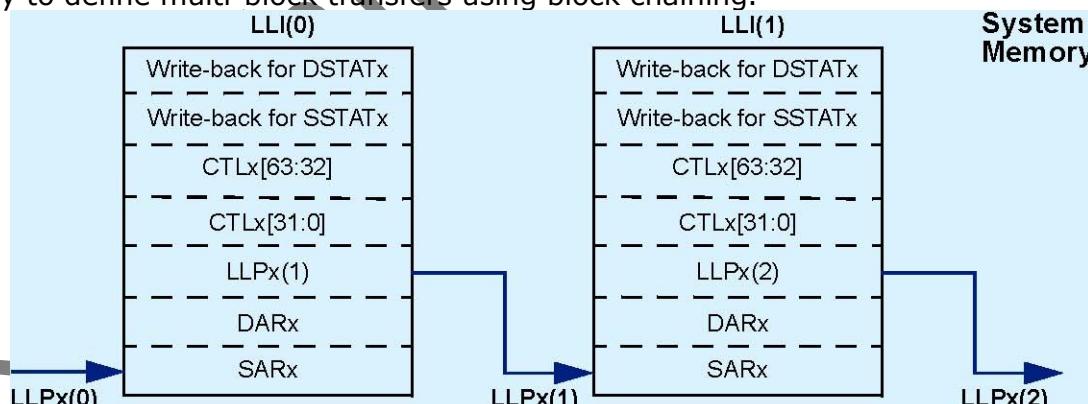


Fig. 11-2 Multi-Block Transfer Using Linked Lists When DMAH_CHx_STAT_SRC Set to True

In order not to confuse the SARx, DARx, LLPx, CTLx, STATx, and DSTATx register locations of the LLI with the corresponding DMAC memory mapped register locations, the LLI register locations are prefixed with LLI; that is, LLI.SARx, LLI.DARx, LLI.LLPx, LLI.CTLx, LLI.SSTATx, and LLI.DSTATx.

For rows 6 through 10 of following Table, the LLI.CTLx, LLI.LLPx, LLI.SARx, and LLI.DARx register locations of the LLI are always affected at the start of every block transfer. The LLI.LLPx and LLI.CTLx locations are always used to reprogram the DMAC LLPx and CTLx

registers. However, depending on the Table 5 row number, the LLI.SARx/LLI.DARx address may or may not be used to reprogram the DMAC SARx/DARx registers.

Table 11-1 Programming of Transfer Types and Channel Register Update Method

Transfer Type	LLP.LOC= 0	LLP_SRC_EN(CTLx)	RELOAD_SRC(CFGx)	RELOAD_DST(CTLx)	RELOAD_DST(CFGx)	CTLx, LLPxUpdateMethod	SARxUpdateMethod	DARxUpdateMethod	WriteBack
1. Single-block or last transfer of multi-block.	Yes	0	0	0	0	None, user reprograms	None (single)	None (single)	No
2. Auto-reload multi-block transfer with contiguous SAR	Yes	0	0	0	1	CTLx, LLPx are reloaded from initial values	Contiguous	Auto-reload	No
3. Auto-reload multi-block transfer with contiguous DAR.	Yes	0	1	0	0	CTLx, LLPx are reloaded from initial values	Auto-reload	Contiguous	No
4. Auto-reload multi-block transfer	Yes	0	1	0	1	CTLx, LLPx are reloaded from initial values	Auto-reload	Auto-reload	No
5. Single-block or last transfer of multi-block.	No	0	0	0	0	None, user reprograms	None (single)	None (single)	Yes
6. Linked list multi-block transfer with	No	0	0	1	0	CTLx, LLPx loaded from next LinkedList item.	Contiguous	Linked List	Yes

						SARxUpdateMethod	DARxUpdateMethod	WriteBack
						CTLx, LLPxUpdateMethod		
contiguous SAR						Auto-reload	Linked List	Yes
7. Linked list multi-block transfer with auto-reload SAR	No	0	1	1	0	CTLx, LLPx loaded from next LinkedList item	Linked List	
8. Linked list multi-block transfer with contiguous DAR	No	1	0	0	0	CTLx, LLPx loaded from next LinkedList item	Contiguous	Yes
9. Linked list multi-block transfer with auto-reload DAR	No	1	0	0	1	CTLx, LLPx loaded from next LinkedList item	Linked List	Auto-Reload
10. Linked list multi-block transfer	No	1	0	1	0	CTLx, LLPx loaded from next LinkedList item	Linked List	Linked List
Transfer Type								

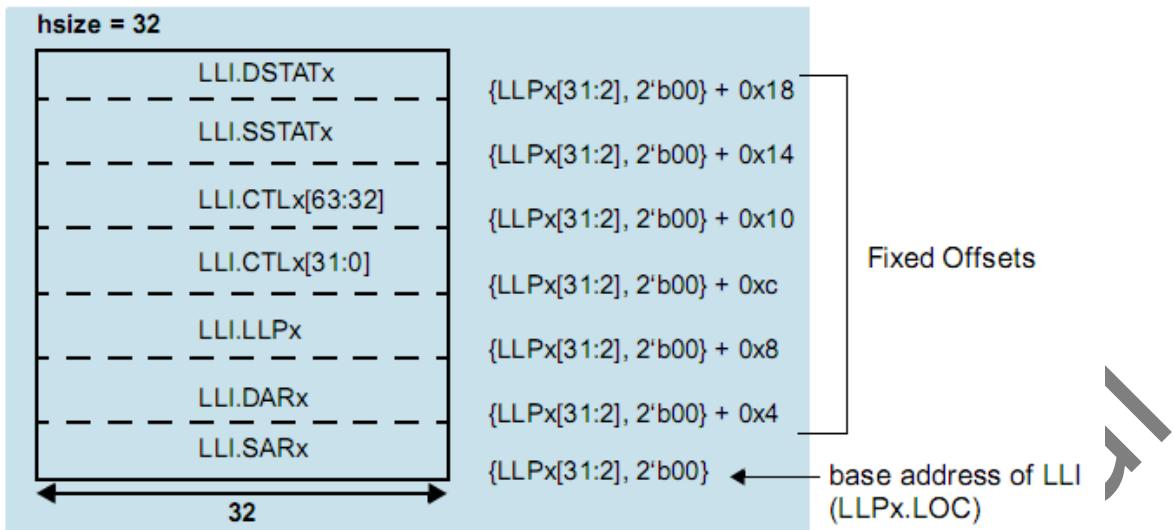


Fig. 11-3 Mapping of Block Descriptor (LLI) in Memory to Channel Registers

Rockchip Confidential

Chapter 12 Mailbox

12.1 Overview

The Mailbox module is a simple APB peripheral that allows Cortex-M4 to communicate with Cortex-M0 and HiFi3 by writing operation to generate interrupt. The registers are accessible via APB interface.

The Mailbox has the following main features:

- Support two Mailboxes for the communication
 - Mailbox0 between Cortex-M4 and Cortex-M0
 - Mailbox1 between Cortex-M4 and HiFi3
- Support APB interface
- Support four mailbox elements, each element includes one data word, one command word register and one flag bit that can represent one interrupt
- Support interrupts to Cortex-M4, Cortex-M0 and HiFi3
- Provide 32 lock registers for software to use to indicate whether mailbox is occupied

12.2 Block Diagram

The following figure shows the block diagram of MAILBOX:

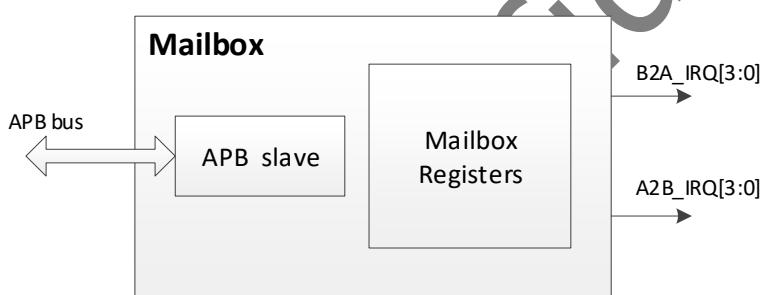


Fig 12-1 Block Diagram of Mailbox

12.3 Function Description

12.3.1 Mailbox0

- Regard Cortex-M4 as the “AP” side of the Mailbox0. The four interrupt sources (B2A0_0~B2A0_3) from Cortex-M0 to Cortex-M4 are:
 - Enabled when IRQ23~IRQ26 is enabled in the NVIC of the Cortex-M4 and MAILBOX0_B2A_INTEN[i] is set to 1. (i=0~3)
 - Generated when there are writing operation to corresponding MAILBOX0_B2A_CMD_i and MAILBOX0_B2A_DAT_i orderly.
 - Cleared when writing 1 to corresponding MAILBOX0_B2A_STATUS[i].
- Regard Cortex-M0 as the “BB” side of the Mailbox0. The four interrupt sources (A2B0_0~A2B0_3) from Cortex-M4 to Cortex-M0 are:
 - Enabled when IRQ14~IRQ17 is enabled in the NVIC of the Cortex-M0 and MAILBOX0_A2B_INTEN[i] is set to 1. (i=0~3)
 - Generated when there are writing operation to corresponding MAILBOX0_A2B_CMD_i and MAILBOX0_A2B_DAT_i orderly.
 - Cleared when writing 1 to corresponding MAILBOX0_A2B_STATUS[i].

12.3.2 Mailbox1

- Regard Cortex-M4 as the “AP” side of the Mailbox1. The four interrupt sources (B2A1_0~B2A1_3) from HiFi3 to Cortex-M4 are:

- Enabled when IRQ27~IRQ30 is enabled in the NVIC of the Cortex-M4 and MAILBOX1_B2A_INTEN[i] is set to 1. (i=0~3)
- Generated when there are writing operation to corresponding MAILBOX1_B2A_CMD_i and MAILBOX1_B2A_DAT_i orderly.
- Cleared when writing 1 to corresponding MAILBOX1_B2A_STATUS[i].
- Regard HiFi3 as the “BB” side of the Mailbox1. The four interrupt sources (A2B1_0~A2B1_3) from Cortex-M4 to HiFi3 are:
 - Enabled when IRQ14~IRQ17 is enabled in the INTC and MAILBOX1_A2B_INTEN[i] is set to 1. (i=0~3)
 - Generated when there are writing operation to corresponding MAILBOX1_A2B_CMD_i and MAILBOX1_A2B_DAT_i orderly.
 - Cleared when writing 1 to corresponding MAILBOX1_A2B_STATUS[i].

12.4 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 two Mailboxes (Mailbox0 ~ Mailbox1), and each of them has same register group. Therefore, two Mailboxes’ register groups have two different base addresses.

12.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
MAILBOX A2B INTEN	0x0000	W	0x00000000	AP to BB Interrupt Enable Register
MAILBOX A2B STATUS	0x0004	W	0x00000000	AP to BB Interrupt Status Register
MAILBOX A2B CMD 0	0x0008	W	0x00000000	AP to BB Command 0 Register
MAILBOX A2B DAT 0	0x000c	W	0x00000000	AP to BB Data 0 Register
MAILBOX A2B CMD 1	0x0010	W	0x00000000	AP to BB Command 1 Register
MAILBOX A2B DAT 1	0x0014	W	0x00000000	AP to BB Data 1 Register
MAILBOX A2B CMD 2	0x0018	W	0x00000000	AP to BB Command 2 Register
MAILBOX A2B DAT 2	0x001c	W	0x00000000	AP to BB Data 2 Register
MAILBOX A2B CMD 3	0x0020	W	0x00000000	AP to BB Command 3 Register
MAILBOX A2B DAT 3	0x0024	W	0x00000000	AP to BB Data 3 Register
MAILBOX B2A INTEN	0x0028	W	0x00000000	BB to AP Interrupt Enable Register
MAILBOX B2A STATUS	0x002c	W	0x00000000	BB to AP Interrupt Status Register
MAILBOX B2A CMD 0	0x0030	W	0x00000000	BB to AP Command 0 Register
MAILBOX B2A DAT 0	0x0034	W	0x00000000	BB to AP Data 0 Register
MAILBOX B2A CMD 1	0x0038	W	0x00000000	BB to AP Command 1 Register
MAILBOX B2A DAT 1	0x003c	W	0x00000000	BB to AP Data 1 Register
MAILBOX B2A CMD 2	0x0040	W	0x00000000	BB to AP Command 2 Register
MAILBOX B2A DAT 2	0x0044	W	0x00000000	BB to AP Data 2 Register
MAILBOX B2A CMD 3	0x0048	W	0x00000000	BB to AP Command 3 Register
MAILBOX B2A DAT 3	0x004c	W	0x00000000	BB to AP Data 3 Register
MAILBOX ATOMIC LOCK 00	0x0100	W	0x00000000	Atomic Lock 00 Register
MAILBOX ATOMIC LOCK 01	0x0104	W	0x00000000	Atomic Lock 01 Register

Name	Offset	Size	Reset Value	Description
MAILBOX ATOMIC LOCK 02	0x0108	W	0x00000000	Atomic Lock 02 Register
MAILBOX ATOMIC LOCK 03	0x010c	W	0x00000000	Atomic Lock 03 Register
MAILBOX ATOMIC LOCK 04	0x0110	W	0x00000000	Atomic Lock 04 Register
MAILBOX ATOMIC LOCK 05	0x0114	W	0x00000000	Atomic Lock 05 Register
MAILBOX ATOMIC LOCK 06	0x0118	W	0x00000000	Atomic Lock 06 Register
MAILBOX ATOMIC LOCK 07	0x011c	W	0x00000000	Atomic Lock 07 Register
MAILBOX ATOMIC LOCK 08	0x0120	W	0x00000000	Atomic Lock 08 Register
MAILBOX ATOMIC LOCK 09	0x0124	W	0x00000000	Atomic Lock 09 Register
MAILBOX ATOMIC LOCK 10	0x0128	W	0x00000000	Atomic Lock 10 Register
MAILBOX ATOMIC LOCK 11	0x012c	W	0x00000000	Atomic Lock 11 Register
MAILBOX ATOMIC LOCK 12	0x0130	W	0x00000000	Atomic Lock 12 Register
MAILBOX ATOMIC LOCK 13	0x0134	W	0x00000000	Atomic Lock 13 Register
MAILBOX ATOMIC LOCK 14	0x0138	W	0x00000000	Atomic Lock 14 Register
MAILBOX ATOMIC LOCK 15	0x013c	W	0x00000000	Atomic Lock 15 Register
MAILBOX ATOMIC LOCK 16	0x0140	W	0x00000000	Atomic Lock 16 Register
MAILBOX ATOMIC LOCK 17	0x0144	W	0x00000000	Atomic Lock 17 Register
MAILBOX ATOMIC LOCK 18	0x0148	W	0x00000000	Atomic Lock 18 Register
MAILBOX ATOMIC LOCK 19	0x014c	W	0x00000000	Atomic Lock 19 Register
MAILBOX ATOMIC LOCK 20	0x0150	W	0x00000000	Atomic Lock 20 Register
MAILBOX ATOMIC LOCK 21	0x0154	W	0x00000000	Atomic Lock 21 Register
MAILBOX ATOMIC LOCK 22	0x0158	W	0x00000000	Atomic Lock 22 Register
MAILBOX ATOMIC LOCK 23	0x015c	W	0x00000000	Atomic Lock 23 Register

Name	Offset	Size	Reset Value	Description
MAILBOX ATOMIC LOCK 24	0x0160	W	0x00000000	Atomic Lock 24 Register
MAILBOX ATOMIC LOCK 25	0x0164	W	0x00000000	Atomic Lock 25 Register
MAILBOX ATOMIC LOCK 26	0x0168	W	0x00000000	Atomic Lock 26 Register
MAILBOX ATOMIC LOCK 27	0x016c	W	0x00000000	Atomic Lock 27 Register
MAILBOX ATOMIC LOCK 28	0x0170	W	0x00000000	Atomic Lock 28 Register
MAILBOX ATOMIC LOCK 29	0x0174	W	0x00000000	Atomic Lock 29 Register
MAILBOX ATOMIC LOCK 30	0x0178	W	0x00000000	Atomic Lock 30 Register
MAILBOX ATOMIC LOCK 31	0x017c	W	0x00000000	Atomic Lock 31 Register

Notes: **S**ize: **B**- Byte (8 bits) access, **H**W- Half WORD (16 bits) access, **W**-WORD (32 bits) access

12.4.2 Detail Register Description

MAILBOX A2B INTEN

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX A2B STATUS

Address: Operational Base + offset (0x0004)

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

Bit	Attr	Reset Value	Description
3	W1 C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1 C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
1	W1 C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1 C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX A2B CMD 0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX A2B DAT 0

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX A2B CMD 1

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX A2B DAT 1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX A2B CMD 2

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX A2B DAT 2

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX A2B CMD 3

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX A2B DAT 3

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX B2A INTEN

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	int3 Interrupt enable for int3. 1'b0: Disable 1'b1: Enable
2	RW	0x0	int2 Interrupt enable for int2. 1'b0: Disable 1'b1: Enable
1	RW	0x0	int1 Interrupt enable for int1. 1'b0: Disable 1'b1: Enable
0	RW	0x0	int0 Interrupt enable for int0. 1'b0: Disable 1'b1: Enable

MAILBOX B2A STATUS

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	W1 C	0x0	int3 Interrupt status for int3. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
2	W1 C	0x0	int2 Interrupt status for int2. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
1	W1 C	0x0	int1 Interrupt status for int1. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active
0	W1 C	0x0	int0 Interrupt status for int0. Clear the interrupt by writing 1 to this bit. 1'b0: Interrupt is inactive 1'b1: Interrupt is active

MAILBOX B2A CMD 0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX B2A DAT 0

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX B2A CMD 1

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX B2A DAT 1

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX B2A CMD 2

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX B2A DAT 2

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX B2A CMD 3

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	command Command register.

MAILBOX B2A DAT 3

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	data Data register.

MAILBOX ATOMIC LOCK 00

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 01

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 02

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 03

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 04

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 05

Address: Operational Base + offset (0x0114)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 06

Address: Operational Base + offset (0x0118)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 07

Address: Operational Base + offset (0x011c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 08

Address: Operational Base + offset (0x0120)

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

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 09

Address: Operational Base + offset (0x0124)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 10

Address: Operational Base + offset (0x0128)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 11

Address: Operational Base + offset (0x012c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 12

Address: Operational Base + offset (0x0130)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 13

Address: Operational Base + offset (0x0134)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 14

Address: Operational Base + offset (0x0138)

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

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 15

Address: Operational Base + offset (0x013c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 16

Address: Operational Base + offset (0x0140)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 17

Address: Operational Base + offset (0x0144)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 18

Address: Operational Base + offset (0x0148)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 19

Address: Operational Base + offset (0x014c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 20

Address: Operational Base + offset (0x0150)

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

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 21

Address: Operational Base + offset (0x0154)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 22

Address: Operational Base + offset (0x0158)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 23

Address: Operational Base + offset (0x015c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 24

Address: Operational Base + offset (0x0160)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 25

Address: Operational Base + offset (0x0164)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 26

Address: Operational Base + offset (0x0168)

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

Bit	Attr	Reset Value	Description
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 27

Address: Operational Base + offset (0x016c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 28

Address: Operational Base + offset (0x0170)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 29

Address: Operational Base + offset (0x0174)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 30

Address: Operational Base + offset (0x0178)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

MAILBOX ATOMIC LOCK 31

Address: Operational Base + offset (0x017c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	atomic_lock Atomic lock flag bit.

12.5 Application Notes

- It is recommended to read one ATOMIC_LOCK register first when using the Mailbox. The Mailbox is available only when the read value is 0. Writing to the ATOMIC_LOCK register will clear this bit to 0.

- Write to the CMD register before writing to the DAT register. If wrong order is used, then the interrupt cannot be generated successfully.
- If you want to clear the interrupt, you can read out the STATUS register and writing 1 to corresponding bit.

Rockchip Confidential

Chapter 13 Internal SRAM

13.1 Overview

There are two Internal SRAMs in RK2206, named System SRAM0 and System SRAM1. System SRAM0 and System SRAM1 are AHB slave devices, which support read and write access to provide system fast access data storage. RK2206 also has a buffer in wlan subsystem.

Specially, there are DSP ITCM and DSP DTCM, which can be used not only by dsp, but also by other master.

13.1.1 Features supported

- System SRAM0
 - Support 128KB capacity
- System SRAM1
 - Support 128KB capacity
 - Share space with VAD
- Buffer
 - Support 96KB capacity
- DSP ITCM/DTCM
 - Support 32KB ITCM and 192KB DTCM
 - Share space with other master

13.2 Block Diagram

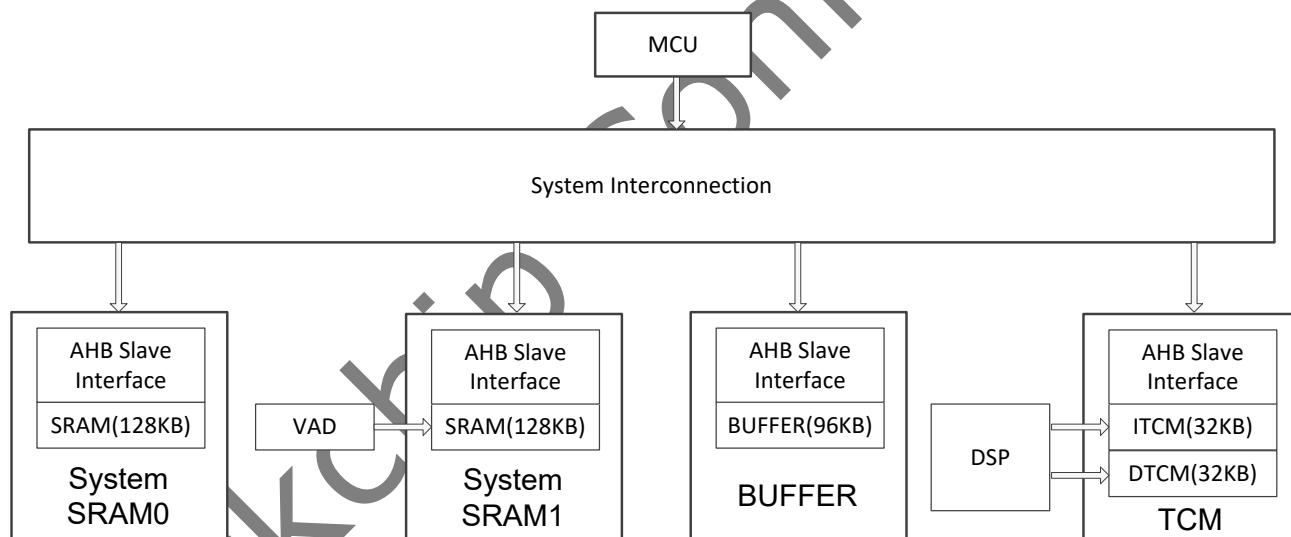


Fig. 13-1 System SRAM0/1 BUFFER TCM block diagram

13.3 Function Description

13.3.1 AHB slave interface of System SRAM0/SRAM1

The AHB slave interface are bridges which translate AHB bus access to System SRAM0, System SRAM1, BUFFER, TCM interface.

13.3.2 System SRAM0 access path

The System SRAM0 can only be accessed by Cortex M4F Ibus, Cortex M4F Dbus, Cortex M4F Sbus, Cortex M0, DSP, FSPIO, DMAC, WLAN, VIP, VOP, CRYPTO, USB2OTG, SD/MMC, SPI2APB, and VAD.

13.3.3 System SRAM1 access path

The System SRAM1 can only be accessed by Cortex M4F Ibus, Cortex M4F Dbus, Cortex M4F Sbus, Cortex M0, DSP, FSPIO, DMAC, WLAN, VIP, VOP, CRYPTO, USB2OTG, SD/MMC, SPI2APB, and VAD. VAD can write System SRAM1 directly. Also, VAD can read or write System SRAM1 through system interconnect.

13.3.4 BUFFER access path

Buffer can only be accessed by M0, WLAN and DMAC.

13.3.5 DSP ITCM and DTCM access path

DSP ITCM and DTCM can only be accessed by Cortex M4F Ibus, Cortex M4F Dbus, Cortex M4F Sbus, Cortex M0, DSP, FSPIO, DMAC, WLAN, VIP, VOP, CRYPTO, USB2OTG, SD/MMC, SPI2APB, and VAD. DSP accessed ITCM and DTCM directly, and other master access ITCM and DTCM through system interconnection.

Rockchip Confidential

Chapter 14 Voice Activity Detect (VAD)

14.1 Overview

Voice Activity Detect (VAD) is used to detect the amplitude of voice which is received by Analog Mic, I2S Digital Mic or PDM digital Mic when SoC is in low power mode. If the amplitude of voice is over threshold, the VAD will assert interrupt to wake up SoC, then SoC will exit low power mode.

VAD supports the following features:

- Support AHB bus interface
- Support read voice data from I2S0, I2S1,PDM
 - Support to configure the voice source address
 - Support to configure increment or fixed for the direction of voice data address
 - Support DMA request and acknowledge
 - Support transfer 1~8 burst per DMA request
 - Support read 1~8 Mic voice data, and only support single Mic voice detection, user can select any Mic voice data to detect the amplitude of voice
 - Support 16/24 bits voice data
- Support voice amplitude detection
 - Support an Amplifier for the voice data
 - Support a IIR high pass filter for the voice frequency band, and the filter coefficient can be configured
 - Support a voice detect threshold that take the ambient noise to account
- Support Multi-Mic array data storing
 - Buffer memory is shared with Internal SRAM
 - The start and end address of storing can be configured
 - When current storing address is up to end address, it will loop to start address and overlap previous data, it will also assert a flag
 - Support 3 data storing mode: mode 0 start storing data after the voice detect event, mode 1 start storing after VAD is enabled and mode 2 do not storing data
 - Support storing data through bus or ram write interface
- Support a level combined interrupt
 - Support voice detect interrupt
 - Support time out interrupt
 - Support transfer error interrupt
 - Support data transfer interrupt

14.2 Block Diagram

VAD comprises with:

- ahb_master: AHB Master Interface
- ahb_sram_if: AHB Slave Interface
- vad_reg_bank : Register bank
- dmac_engine: DMA control engine
- vad_det : Voice detection

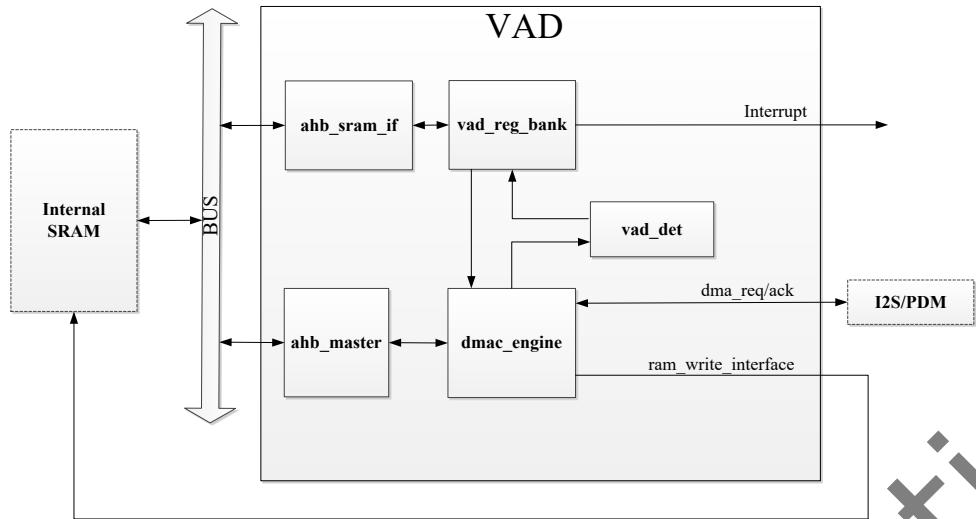


Fig. 14-1 VAD Block Diagram

14.3 Function Description

14.3.1 DMAC_ENGINE

dmac_engine is used to read voice data from one of I2S0, I2S1 or PDM, and it can store all channels data to Internal SRAM. If the bus can be access, user can configure to store through bus. Otherwise user should configure to store through RAM write interface. When VAD is working, user can change the data storing mode dynamically. The storing address can be continuous transition or be scattered.

The voice data can be 16 or 24 bits:

- When it's 16 bits, it must be half word transfer mode that low 16 bits in a word for left channel and high 16 bits in a word for right channel.
- When it's 24bits, it must be word transfer mode that only 24 bits data is valid in a word, and it support left or right justified

dmac_engine also selects and sends one channel data to vad_det for voice detection.

vad_det only support 16 bits data to detect the amplitude of voice, so when the voice data is 24 bits, user can use the high or low 16 bits in 24 bits.

- When use high bits, the data value will be divided by 256.
- When use low bits, the data value will be saturation to 16 bits.

14.3.2 VAD_DET

vad_det is used to detect the amplitude of voice.

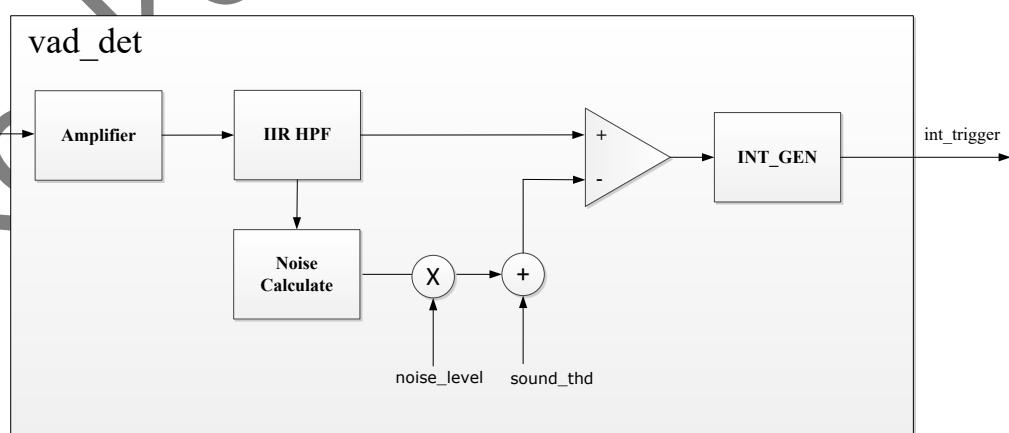


Fig. 14-2 vad_det Block Diagram

Amplifier

$\text{voice_amplitude_amplified} = \text{gain} * \text{voice_amplitude_original} / 8$.

IIR HPF

There is a high pass filter for the human voice frequency band, the filter is a two order direct I type IIR. As the following formula describes:

$$y(n) = -a1*y(n-1) - a2*y(n-2) + b0*x(n) + b1*x(n-1) + b2*x(n-2)$$

The coefficient $a1$, $a2$, $b0$, $b1$ and $b2$ are all quantified by multiplying 16384 and represented as 16 bits, the result is follow registers that can be configured: `iir_anum_0`, `iir_anum_1`, `iir_anum_2`, `iir_aden_1` and `iir_aden_2`.

The output of HPF need some time to achieve convergence after VAD is enabled.

Noise Calculate

VAD supports a voice detection threshold that take the ambient noise to account:

- VAD calculates the average amplitude of voice data within `noise_sample_num` samples, the result is regard as the noise value of one frame. The noise value of last 128 frames also can be configured directly.
- VAD finds the minimum noise value within `noise_frm_num` frames, the result is regard as the `noise_min`(minimum noise value).User can configure `min_noise_find_mode` to change the mode to find the minimum noise.
- The `noise_min` will be smooth updated to `noise_abs`(current noise value), As the following formula describes: $noise_abs = (noise_abs * noise_alpha + noise_min * (256 - noise_alpha)) / 256$. `noise_abs` will be updated once every frame. `noise_abs` also can be configured directly, and it is not clear until VAD is reset.

Voice Detect Threshold

The final threshold is `sound_thd + noise_abs * noise_level`.

INT_GEN

VAD support 3 modes to assert the voice detection interrupt.

- Normal mode: when equal or more than a number (`vad_con_thd`) of continuous samples over the threshold, the voice detection interrupt will be asserted. The `vad_con_thd` can be configured by register.
- Allow an exception mode: base on normal mode, it can be configured to allow exceptions during continuous sample judgment. The exception number can be configured by register. When the exceptions is more, the voice detect condition is less strict.
- Accumulating mode: A counter is used for accumulating, the counter will plus 1 when current sample is over threshold (it will not plus when it reach maximum value 256), the counter will minus 1 when current sample is not over threshold (it will not minus when it reach 0); When the counter is equal to or more than a number (`vad_con_thd`), the voice detection interrupt will be asserted. Compare with normal mode, the voice detect condition is less strict when use the same value of `vad_con_thd`.

14.4 Register Description

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
VAD CONTROL	0x0000	W	0x03000000	Control register
VAD VS ADDR	0x0004	W	0x00000000	Voice source address register
VAD TIMEOUT	0x004c	W	0x00000000	Timeout register
VAD RAM START ADDR	0x0050	W	0x00000000	RAM start address register
VAD RAM END ADDR	0x0054	W	0x00000000	RAM end address register
VAD RAM CUR ADDR	0x0058	W	0x00000000	RAM current address register
VAD DET CON0	0x005c	W	0x01024008	Detect control register0
VAD DET CON1	0x0060	W	0x04ff0064	Detect control register1
VAD DET CON2	0x0064	W	0x3bf5e663	Detect control register2
VAD DET CON3	0x0068	W	0x3bf58817	Detect control register3
VAD DET CON4	0x006c	W	0x382b8858	Detect control register4
VAD DET CON5	0x0070	W	0x00000000	Detect control register5

Name	Offset	Size	Reset Value	Description
VAD_INT	0x0074	W	0x00000000	VAD interrupt register
VAD_AUX_CON0	0x0078	W	0x00000000	Auxiliary control register0
VAD_SAMPLE_CNT	0x007c	W	0x00000000	Sample counter register
VAD_RAM_START_ADDR_BUS	0x0080	W	0x00000000	RAM start address register for bus write mode
VAD_RAM_END_ADDR_B_US	0x0084	W	0x00000000	RAM end address register for bus write mode
VAD_RAM_CUR_ADDR_BU_S	0x0088	W	0x00000000	RAM current address register for bus write mode
VAD_AUX_CON1	0x008c	W	0x00000000	Auxiliary control register1
VAD_NOISE_FIRST_DATA	0x0100	W	0x00000000	Noise first data register
VAD_NOISE_LAST_DATA	0x02fc	W	0x00000000	Noise last data register

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

14.4.2 Detail Register Description

VAD CONTROL

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:29	RW	0x0	vad_det_channel Index of the channel for voice detect, from channel 0 to channel 7.
28	RW	0x0	voice_24bit_sat The mode of voice 24bit data change to 16bit 1'b0: Get the high 16bit data(divided by 256) 1'b1: Saturation from 24bit to 16bit
27	RW	0x0	voice_24bit_align_mode Align mode of channel 24bit width 1'b0: 8~31bits is valid 1'b1: 0~23bits is valid
26	RW	0x0	voice_channel_bitwidth 1'b0: 16bits 1'b1: 24bits
25:23	RW	0x6	voice_channel_num Voice channel number, the value N means N+1 channel.
22	RO	0x0	reserved
21:20	RW	0x0	vad_mode 2'h0: Begin to store the data after voice detect 2'h1: Begin to store the data after VAD is enable 2'h2: Don't store the data 2'h3: Reserved
19:15	RO	0x0	reserved
14	RW	0x0	source_fixaddr_en Direction of source address 1'b0: Increment 1'b1: Fixed

Bit	Attr	Reset Value	Description
13:10	RW	0x0	incr_length INCR burst length, 0~15 is valid. It is valid when source_burst is set to 3'h1.
9:7	RW	0x0	source_burst_num Source burst number per dma_req, the value N means N+1 burst.
6:4	RW	0x0	source_burst 3'h0: SINGLE 3'h1: INCR 3'h3: INCR4 3'h5: INCR8 3'h7: INCR16 Others: Reserved
3:1	RW	0x0	source_select Voice source select 3'h0: sourc0 3'h1: sourc1 3'h2: sourc2 3'h3: sourc3 3'h4: sourc4 Others: Reserved
0	RW	0x0	vad_en VAD enable 1'b0: Disable 1'b1: Enable

VAD VS ADDR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vs_addr Voice source address

VAD TIMEOUT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31	RW	0x0	work_timeout_en Work timeout enable 1'b0: Disable 1'b1: Enable
30	RW	0x0	idle_timeout_en Idle timeout enable 1'b0: Disable 1'b1: Enable
29:20	RW	0x000	work_timeout_thd work timeout threshold, the unit is one cycle of hclk.

Bit	Attr	Reset Value	Description
19:0	RW	0x000000	idle_timeout_thd Idle timeout threshold, the unit is one cycle of hclk.

VAD RAM START ADDR

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_start_addr RAM start address to store voice data, the address must be double word alignment.

VAD RAM END ADDR

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_end_addr RAM end address to store voice data, the address must be double word alignment.

VAD RAM CUR ADDR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_cur_addr RAM current address to store voice data, The last valid double word data is at address ram_cur_addr-0x8. When the ram_loop_flag is valid, the valid voice data will be ram_cur_addr ~ ram_end_addr ~ loop to ram_begin_addr ~ ram_cur_addr-0x8. When the ramp_loop_flag is not valid, the valid voice data will be ram_begin_addr ~ ram_cur_addr-0x8.

VAD DET CON0

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:28	RW	0x0	vad_thd_mode Threshold mode for vad_con_thd 2'b00: Normal mode 2'b01: Allow an exception mode 2'b10: Accumulating mode 2'b11: reserved
27:24	RW	0x1	dis_vad_con_thd In the determining of continuous sample number exceed threshold, allow some number of sample as an exception. It's valid only when vad_thd_mode=1. When this value is lower, the voice detect condition is more strict.

Bit	Attr	Reset Value	Description
23:16	RW	0x02	vad_con_thd When continuous sample number(>=vad_con_thd) exceed threshold, then assert the vad_det interrupt, the value N means N+1. When this value is higher, the voice detect condition is more strict.
15	RO	0x0	reserved
14:12	RW	0x4	noise_level Noise level, valid value is 0x1~0x6. When this value is higher, the voice detect condition is more strict.
11:0	RW	0x008	gain The gain control of voice data amplifier, the value of gain is unsigned and is valid from 0 to 4095. voice_amplitude_amplified=gain*voice_amplitude_original/8.

VAD DET CON1

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	min_noise_find_mode Minimal noise value find mode 1'b0: Always find the value at the range of noise_frm_num. 1'b1: When receive N frame, if N is less than noise_frm_num, find the value at the range of N; if N is more than noise_frm_num, find the value at the range of noise_frm_num.
29	RW	0x0	clean_noise_at_begin 1'b0: The noise will be clean only at the begin of the first time VAD is enable after reset. 1'b1: The noise will be clean every time at the begin of VAD is enable.
28	RW	0x0	force_noise_clk_en Force noise calculate clk enable 1'b0: The clock will be auto gating for low power 1'b1: The clock will be always enable
27	RO	0x0	reserved
26	RW	0x1	clean_iir_en Clean IIR filter when VAD is disable 1'b0: Not clean 1'b1: Clean
25:16	RW	0x0ff	noise_sample_num The number of sample in one frame to calculate the noise, the value N means N+1 sample. When this value is higher, the voice detect condition is more strict.

Bit	Attr	Reset Value	Description
15:0	RW	0x0064	sound_thd Initial sound threshold when this value is higher, the voice detect condition is more strict.

VAD DET CON2

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	RW	0x3bf5	iir_anum_0 IIR numerator coefficient b0
15:8	RW	0xe6	noise_alpha The update smooth speed of noise When this value is lower, the voice detect condition is more strict.
7	RO	0x0	reserved
6:0	RW	0x63	noise_frm_num The number of frame to calculate the noise, the value N means N+1 frame. When this value is lower, the voice detect condition is more strict.

VAD DET CON3

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:16	RW	0x3bf5	iir_anum_2 IIR numerator coefficient b2
15:0	RW	0x8817	iir_anum_1 IIR numerator coefficient b1

VAD DET CON4

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31:16	RW	0x382b	iir_aden_2 IIR denominator coefficient a2
15:0	RW	0x8858	iir_aden_1 IIR denominator coefficient a1

VAD DET CON5

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	iir_result Voice real time data after IIR filter
15:0	RW	0x0000	noise_abs Noise abs value

VAD INT

Address: Operational Base + offset (0x0074)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	ramp_loop_flag_bus RAM address loop flag for AHB bus interface write mode. Only valid when bus_write_addr_mode=1'b1. 1'b0: not loop 1'b1: loop
11	W1C	0x0	vad_data_trans_int VAD data transfer interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
10	RW	0x0	vad_data_trans_int_en VAD data transfer interrupt enable 1'b0: Disable 1'b1: Enable
9	RW	0x0	vad_idle VAD idle flag 1'b0: Not idle 1'b1: Idle
8	RO	0x0	ramp_loop_flag RAM address loop flag 1'b0: not loop 1'b1: loop
7	W1C	0x0	work_timeout_int Work timeout interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
6	W1C	0x0	idle_timeout_int Idle timeout interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
5	RW	0x0	error_int Error interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
4	W1C	0x0	vad_det_int VAD detect interrupt 1'b0: interrupt not generated 1'b1: interrupt generated
3	RW	0x0	work_timeout_int_en Work timeout interrupt enable 1'b0: Disable 1'b1: Enable

Bit	Attr	Reset Value	Description
2	RW	0x0	idle_timeout_int_en Idle timeout interrupt enable 1'b0: Disable 1'b1: Enable
1	RW	0x0	error_int_en Error interrupt enable 1'b0: Disable 1'b1: Enable
0	RW	0x0	vad_det_int_en VAD detect interrupt enable 1'b0: Disable 1'b1: Enable

VAD_AUX CON0

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	sample_cnt_en Sample counter enable 1'b0: Disable 1'b1: Enable
28	RW	0x0	int_trig_ctrl_en The VAD detection interrupt trigger control enable. 1'b0: Disable, the VAD detection interrupt is always triggered 1'b1: Enable, the VAD detection interrupt trigger is controlled by int_trig_valid_thd.
27:16	RW	0x000	int_trig_valid_thd VAD detection interrupt trigger valid threshold. The VAD detection interrupt will be triggered valid after sample_cnt exceed int_trig_valid_thd. The value N means N+1, The unit is one voice sample point.
15	RO	0x0	reserved
14	RW	0x0	ram_write_rework_addr_mode The rework address for RAM interface write mode. 1'b0: Store the data from the current address 1'b1: Store the data from the start address
13	RW	0x0	bus_write_rework_addr_mode The rework address for AHB bus interface write mode. 1'b0: Store the data from the current address 1'b1: Store the data from the start address

Bit	Attr	Reset Value	Description
12	RW	0x0	bus_write_addr_mode The address selection when use AHB bus interface write mode. 1'b0: Use RAM_START_ADDR, RAM_END_ADDR, RAM_CUR_ADDR(same with RAM interface write mode). The internal address will continuous when dynamic change between AHB bus interface write mode and RAM interface write mode. 1'b1: Use RAM_START_ADDR_BUS, RAM_END_ADDR_BUS and RAM_CUR_ADDR_BUS.
11:4	RW	0x00	data_trans_kbyte_thd Data transfer number threshold, the unit is KByte. The value N means N+1 KByte. The interrupt is generated per data_trans_kbyte_thd+1 KBytes.
3	RO	0x0	reserved
2	RW	0x0	data_trans_trig_int_en Trigger an interrupt for data transfer , It's valid only when bus_write_en=1'b1. 1'b0: Disable 1'b1: Enable
1	RW	0x0	dis_ram_itf Disable write voice data to Internal SRAM through RAM interface. 1'b0: Enable ram interface 1'b1: Disable ram interface
0	RW	0x0	bus_write_en Enable write voice data to Internal SRAM through AHB bus interface. 1'b0: Disable 1'b1: Enable

VAD SAMPLE CNT

Address: Operational Base + offset (0x007c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	sample_cnt sample counter

VAD RAM START ADDR BUS

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_start_addr_bus RAM start address to store voice data, the address must be double word alignment. Only used for AHB bus interface write mode and when bus_write_addr_mode=1'b1.

VAD RAM END ADDR BUS

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_begin_addr_bus RAM start address to store voice data, the address must be double word alignment. Only used for AHB bus interface write mode and when bus_write_addr_mode=1'b1.

VAD RAM CUR ADDR BUS

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ram_cur_addr_bus RAM current address to store voice data. Only used for AHB bus interface write mode and when bus_write_addr_mode=1'b1. The last valid double word data is at address ram_cur_addr_bus-0x8. When the ram_loop_flag_bus is valid, the valid voice data will be ram_cur_addr_bus ~ ram_end_addr_bus ~ loop to ram_begin_addr_bus ~ ram_cur_addr_bus-0x8. When the ramp_loop_flag is not valid, the valid voice data will be ram_begin_addr_bus ~ ram_cur_addr_bus-0x8.

VAD AUX CON1

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	data_trans_int_mode_sel Data transfer number threshold selection for interrupt trigger 1'b0: data_trans_kbyte_thd 1'b1: data_trans_word_thd
15:0	RW	0x0000	data_trans_word_thd Data transfer number threshold, the unit is word. The value N means N+1 words. The interrupt is generated per trans_word_thd+1 words.

VAD NOISE FIRST DATA

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	noise_first_data Noise first data

VAD NOISE LAST DATA

Address: Operational Base + offset (0x02fc)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	noise_last_data Noise last data

14.5 Application Notes

14.5.1 VAD usage flow

VAD usage flow is as following figure.

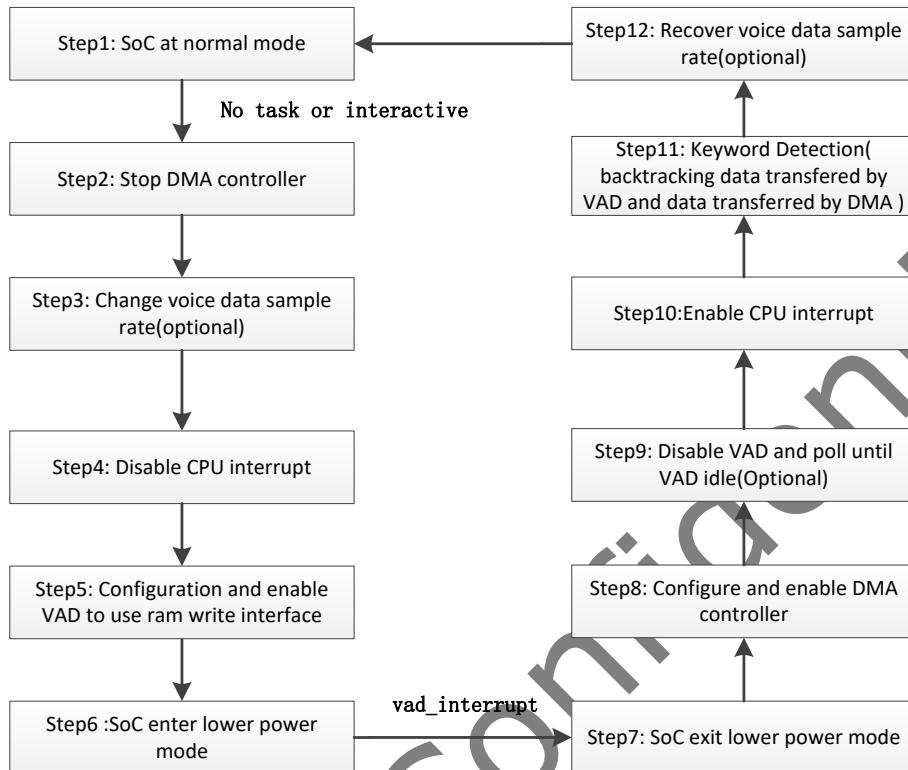


Fig. 14-3 VAD usage flow (only used ram write interface mode)

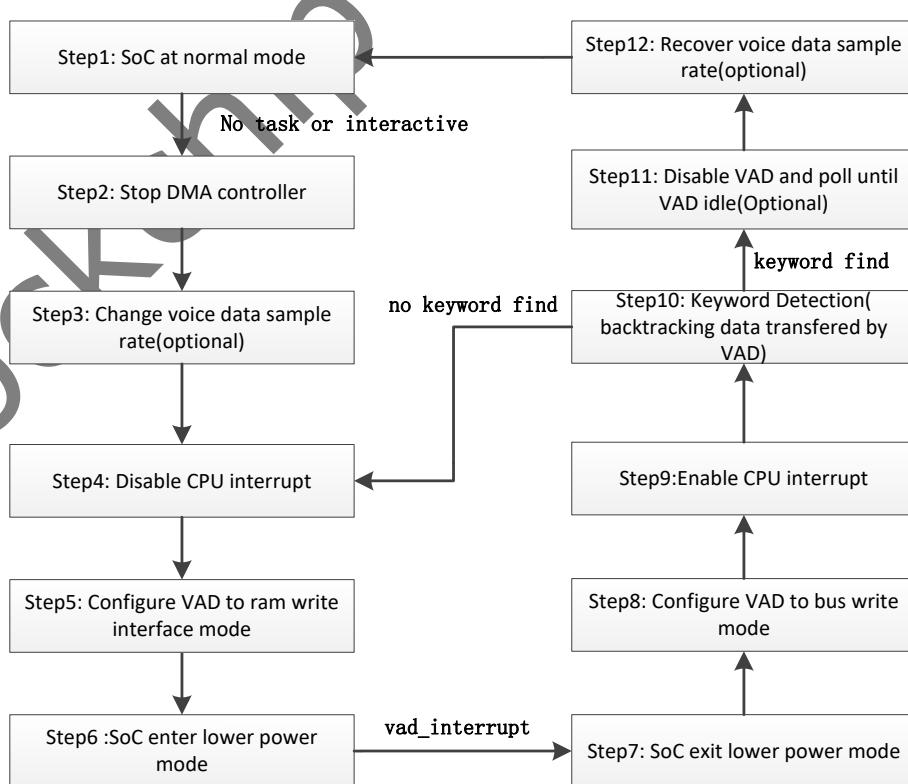


Fig. 14-4 VAD usage flow (use bus mode and ram write interface mode)

- Step3, step12 are optional, user should consider the power consumption and keyword detection accuracy for these steps.
- Disable VAD step is optional, user can keep VAD working, and use it as a DMA Controller.

14.5.2 VAD configuration usage flow

1. Set VAD_VS_ADDR.vs_addr=I2S0 base address + I2S_INCR_RXDR.
2. Set VAD_RAM_BEGIN_ADDR.ram_begin_addr and VAD_RAM_END_ADDR.ram_end_addr, the address should match with the Internal SRAM sharing scheme.
3. Set to bus write mode or ram interface mode when enable VAD
Set VAD_AUX_CON0.bus_write_en=0, and VAD_AUX_CON0.dis_ram_itf =0 to enable VAD as ram interface write mode.
Set VAD_AUX_CON0.bus_write_en=1, and VAD_AUX_CON0.dis_ram_itf =1 to enable VAD as bus write mode.
4. Adjust the sensitivity of voice activity detect by setting follow registers:
VAD_DET_CON0.noise_level
VAD_DET_CON0.vad_con_thd
VAD_DET_CON0.dis_vad_con_thd
VAD_DET_CON0.vad_thd_mode
VAD_DET_CON1.noise_sample_num
VAD_DET_CON1.sound_thd
VAD_DET_CON2.noise_frm_num
VAD_DET_CON2.noise_alpha
5. Set the iir_anum_0~3 and iir_aden_1~2 to adjust the IIR HPF coefficient.

For 48Khz sample rate:

iir_anum_0: 0x382d
iir_anum_1: 0x8fa5
iir_anum_2: 0x382d
iir_aden_1: 0x909b
iir_aden_2: 0x3150

For 16Khz sample rate (default):

iir_anum_0: 0x3bf5
iir_anum_1: 0x8817
iir_anum_2: 0x3bf5
iir_aden_1: 0x8858
iir_aden_2: 0x382b

For 8Khz sample rate:

iir_anum_0: 0x3e9f
iir_anum_1: 0x82c2
iir_anum_2: 0x3e9f
iir_aden_1: 0x82c9
iir_aden_2: 0x3d46

6. Set DET_CON5.noise_abs to ambient noise which is calculated by software.

Set VAD_NOISE_DATA+offset to initial the noise data of all frames. The frame number is VAD_DET_CON2.noise_frm_num. The first frame noise data address is VAD_NOISE_DATA, the second frame noise data address is VAD_NOISE_DATA+0x4, and so on.

7. Set VAD_INT.vad_det_int_en=0x1 to enable the interrupt.

8. Set VAD_AUX_CONTROL to disable detection at the beginning after VAD is enabled.

Set sample_cnt_en=0x1

Set int_trig_ctrl_en=0x1

Set int_trig_valid_thd to appropriate value, it recommended configuration to 4ms.

For 48KHz sample rate, int_trig_valid_thd = 0xc0

For 16KHz sample rate, int_trig_valid_thd = 0x40

For 8KHz sample rate, int_trig_valid_thd = 0x20

9. Set VAD_CONTROL register:

Set source_select=0x1, select I2S_8CH_0

Set source_burst=0x3, select INCR4 burst type

Set source_burst_num=0x0, select 1 burst transfer per DMA request

Set vad_mode=0x0, select Mode 0

Set voice_channel_num=0x7, all voice channel number is 8

Set voice_channel_bitwidth=0x0, voice data width is 16 bits

Set vad_det_channel=0x0, use channel 0 to voice activity detect

Set vad_en=0x1, enable VAD

10. After above setting, VAD will start to work and system can enter low power mode.

When VAD is working, user can configure the VAD_AUX_CON0.bus_write_en and VAD_AUX_CON0.ram_itf_dis to change the data storing mode dynamically. The storing address can be continuous transition or be spitted. It's controlled by following bits:

VAD_AUX_CON0.bus_write_addr_mode

VAD_AUX_CON0.bus_write_rework_addr_mode

VAD_AUX_CON0.ram_write_rework_addr_mode

14.5.3 Data Transfer Interrupt usage flow

When VAD is working at bus write mode. User can get data transfer interrupt by following additional configuration.

1. Set VAD_AUX_CONTROL.data_trans_trig_int_en=0x1
2. Set VAD_AUX_CONTROL.data_trans_kbyte_thd at appropriate value
3. Set VAD_INT.vad_data_trans_int_en=0x1

14.5.4 Timeout configuration usage flow

1. Set VAD_TIMEOUT.idle_timeout_thd=0xffff, set VAD_TIMEOUT.idle_timeout_en=0x1, set VAD_INT.idle_timeout_int=0x1. After above setting, a counter is increase at AHB clock when dmac_engine is idle, the counter will be clear to 0 once dmac_engine start to read voice data. An interrupt will be asserted when the counter up to idle_timeout_thd. This idle timeout is used for I2S/PDM work fail(Don't assert DMA request for a long time).

2. Set VAD_TIMEOUT.work_timeout_thd=0x3ff, set VAD_TIMEOUT.work_timeout_en=0x1, set VAD_INT.work_timeout_int=0x1. After above setting, a counter is increase at AHB clock when dmac_engine is busy, the counter will be clear to 0 once dmac_engine is idle. An interrupt will be asserted when the counter up to work_timeout_thd. This work timeout is used for bus transmission congestion (a burst transferring is not completed for a long time).

Chapter 15 WLAN Subsystem

15.1 Overview

The WLAN Subsystem is 802.11 b/g/n compliancy, and it consists of baseband PHY, MAC and AFE&RF. The key features are:

- PHY
 - SISO
 - Data rates:
 - ◆ 802.11n: MCS0-7(up to 72Mbps)
 - ◆ 802.11g: 6,9, 12, 18, 24, 36, 48, 54 Mbps
 - ◆ 802.11b: 1, 2, 5.5, 11 Mbps
 - 20 MHz bandwidth
 - Transmit power up to 16 dBm
 - Optimized bit width to reduce area/power
 - Slowest clock speed
 - Optimized listen mode power consumption
 - Increased immunity to noise/interference
 - Dynamic power management based on packet signal quality
 - Meets or exceeds standard specified sensitivity requirements including ACI/AACI
- MAC
 - HW/SW partition optimized to minimize power consumption
 - Processor to absorb traditional HOST functionality
 - Optimum PHY configuration based on previous beacon processing
 - Save/restore implementation and retention memories
 - Short Guard Interval(400ns)
 - AMPDU Aggregation, AMSDU Aggregation
- AFE&RF
 - Number of Antennas:1TX, 1RX
 - 2.4 GHz Operating Frequency Band
 - Power efficient AGC
 - PLL power optimization
 - Integrated Customized PA to minimize TX power consumption
 - Power optimized DAC/ADC design
- Support 96KB data buffer
- Ultra-Low Power Consumption
 - Low Power Rx Listen
 - IEEE Power Save Mode
 - U-APSD
 - Power Optimized Listen, Receive and Transmit Mode
 - Standby/Suspend Mode of operation with Power Gating
- Security: WPS, 802.1X, EAP, WEP, WEP2, WPA, WPA2, TKIP, 802.11i
- Network Infrastructure: Infrastructure Mode
- JTAG serial interface for debugging.

15.2 Block Diagram

The WLAN Subsystem includes the following key components:

- 802.11 b/g/n Baseband Hardware.
 - 802.11 b/g/n PHY
 - 802.11 MAC Hardware (Lower MAC, PHY Control, embedded microAptiv MIPS Processor)
- 802.11 b/g/n AFE & RF.
- Always-on support block.
- 96KB data buffer

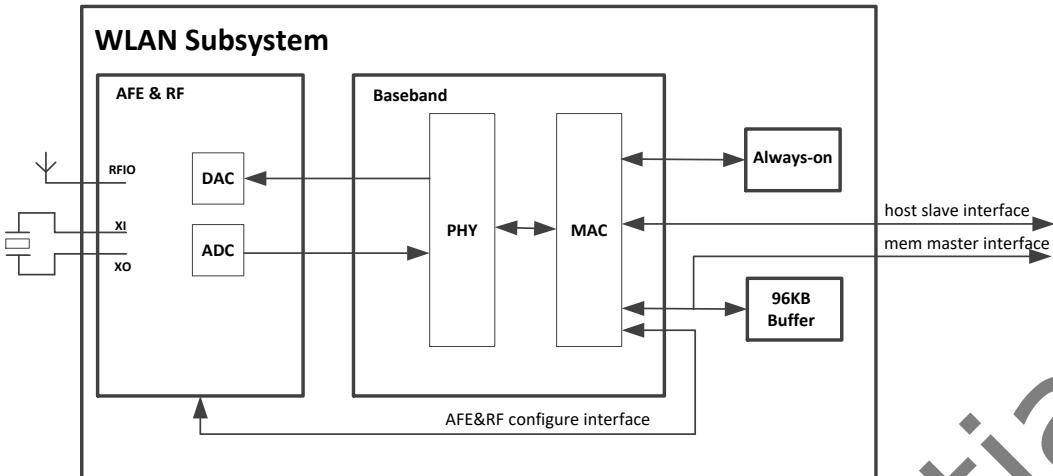


Fig. 15-1 WLAN Subsystem Block Diagram

15.3 Function Description

15.3.1 Clock

The functional clocks of the WLAN Subsystem are all generated by the 40 MHz crystal oscillator within the AFE&RF. The 40 MHz clock from the crystal is doubled within the RF to provide an 80 MHz clock that is used to drive the majority of the WLAN Subsystem components. The 40MHz and 80MHz clock output from WLAN Subsystem are the main clock source of the SoC.

15.3.2 Internal Memory Address Map

The WLAN Subsystem is embedded with 4 memory as shown below:

Table 15-1 WLAN Subsystem embedded memory

Region	Physical base Address(Byte) in MIPS side
ROM	0x00000000
RAM A	0x00040000
RAM B	0x00080000
GRAM	0x17000000

ROM, RAM A are the code memory of MIPS, and RAM B is the data memory of MIPS. They all can be accessed by Cortex-M0 for debugging, and RAM A is also used to apply patch to MIPS. Outside the WLAN Subsystem, ROM, RAM A and RAM B can only be accessed indirectly by follow registers: WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_CTRL, WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_WDATA and WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_RDATA.

Global RAM (GRAM) is used as packet buffer in WLAN Subsystem, it is also used as a shared memory mailbox for the HAL message. Outside the WLAN Subsystem, it can be accessed at base address 0x0xB7000000.

15.3.3 Interaction with Cortex-M0

The WLAN Subsystem is controlled by Cortex-M0. The interaction between the WLAN Subsystem and Cortex-M0 using HAL (Hardware Abstraction Layer) message control and Host Port Communications Interface. Messages flowing from Cortex-M0 to WLAN Subsystem are known as commands. Messages flowing from WLAN Subsystem to Cortex-M0 are known as events.

The HAL messages use a shared memory mailbox in WLAN Subsystem. And the Host Port Communications Interface use a group host port register. The host_to_mtx_cmd register is written to by the Cortex-M0 in order to send data to the WLAN Subsystem MIPS. The act of writing causes an event on the MIPS (the host_int interrupt occurs). The MIPS handles this event by reading host_to_mtx_cmd, collecting the message data. The MIPS clears the interrupt and in the process acknowledges reception of the message by writing to the mtx_to_host_ack register. The Cortex-M0 checks for this acknowledgement by reading host_to_mtx_cmd, checking the state of the HOST_INT bit. A message initiated by the

MIPS and destined for the Cortex-M0 uses the same scheme, but utilizing the mtx_to_host_cmd and host_to_mtx_ack registers and by responding to the mtx_int interrupt.

15.3.4 Power mode

The WLAN Subsystem transitions to different power states. The power state of WLAN Subsystem is controlled by always-on block. These states influence whether RAMs and functional blocks are powered.

- WLAN Subsystem Off: WLAN Subsystem MIPS firmware has not yet been booted. WLAN Subsystem state change from Off to On/Awake is cold boot. When WLAN Subsystem have been cold booted, a WIFI_READY interrupt will be asserted.
- WLAN Subsystem On/Awake: LMAC MIPS is running. GRAM is powered and accessible.
- Sleeping: WLAN Subsystem has been booted and is in a state which is minimizing power consumption. All the logic in WLAN Subsystem is power off except the retention RAM A. Sleeping mode have two situations as follow:
 - Not associated: The WLAN Subsystem will go to sleep after inactivity of HAL commands. The WLAN Subsystem will generally remain asleep in this state indefinitely until the Cortex-M0 wakes it up.
 - Associated: The WLAN Subsystem will enter power save mode and autonomously transition between awake and sleeping power states during the association to conserve power. In general, the WLAN Subsystem will remain awake when TX/RX data traffic is flowing. After traffic stops, the WLAN Subsystem will begin long sleeps and awaken for beacons. In this case, the WLAN Subsystem can be wake up by Cortex-M0 too.

WLAN Subsystem change from sleeping state to awake state is warm boot. A WIFI_WAKEUP interrupt will be asserted when leaving sleeping state. Cortex-M0 can polling for PMU.PWRDN_ST[6](lpw_sleep_state)=0 to check whether the power up sequence is end.

When WLAN Subsystem has been warm booted, a WIFI_READY interrupt will be asserted. WLAN Subsystem change from awake to sleeping state, a WIFI_SLEEP interrupt will be asserted. Cortex-M0 can polling for PMU.PWRDN_ST[6](lpw_sleep_state)=1 to check whether the power down sequence is end.

When WLAN subsystem enter power save mode and SoC enter low power mode, User can chose following 2 mode for SoC exiting low power mode:

- Use WIFI_WAKEUP interrupt as the awaken source, every time the WLAN awaken for beacons, the SoC exit low power mode.
- Use WLAN_WRITE_TRIGGER interrupt as the awaken source, the SoC will exit low power mode only when WLAN awaken for beacons and there is data to receive.

15.3.5 TX Sequence

- During the initialization, the Cortex-M0 allocates packet buffers in Internal SRAM for all of the TX queues. Logically, there is one TX queue per EDCA access category.
- SDIO transfer the packet from host chip to packet buffer.
- Cortex-M0 sends TX command to WLAN Subsystem via HAL and Host Port Communications Interface. The message header will contain a pointer to the packet buffer and related descriptor information.
- Immediately when it is time to transmit (i.e. the wireless channel is available), the WLAN Subsystem will initiate a DMA of a host packet buffer at the head of one of the EDCA queues. The queue is not determined until the time of transmission.
- When the transmission is done, the WLAN Subsystem will send a TX complete event to Cortex-M0. Cortex-M0 can free the packet buffer. A single TX complete event can be used to signal the transmission of multiple packets.

Cortex-M0 can send up to 11 commands to WLAN Subsystem continuous, then only after a TX complete event is received, Cortex-M0 can send next command to WLAN Subsystem.

15.3.6 RX Sequence

- During initialization, the Cortex-M0 allocates a ring of buffers for the packets reception from WLAN Subsystem
- The data pointers corresponding to these buffers passed to the WLAN Subsystem as RX descriptors.

- WLAN Subsystem uses these buffers to store the received packet. When a packet is received, the WLAN Subsystem sends an event to Cortex-M0.
- Cortex-M0 reads the event and identifies RX descriptors, then transfer each received packet to host chip through SDIO.
- After this, Cortex-M0 allocates new network buffers and replenishes the RX ring and informs the WLAN Subsystem regarding the new buffer pointer.

15.4 Register Description

15.4.1 WLAN Subsystem System Bus(SysBus) Registers Summary

Name	Offset	Size	Reset Value	Description
WLAN_SYSBUS_MIPS_MC_U_CONTROL	0x0000	W	0x00000000	MIPS MCU reset control
WLAN_SYSBUS_MIPS_MC_U_SYS_CORE_MEM_CTRL	0x0030	W	0x00000000	Address and control to MIPS MCU core memory
WLAN_SYSBUS_MIPS_MC_U_SYS_CORE_MEM_WDATA	0x0034	W	0x00000000	WDATA to MIPS MCU core memory
WLAN_SYSBUS_MIPS_MC_U_SYS_CORE_MEM_RDATA	0x0038	W	0x00000000	RDATA from MIPS MCU core memory
WLAN_SYSBUS_MIPS_MC_U_BOOT_EXCP_INSTR_0	0x0050	W	0x34010004	MIPS MCU boot exception instruction 0
WLAN_SYSBUS_MIPS_MC_U_BOOT_EXCP_INSTR_1	0x0054	W	0x40816000	MIPS MCU boot exception instruction 1
WLAN_SYSBUS_MIPS_MC_U_BOOT_EXCP_INSTR_2	0x0058	W	0x00000000	MIPS MCU boot exception instruction 2
WLAN_SYSBUS_MIPS_MC_U_BOOT_EXCP_INSTR_3	0x005c	W	0x42000020	MIPS MCU boot exception instruction 3
WLAN_SYSBUS_UCCP_CODE_HOST_TO mtx_CMD	0x0430	W	0x00000000	Host to MIPS MCU command
WLAN_SYSBUS_UCCP_CODE mtx_TO_HOST_CMD	0x0434	W	0x00000000	MIPS MCU to host command
WLAN_SYSBUS_UCCP_CODE_HOST_TO mtx_ACK	0x0438	W	0x00000000	Host to MIPS MCU command acknowledge
WLAN_SYSBUS_UCCP_CODE mtx_TO_HOST_ACK	0x043c	W	0x00000000	MIPS MCU to host command acknowledge
WLAN_SYSBUS_UCCP_CODE_HOST_INT_ENABLE	0x0440	W	0x00000000	Host to MIPS MCU interrupt enable
WLAN_SYSBUS_UCCP_CODE mtx_INT_ENABLE	0x0444	W	0x00000000	MIPS MCU to host interrupt enable
WLAN_SYSBUS_UCCP_CODE_FAB_STATUS	0x078c	W	0x00000000	ExtRAM and GRAM Request Status
WLAN_SYSBUS_UCC_SLE_EP_CTRL_DATA_0	0x2c2c	W	0x00000000	Sleep controller data

Name	Offset	Size	Reset Value	Description
WLAN_SYSBUS_UCC_SLE EP_CTRL MCU_BOOT_AD DR_MS	0x2dd4	W	0x00002000	MIPS MCU warm boot address
WLAN_SYSBUS_UCC_SLE EP_CTRL MCU_BOOT_AD DR_LS	0x2dd8	W	0x00000000	MIPS MCU warm boot address

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

15.4.2 WLAN System Bus(SysBus) Detail Register Description

WLAN_SYSBUS_MIPS MCU_CONTROL

Address: Operational Base + offset (0x0000)

MIPS MCU reset control

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	MIPS MCU_LATCH_SOFT_RESET A write of '1' to this register generates a soft reset. This soft reset is then latched. SW later on needs to write a '0' to negate the latched soft reset.
0	R/W SC	0x0	MIPS MCU_PULSE_SOFT_RESET The default value indicates that the MCU is held in reset. A write to this register generates a SW reset and this bit will be set to '1' to indicate that pulsed_soft_reset has been asserted. After 8 cycles (minimum assertion period required by MIPS MCU for the assertion of the pulsed_soft_reset), this bit will be cleared to '0', to indicate that the pulsed_soft_reset has been negated. NOTE: During hard reset, even if SW writes this bit, it will not be negated after 8 cycles, until the MCU comes out of reset.

WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_CTRL

Address: Operational Base + offset (0x0030)

Address and control to MIPS MCU core memory

Bit	Attr	Reset Value	Description
31	RW	0x0	MIPS MCU_CORE_MEM_RDNWR Controls direction of access to core memory. 1'b0: Allows data to be written to core memory using the MIPS MCU_CORE_MEM_WDATA register. 1'b1: Allows data to be read from core memory using the MIPS MCU_CORE_MEM_RDATA register.
30	RW	0x0	MIPS MCU_CORE_MEM_REGION Controls which of code/data core memory region is accessed. 1'b0: Code core memory region accessed. 1'b1: Data core memory region accessed.
29:1	RO	0x0	reserved
0	RW	0x0	MIPS MCU_CORE_MEM_ADDR Address (word) for core memory read/write access.

WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_WDATA

Address: Operational Base + offset (0x0034)

WDATA to MIPS MCU core memory

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	MIPS MCU CORE MEM WDATA Data to be written to MIPS MCU core memory. Reads always return zero. NOTE: The address is auto incremented after a data write operation.

WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_RDATA

Address: Operational Base + offset (0x0038)

RDATA from MIPS MCU core memory

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	MIPS MCU CORE MEM RDATA Data to be read from MIPS MCU core memory. Write any value to this register to auto-increment the address for the next read. NOTE: Only valid when MIPS MCU CORE MEM RDNWR=1.

WLAN_SYSBUS_MIPS MCU_BOOT_EXCP_INSTR_0

Address: Operational Base + offset (0x0050)

MIPS MCU boot exception instruction 0

Bit	Attr	Reset Value	Description
31:0	RW	0x34010004	MCU_BOOT_EXCP_INSTR_0 Indicates the first 32-bit instruction for the Boot Exception Vector. Note: The default value for this instruction is ORI \$1, \$0, 0x0004. Host can re-program this instruction as required. Note2: When the boot_in pin is asserted, then this register is re-programmed with the default value of (0x3C17 & boot_addr[31:16]) (LUI \$23, boot_addr[31:16]).

WLAN_SYSBUS_MIPS MCU_BOOT_EXCP_INSTR_1

Address: Operational Base + offset (0x0054)

MIPS MCU boot exception instruction 1

Bit	Attr	Reset Value	Description
31:0	RW	0x40816000	MIPS MCU_BOOT_EXCP_INSTR_1 Indicates the second 32-bit instruction for the Boot Exception Vector. Note: The default value for this instruction is mtc0, \$0, status. Host can re-program this instruction as required. Note2: When the boot_in pin is asserted, then this register is re-programmed with the default value of (0x26F7 & boot_addr[15:2] & b"00") (ORI \$23, (boot_addr[15:2] & b"00")).

WLAN_SYSBUS_MIPS MCU_BOOT_EXCP_INSTR_2

Address: Operational Base + offset (0x0058)

MIPS MCU boot exception instruction 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MCU_BOOT_EXCP_INSTR_2 Indicates the third 32-bit instruction for the Boot Exception Vector. Note: The default value for this instruction is EHB. Host can re-program this instruction as required. Note2: When the boot_in pin is asserted, then this register is re-programmed with the default value of 02E0_0008 (JR \$23).

WLAN_SYSBUS_MIPS MCU_BOOT_EXCP_INSTR_3

Address: Operational Base + offset (0x005c)

MIPS MCU boot exception instruction 3

Bit	Attr	Reset Value	Description
31:0	RW	0x42000020	MCU_BOOT_EXCP_INSTR_3 Indicates the fourth 32-bit instruction for the Boot Exception Vector. Note: The default value for this instruction is WAIT. Host can re-program this instruction as required. Note2: When the boot_in pin is asserted, then this register is re-programmed with the default value of 0x0 (NOP)

WLAN_SYSBUS_UCCP_CORE_HOST_TO mtx_CMD

Address: Operational Base + offset (0x0430)

Host to MIPS MCU command

Bit	Attr	Reset Value	Description
31	RO	0x0	HOST_INT This Read Only bit is set to 1 when the host_int event is asserted. Writing to HOST_INT_CLR in the mtx_to_host_ack register clears this bit.
30:0	RW	0x00000000	HOST_DATA Message data to be passed from the host to the MCU. A write to this register causes host_int to be asserted.

WLAN_SYSBUS_UCCP_CORE_MTX_TO_HOST_CMD

Address: Operational Base + offset (0x0434)

MIPS MCU to host command

Bit	Attr	Reset Value	Description
31	RO	0x0	MTX_INT This Read Only bit is set to 1 when mtx_int is asserted. Writing to MTX_INT_CLR in the host_to_mtx_ack register clears this bit.
30:0	RW	0x00000000	MTX_DATA Message data to be passed from the MCU to the host. A write to this register causes the mtx_int interrupt to be asserted.

WLAN_SYSBUS_UCCP_CORE_HOST_TO_MTX_ACK

Address: Operational Base + offset (0x0438)

Host to MIPS MCU command ack

Bit	Attr	Reset Value	Description
31	RW	0x0	MTX_INT_CLR Writing 1 to this bit causes the MTX_INT bit in mtx_to_host_cmd to be cleared.
30:0	RO	0x0	reserved

WLAN_SYSBUS_UCCP_CORE_MTX_TO_HOST_ACK

Address: Operational Base + offset (0x043c)

MIPS MCU to host command ack

Bit	Attr	Reset Value	Description
31	RW	0x0	HOST_INT_CLR Writing 1 to this bit causes the HOST_INT bit in host_to_mtx_cmd to be cleared.
30:0	RO	0x0	reserved

WLAN_SYSBUS_UCCP_CORE_HOST_INT_ENABLE

Address: Operational Base + offset (0x0440)

Host to MIPS MCU interrupt enable

Bit	Attr	Reset Value	Description
31	RW	0x0	HOST_INT_EN Controls the host interrupt line (host_int). 1'b0: The host interrupt is not enabled 1'b1: The host interrupt is enabled and the interrupt raised when HOST_INT=1
30:0	RO	0x0	reserved

WLAN_SYSBUS_UCCP_CORE_MTX_INT_ENABLE

Address: Operational Base + offset (0x0444)

MIPS MCU to host interrupt enable

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

Bit	Attr	Reset Value	Description
0	RW	0x0	MTX_INT_EN Controls the MCU interrupt line (mtx_int). 1'b0: The MCU interrupt is not enabled 1'b1: The MCU interrupt is enabled and the interrupt raised when MTX_INT=1

WLAN_SYSBUS_UCCP_SOC_FAB_STATUS

Address: Operational Base + offset (0x078c)

ExtRAM and GRAM Request Status

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	CR_EXTOPT_MEM_IDLE Indicates that there are no outstanding ExtRAM requests.
4	RW	0x0	CR_EXTOPT_REQ_IDLE Indicates that there are no outstanding ExtRAM requests.
3	RO	0x0	reserved
2	RW	0x0	CR_MEMOPT_IDLE Indicates that there are no outstanding GRAM requests.
1:0	RO	0x0	reserved

WLAN_SYSBUS_UCC_SLEEP_CTRL_DATA_0

Address: Operational Base + offset (0x2c2c)

Sleep controller data

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	DATA_0 Arbitrary 20-bit data storage register.

WLAN_SYSBUS_UCC_SLEEP_CTRL_MCU_BOOT_ADDR_MS

Address: Operational Base + offset (0x2dd4)

MIPS MCU warm boot address

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RW	0x2000	MCU_BOOT_ADDR_MS Bits 29:16 of the 30-bit word aligned MCU boot address.

WLAN_SYSBUS_UCC_SLEEP_CTRL_MCU_BOOT_ADDR_LS

Address: Operational Base + offset (0x2dd8)

MIPS MCU warm boot address

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	MCU_BOOT_ADDR_LS Bits 15:0 of the 30-bit word aligned MCU boot address.

15.4.3 WLAN Peripheral Bus(Pbus) Registers Summary

Name	Offset	Size	Reset Value	Description
WLAN_PBUS_RPU_CLOCK_RESET_CTRL_CLOCK_ENABLE_1	0x08c20	W	0x00000000	WLAN clock enable control
WLAN_PBUS_EDC_GPIO0_OUT	0x13c00	W	0x00000000	GPIO0 out
WLAN_PBUS_EDC_GPIO1_OUT	0x13c04	W	0x00000000	GPIO1 out
WLAN_PBUS_EDC_GPIO0_IN	0x13c08	W	0x00000000	GPIO0 in
WLAN_PBUS_EDC_GPIO1_IN	0x13c0c	W	0x00000000	GPIO1 in

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

15.4.4 WLAN Peripheral Bus(Pbus) Detail Register Description

WLAN_PBUS_RPU_CLOCK_RESET_CTRL_CLOCK_ENABLE_1

Address: Operational Base + offset (0x08c20)

WLAN clock enable control

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	CLOCK_ENABLE_MCP Enable the MCP #1 clock. 1'b0: No action. 1'b1: Enable clock.
15:5	RO	0x0	reserved
4	RW	0x0	CLOCK_ENABLE MCU Enable the MCU #1 clock. 1'b0: No action. 1'b1: Enable clock.
3:1	RO	0x0	reserved
0	RW	0x0	CLOCK_ENABLE_ALL Enable all clocks. 1'b0: No action. 1'b1: Enable all clocks.

WLAN_PBUS_EDC_GPIO0_OUT

Address: Operational Base + offset (0x13c00)

GPIO0 out

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GPIO0_OUT The bits 15:0 of the WLAN GPIO_OUT.

WLAN_PBUS_EDC_GPIO1_OUT

Address: Operational Base + offset (0x13c04)

GPIO1 out

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GPIO1_OUT The bits 31:16 of the WLAN GPIO_OUT.

WLAN_PBUS_EDC_GPIO0_IN

Address: Operational Base + offset (0x13c08)

GPIO0 in

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GPIO0_IN The bits 15:0 of the WLAN GPIO_IN.

WLAN_PBUS_EDC_GPIO1_IN

Address: Operational Base + offset (0x13c0c)

GPIO1 in

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	GPIO1_IN The bits 31:16 of the WLAN GPIO_IN.

15.5 Interface Description

Table 15-2 WLAN Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
RF_PAD_XI	I	RF_PAD_XI	NS
RF_PAD_XO	O	RF_PAD_XO	NS
RF_PAD_RFIO	I/O	RF_PAD_RFIO	NS
AONJTAG_TCK	I	LCD_D0/CIF_D0/I2C0_SDA_M2/TKEY1_2/M4F_WFI/M4F_JTAG_TCK/M0_JTAG_TCK/AONJTAG_TCK/DSPJTAG_TCK/GPIO0_A0_u	GRF_GPIO0A_IOMUX_L [3:0]=4'b0111
AONJTAG_TRST_N	I	LCD_RDN/CIF_CLKOUT/UART1_CTSN_M1/TKEY16/PMU_SLEEP/PMU_STATE2/CODEC_CLK_M1/AONJTAG_TRSTn/DS_PJTAG_TRSTn/GPIO0_A4_u	GRF_GPIO0A_IOMUX_H [3:0]=4'b1000
AONJTAG_TMS	I	LCD_D1/CIF_D1/I2C0_SCL_M2/TKEY1_3/M0_WFI/M4F_JTAG_TMS/M0_JTAG_TMS/AONJTAG_TMS/DSPJTAG_TMS/GPIO0_A1_u	GRF_GPIO0A_IOMUX_L [7:4]=4'b0111
AONJTAG_TDI	I	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_DRIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPJTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_L [11:8]=4'b0111
AONJTAG_TDO	O	LCD_CSn/CIF_VSYNC/I2C1_SCL_M2/TKEY15/PMU_DEBUG/PMU_STATE1/AONJTAG_TDO/DSPJTAG_TDO/GPIO0_A3_u	GRF_GPIO0A_IOMUX_L [15:12]=4'b0111
bt_mbsy	I	LCD_CSn/CIF_VSYNC/I2C1_SCL_M2/TKEY15/PMU_DEBUG/PMU_STATE1/AONJTAG_TDO/DSPJTAG_TDO/GPIO0_A3_u	GRF_GPIO0A_IOMUX_L [15:12]=4'b1001

Module Pin	Direction	Pad Name	IOMUX Setting
bt_rxntx	I	LCD_RDN/CIF_CLKOUT/UART1_CTSN_M1/TKEY16/PMU_SLEEP/PMU_STATE2/CODEC_CLK_M1/AONJTAG_TRSTn/DSPJTAG_TRSTn/GPIO0_A4_u	GRF_GPIO0A_IOMUX_H[3:0]=4'b1010
bt_deny	O	LCD_WRN/CIF_CLKIN/UART1_RTSN_M1/PDM_CLK_M0/TKEY17/PMU_STATE3/CODEC_SYNC_M1/GPIO0_A5_d	GRF_GPIO0A_IOMUX_H[7:4]=4'b1000
bt_confirm	O	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_DRIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPJTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_L[11:8]=4'b1001
bt_pti0	I	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_DRIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPJTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_L[11:8]=4'b1010

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

15.6 Application Notes

15.6.1 Cold boot

- Set GRF.GRF_WLANCLK_CON[5](lpw_aon_wakeup)=1 to wake up WLAN.
- Polling for PMU.PWRDN_ST[6](lpw_sleep_state)=0
- Configure the RPU_CLOCK_RESET_CTRL_CLOCK_ENABLE_1.CLOCK_ENABLE_ALL=1 to enable the clock in WLAN
- Enable the WIFI_READY interrupt
- If a patch need to apply, download the patch into WLAN RAM A, and configure the UCC_SLEEP_CTRL_DATA_0 to 0x4b00
- Boot the MIPS

Configure the follow register:

MIPS MCU_BOOT_EXCP_INSTR_0 = 0x3c1a8000
MIPS MCU_BOOT_EXCP_INSTR_1 = 0x275a0000
MIPS MCU_BOOT_EXCP_INSTR_2 = 0x03400008
MIPS MCU_BOOT_EXCP_INSTR_3 = 0x00000000

Processor execution should then be started by asserting

MIPS MCU_PULSE_SOFT_RESET=1 to start execution. This bit automatically clears when the MIPS core is brought out of reset. After reset, the MIPS will boot from KSeg0 Virtual

address 0x80000000(comprised by low 4 byte of the MIPS MCU_BOOT_EXCP_INSTR_0 and MIPS MCU_BOOT_EXCP_INSTR_1), it's the start address of WLAN ROM. The MIPS will decide whether apply a patch by reading the register UCC_SLEEP_CTRL_DATA_0.

- Set GRF.GRF_WLANCLK_CON[5](lpw_aon_wakeup)=0.
- A WIFI_READY interrupt will be generated when the MIPS have booted. The Cortex-M0 can clear the WIFI_READY interrupt, and then choose to send host port jobs to initiate transmit, receive, or power operations using the HAL message and Host Port Communications Interface.

15.6.2 Cortex-M0 request Warm boot

- Set GRF.GRF_WLANCLK_CON[5](lpw_aon_wakeup)=1 to wake up WLAN.
- Polling for PMU.PWRDN_ST[6](lpw_sleep_state)=0.
- Set GRF.GRF_WLANCLK_CON[5](lpw_aon_wakeup)=0.
- A WIFI_READY interrupt will be generated when the MIPS have booted. The Cortex-M0 can clear the WIFI_READY interrupt, and then choose to send host port jobs to initiate transmit, receive, or power operations using the HAL message and Host Port Communications Interface.

15.6.3 Access code/data memory of WLAN

- Write data to code memory of MIPS
 - Set register WLAN_SYSBUS_MIPS_MCU_SYS_CORE_MEM_CTRL set MIPS_MCU_CORE_MEM_RDNWR=0

- set MIPS MCU CORE MEM REGION=0
MIPS MCU CORE MEM ADDR=0x10000
- Set register WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_WDATA continuous as required. The address is auto incremented after a data write operation
- Write data to data memory of MIPS
 - Set register WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_CTRL
set MIPS MCU CORE MEM RDNWR=0
set MIPS MCU CORE MEM REGION=1
MIPS MCU CORE MEM ADDR=0x20000
 - Set register WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_WDATA continuous for the wdata. The address is auto incremented after a data write operation
- Read data from code memory of MIPS
 - Set register WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_CTRL
set MIPS MCU CORE MEM RDNWR=1
set MIPS MCU CORE MEM REGION=0
MIPS MCU CORE MEM ADDR=0x10000
 - Read data from MIPS MCU CORE MEM_RDATA. Write any value to this register to auto-increment the address for the next read.
- Read data from data memory of MIPS
 - Set register WLAN_SYSBUS_MIPS MCU_SYS_CORE_MEM_CTRL
set MIPS MCU CORE MEM RDNWR=1
set MIPS MCU CORE MEM REGION=1
MIPS MCU CORE MEM ADDR=0x20000
 - Read data from MIPS MCU CORE MEM_RDATA. Write any value to this register to auto-increment the address for the next read.

15.6.4 Send command to WLAN

- Polling for UCCP_CORE_HOST_TO_MTX_CMD. HOST_INT=0
- Polling for UCCP_SOC_FAB_STATUS. CR_MEMOPT_IDLE=1
- Get the message buffer address(cmd_address) in GRAM by reading 0xb7000ffc
- Send cmd to GRAM at cmd_address
- Set UCCP_CORE_HOST_TO_MTX_CMD. HOST_DATA=0x7ffdead

15.6.5 Receive event from WLAN

- Wait the WIFI interrupt
- Polling for UCCP_SOC_FAB_STATUS. CR_MEMOPT_IDLE=1
- Get message buffer address(event_address) in GRAM by reading 0xb7001038
- Get event from event_address in GRAM
- Get message buffer clear address(event_clr_address) in GRAM by reading 0xb700103c
- Set WLAN_SYSBUS_UCCP_CORE_HOST_TO_MTX_ACK. MTX_INT_CLR=1
- Write 0 to event_clr_address in GRAM to clear the message buffer.

Chapter 16 Pulse Density Modulation Interface Controller

16.1 Overview

The PDM interface controller and decoder 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.

PDM 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 1 path. The path is composed of two digital microphone channels, the PDM can be used with one stereo or two 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 left and right channel exchange

16.2 Block Diagram

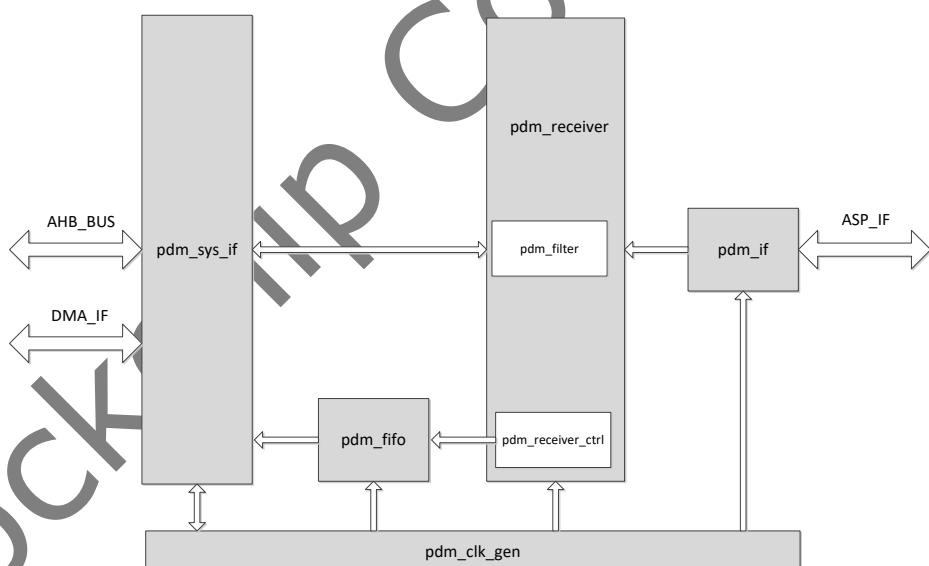


Fig. 16-1 PDM Block Diagram

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.

16.3 Function Description

16.3.1 AHB Interface

There is an AHB slave interface in PDM. It is responsible for accessing registers.

16.3.2 PDM Interface

The PDM interface is a 2-wire interface. The PDM module can support up to one external stereo and two digital microphones.

Following shows two cases of usage of the PDM, but all configurations are possible with stereo and mono digital microphones.

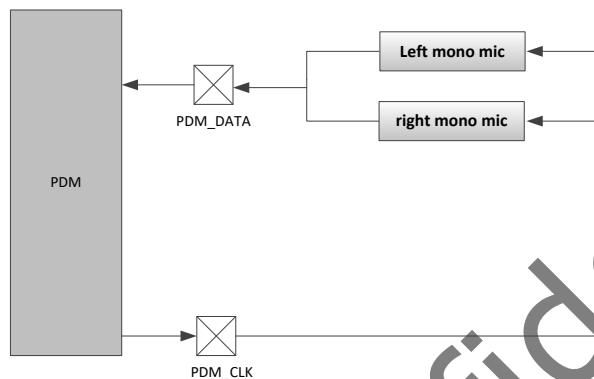


Fig. 16-2 PDM with two mono mic

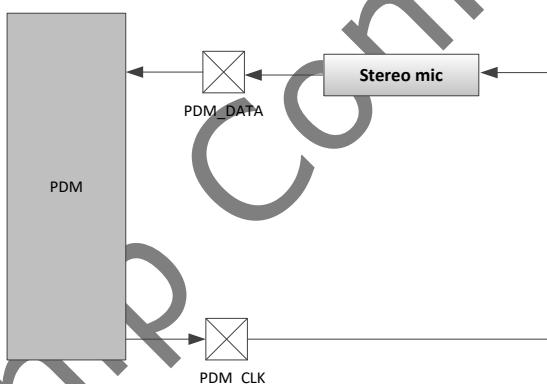


Fig. 16-3 PDM with Four Stereo mic

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

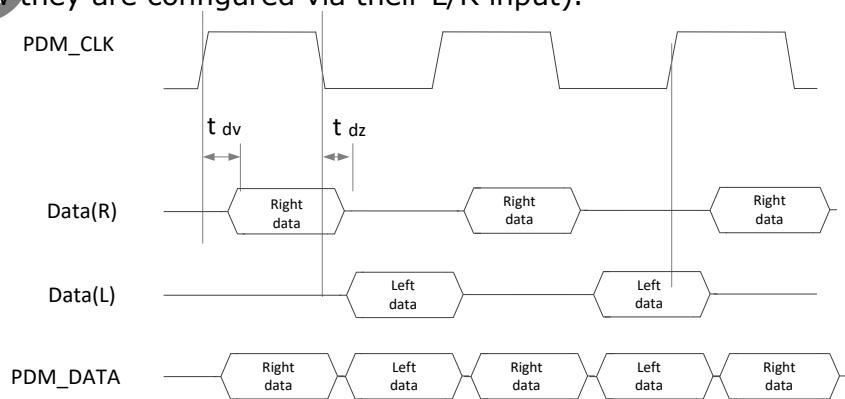


Fig. 16-4 PDM interface diagram with external mic

16.3.3 Digital Filter

The external PDM mic generates a PDM stream of bits and transfers it in one period or one half-period of the clock provided by the PDM. The aim of the PDM 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.

16.3.4 Frequency Configuration

MCLK is the source clock signal. PDM_CLK (refers to O_asp_clk in the Fig. 16-5) is the output clocks generated in the PDM and is fed to the external microphones. They are also the internal clock of the external microphones. User must take care about the frequency of PDM_CLK when selecting the source clock (MCLK).

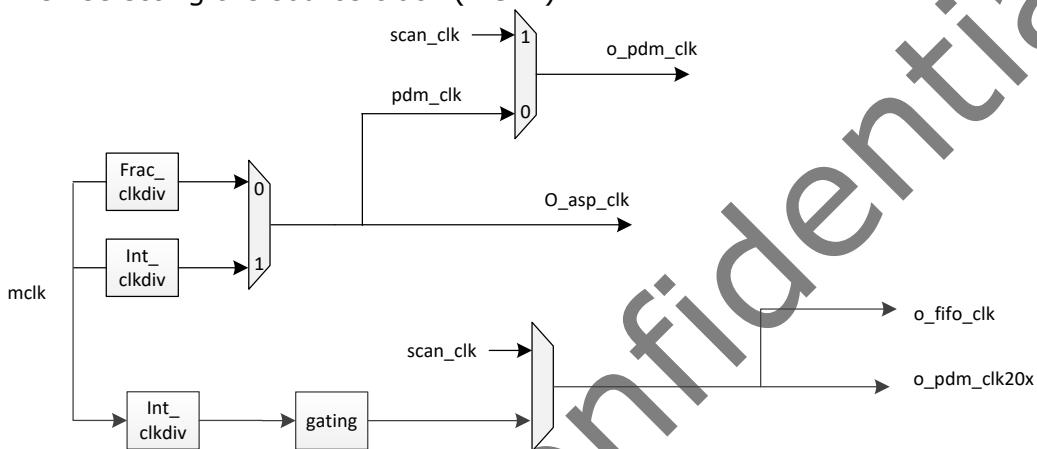


Fig. 16-5 PDM Clock Structure

Table 16-1 Relation between PDM_CLK and sample rate

PDM_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

16.4 Register Description

16.4.1 Registers Summary Internal Address Mapping

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

16.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
PDM_SYSCONFIG	0x0000	W	0x00000000	PDM System Configure Register
PDM_CTRL0	0x0004	W	0x78000060	PDM Control Register 0
PDM_CTRL1	0x0008	W	0x0bb8ea60	PDM Control Register 1
PDM_CLK_CTRL	0x000c	W	0x0000e400	PDM Clock Control Register
PDM_HPF_CTRL	0x0010	W	0x00000000	PDM High-pass Filter Control Register
PDM_FIFO_CTRL	0x0014	W	0x00000000	PDM FIFO Control Register
PDM_DMA_CTRL	0x0018	W	0x0000001f	PDM DMA Control Register
PDM_INT_EN	0x001c	W	0x00000000	PDM Interrupt Enable Register
PDM_INT_CLR	0x0020	W	0x00000000	PDM Interrupt Clear Register
PDM_INT_ST	0x0024	W	0x00000000	PDM Interrupt Status Register

Name	Offset	Size	Reset Value	Description
PDM_RXFIFO_DATA_REG	0x0030	W	0x00000000	PDM Receive FIFO Data Register
PDM_DATA0R_REG	0x0034	W	0x00000000	PDM Path0 Right Channel Data Register
PDM_DATA0L_REG	0x0038	W	0x00000000	PDM Path0 Left Channel Data Register
PDM_DATA_VALID	0x0054	W	0x00000000	PDM Path Data Valid Register
PDM_VERSION	0x0058	W	0x59313031	PDM Version Register
PDM_INCR_RXDR	0x0400	W	0x00000000	Increment Address Receive FIFO Data Register

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

16.4.3 Detail Register Description

PDM_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 PDM RX logic clear This is a self-cleared bit. High active. Write 0x1: Clear RX logic Write 0x0: No action Read 0x1: Clear ongoing Read 0x0: Clear done

PDM_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 contains two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 1'b0: Right justified 1'b1: Left justified
30:28	RW	0x1	reserved

Bit	Attr	Reset Value	Description
27	RW	0x1	path0_en Path 0 enable 1'b1: Enable 1'b0: Disable
26	RW	0x0	hwt_en 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	RW	0x0	filter_gate_en Filter gate enable If some filters not work, the filter and its corresponding memory clock will be gated if filter_gate_en is 1'b1, otherwise the clock will be still active.
24	RW	0x0	sig_scale_mode Signal scale mode select 1'b0: Cic outputs the normal latitude. 1'b1: Scale the cic outputs to half of the normal latitude and scale 2 times after hpf-filter.
23:16	RW	0x00	int_div_20x_con Integer divider for PDM filter operation.
15:8	RW	0x00	int_div_con Integer divider Can be written only when SYSCONFIG[2] is 0.
7:5	RW	0x3	sample_rate_sel Selects which kind of sample rate. 3'b000: 12kHz/11.024kHz/8kHz 3'b001: 24kHz/22.05kHz/16kHz 3'b010: 32kHz 3'b011: 48kHz/44.1kHz 3'b100: 96kHz/88.2kHz/64kHz 3'b101~3'b111: 192kHz/176.4kHz/128kHz

Bit	Attr	Reset Value	Description
4:0	RW	0x00	<p>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</p>

PDM CTRL1

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0bb8	<p>frac_div_numerator Fraction divider numerator Can be written only when SYSCONFIG[2] is 0.</p>
15:0	RW	0xea60	<p>frac_div_denominator Fraction divider denominator Can be written only when SYSCONFIG[2] is 0.</p>

PDM CLK CTRL

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	RW	0x3	reserved
13:12	RW	0x2	reserved
15:10	RW	0x1	reserved
9:8	RW	0x0	<p>rx_path_select0 RX Path Select 2'b00: Path0 data from PDM data0 2'b01~2'b11: Reserved</p>
7:6	RO	0x0	reserved
5	RW	0x0	<p>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</p>
4	RW	0x0	<p>div_type_sel Divider type select signal Can be written only when SYSCONFIG[2] is 0. 1'b0: Fraction divider 1'b1: Integer divider</p>

Bit	Attr	Reset Value	Description
3	RW	0x0	lr_ch_ex Left and right channel data exchange 1'b0: Not inverted 1'b1: Inverted
2	RW	0x0	fir_com_bps Fir compensate filter bypass 1'b0: Not bypass 1'b1: Bypass
1:0	RW	0x0	cic_ds_ratio CIC filter decimation ratio 2'b00: 16 times decimation 2'b01: 8 times decimation 2'b10: 4 times decimation other: 8 times decimation

PDM HPF CTRL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	hpfe 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 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 High-pass filter configure 2'b00: 3.79Hz 2'b01: 60Hz 2'b10: 243Hz 2'b11: 493Hz

PDM FIFO CTRL

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	rft Receive FIFO threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO threshold interrupt is triggered.

Bit	Attr	Reset Value	Description
7:0	RO	0x00	rfl Receive FIFO level Contains the number of valid data entries in the receive FIFO.

PDM 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
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.

PDM 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	rxftie RX full threshold interrupt enable 1'b0: Disable 1'b1: Enable

PDM INT CLR

Address: Operational Base + offset (0x0020)

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

PDM 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	rfxi RX full interrupt 1'b0: Inactive 1'b1: Active

PDM_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.

PDM_DATA0R_REG

Address: Operational Base + offset (0x0034)

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

PDM_DATA0L_REG

Address: Operational Base + offset (0x0038)

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

PDM_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	RO	0x0	reserved
1	RO	0x0	reserved
0	RO	0x0	reserved

PDM VERSION

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x59313031	version PDM version

PDM INCR 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.

16.5 Interface Description

There are two groups of PDM IO interfaces embedded in the chip. Following figure shows the group 0 and group 1 PDM IO interface respectively.

Table 16-2 Group 0 PDM IO Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
o_pdm_ck	O	LCD_WRN/CIF_CLKIN/UART1_RTSN_M1/PDM_CLK_M0/TKEY17/PMU_STATE3/CODEC_SYNC_M1/GPIO0_A5_d or LCD_D3/CIF_D3/UART1_TX_M1/PDM_CLK_S_M0/TKEY19/TEST_CLKOUT/CODEC_DA_C_DL_M1/GPIO0_A7_d	GRF_GPIO0A_IOMUX_H[7:4]=4'b0100 or GRF_GPIO0A_IOMUX_H[15:12]=4'b0100
i_pdm_d ata0	I	LCD_D2/CIF_D2/UART1_RX_M1/PDM_SDI_M0/TKEY18/PMU_STATE4/CODEC_ADC_D_M1/GPIO0_A6_d	GRF_GPIO0A_IOMUX_H[11:8]=4'b0100

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

Table 16-3 Group 1 PDM IO Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
o_pdm_ck	O	SRADC4/PWM4_M0/UART0_CTSN_M1/SPI1_CS0n_M0/PDM_CLK_M1/I2S_SCLK_TX_M0/TKEY5_M0/GPIO0_C4_u or SRADC3/PWM3_M0/UART1_TX_M0/SPI0_MISO_M1/PDM_CLK_S_M1/I2S_MCLK_M0/TKEY4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GRF_GPIO0C_IOMUX_H[3:0]=4'b0101 or GRF_GPIO0C_IOMUX_L[15:12]=4'b0101
i_pdm_d ata0	I	SRADC5/PWM5_M0/UART0_RTSN_M1/SPI1_CLK_M0/PDM_SDI_M1/I2S_LRCK_TX_M0/TKEY6_M0/GPIO0_C5_u	GRF_GPIO0C_IOMUX_H[7:4]=4'b0101

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

16.6 Application Notes

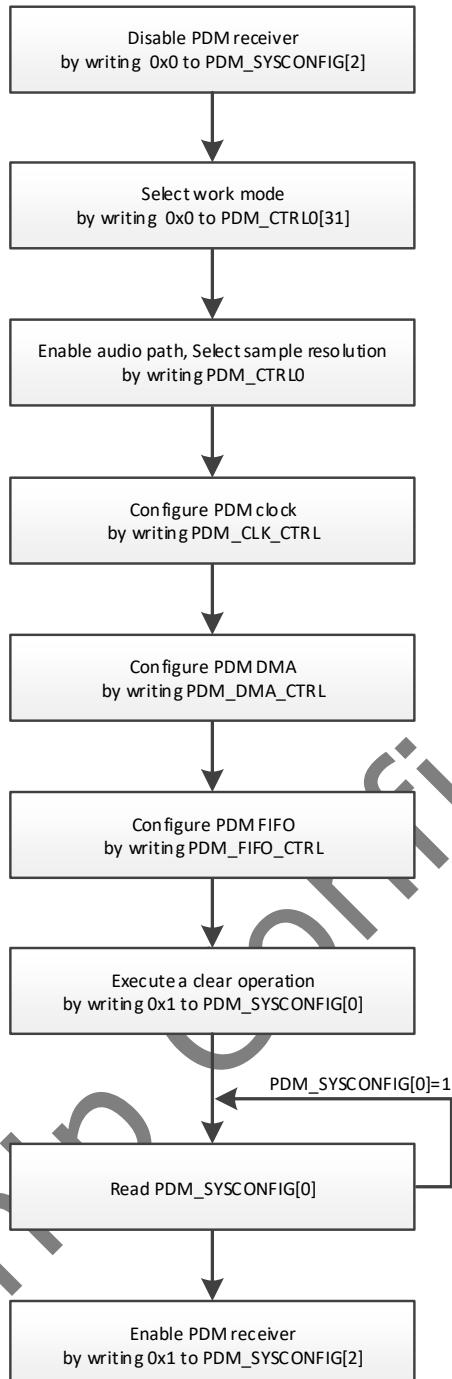


Fig. 16-6 PDM Operation Flow

Chapter 17 FSPI (Flexible Serial Peripheral Interface)

17.1 Overview

The FSPI is a flexible serial peripheral interface host controller to interface with external device.

The FSPI supports the following features:

- Support various vendor devices with flexible command sequencer engine
 - Serial NOR Flash
 - Serial NAND Flash
 - Serial pSRAM
 - Serial SRAM
- Support SDR mode
- Support Single/Dual/Quad IO mode
- Support a 32-bit AHB slave to read and write controller register bank and initiate command sequence, including transfer data from/to external device indirectly
- Support a 32-bit AHB master with embedded DMA engine to transfer data from/to external device indirectly
- Support a 128-bit Memory-Mapped AHB slave to read from and write data to external device directly
- Support Memory-Mapped read with optional prefetch buffer
- Support independent clock for system bus HCLK and controller SCLK
- Support maskable interrupt generation
- Support sampling clock with optionally configurable delay line

17.2 Block Diagram

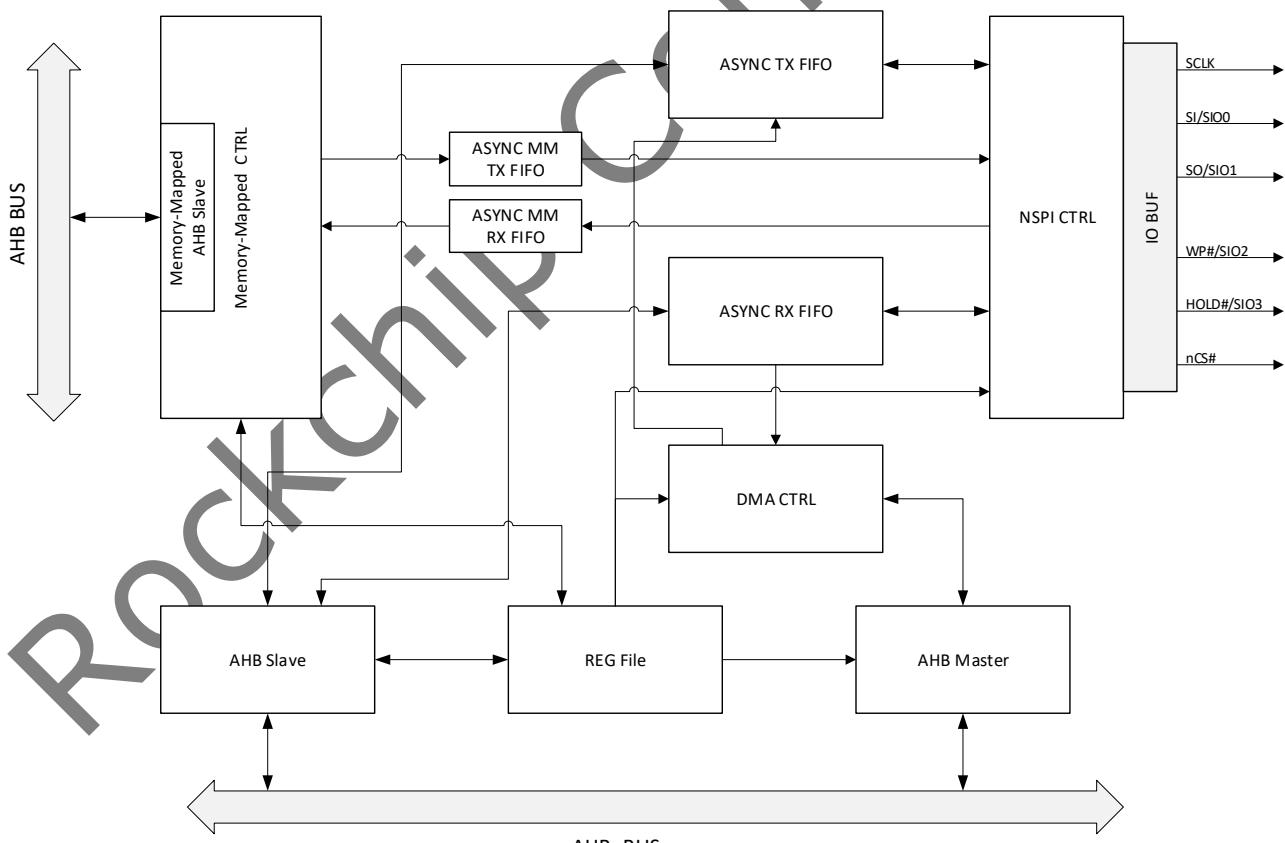


Fig. 17-1 FSPI Architecture

17.3 Function Description

17.3.1 AHB Slave

The AHB slave block is used to configure the register of controller to generate flexible

command sequence, process the interrupt exception, target various device feature and AC timing specification. It is also used to write CMD/ADDR/DATA to TX FIFO and read DATA from RX FIFO which buffer the DATA from external device.

17.3.2 AHB Master

When the embedded DMA CTRL is used to transfer DATA, the AHB master is used to transfer data to other system region, such as internal SRAM, peripheral, external DRAM.

17.3.3 Memory-Mapped AHB Slave

After the software initialize the controller based on the specialized memory device, CPU and other system bus masters can read data from external memory directly. If the external memory is SRAM or pSRAM, it also supports write data to it. The Memory-Mapped AHB Slave module can generate the relative CS# based on the access address from system address, example CS#0 and CS#1.

17.3.4 REG File

The REG File is configurable register bank to control the store the static configuration and dynamic status of controller.

17.3.5 DMA CTRL

A block takes responsible for splitting a long length transfer trans into AHB bus transaction and interfacing with ASYNC TX or RX FIFO.

17.3.6 FIFO

There are four FIFO in the FSPI controller. ASYNC TX FIFO and ASYNC RX FIFO is for normal transaction that initiated by command sequence driver. The ASYNC MM (Memory-Mapped) TX FIFO and ASYNC MM (Memory-Mapped) RX FIFO is only used to buffer DATA from or to external device initiated by system bus master directly.

17.3.7 NSPI CTRL

Sequence decode engine which generates specialized timing sequence for various device. The NSPI decode the transaction from TX FIFO and Memory-Mapped Controller and convert it to relative CMD/ADDR/DATA frame based on the configuration.

17.4 Register Description

17.4.1 Internal Address Mapping

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

Table 17-1 FSPI Address Mapping Table

Name	Address Base	Size
FSPI0 CFG	0x40210000	64KB
FSPI0 MEM	0x10000000/0x30000000	64MB
FSPI1 CFG	0x40220000	64KB
FSPI1 MEM	0x18000000/0x38000000	32MB

For FSPI1 MEM regions access, it's exclusive with HyperBus MEM regions access. The Application should configure the GRF_SOC_CON0.grf_con_hyperx8_sfc1_sel to select these regions mapped as FSPI1 MEM address.

17.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
FSPI_CTRL0	0x0000	W	0x00000000	Control Register for CS0 Device
FSPI_IMR	0x0004	W	0x000001ff	Interrupt Mask Register
FSPI_ICLR	0x0008	W	0x00000000	Interrupt Clear Register
FSPI_FTMR	0x000c	W	0x00001010	FIFO Threshold Level Register
FSPI_RCVR	0x0010	W	0x00000000	FSPI Recover Register

<u>FSPI_AX0</u>	0x0014	W	0x000000a5	FSPI Auxiliary Data Value for CS0 Device
<u>FSPI_ABITO</u>	0x0018	W	0x00000000	Extend Address Bits for CS0 Device
<u>FSPI_ISR</u>	0x001c	W	0x00000000	Interrupt Status
<u>FSPI_FSR</u>	0x0020	W	0x00000001	FIFO Status Register
<u>FSPI_SR</u>	0x0024	W	0x00000000	FSPI Status Register
<u>FSPI_RISR</u>	0x0028	W	0x00000000	Raw Interrupt Status Register
<u>FSPI_VER</u>	0x002c	W	0x00000003	Version Register
<u>FSPI_QOP</u>	0x0030	W	0x00000000	Quad Line Operation IO Level Preset Register
<u>FSPI_EXT_CTRL</u>	0x0034	W	0x00004023	Extend Control Register
<u>FSPI_POLL_CTRL</u>	0x0038	W	0x00000000	Device Status Polling Control Register
<u>FSPI_DLL_CTRL0</u>	0x003c	W	0x00000001	Delay Line Control Register for CS0 Device
<u>FSPI_HRDYMASK</u>	0x0040	W	0x00000000	AMBA HREADY Mask Control Register
<u>FSPI_EXT_AX</u>	0x0044	W	0x0000f0ff	Extend Auxiliary Data Control Register
<u>FSPI_SCLK_INATM_CNT</u>	0x0048	W	0xffffffff	SCLK Inactive Timeout Counter
<u>FSPI_AUTO_RF_CNT</u>	0x004c	W	0xffffffff	Auto Refresh Counter
<u>FSPI_XMMC_WCMD0</u>	0x0050	W	0x00000000	Memory Mapped Control Write Command Register for CS0 Device
<u>FSPI_XMMC_RCMD0</u>	0x0054	W	0x00000000	Memory Mapped Control Read Command Register for CS0 Device
<u>FSPI_XMMC_CTRL</u>	0x0058	W	0x000072e0	Memory Mapped Control Register
<u>FSPI_MODE</u>	0x005c	W	0x00000000	Controller Working Mode Register
<u>FSPI_DEVRGN</u>	0x0060	W	0x00000017	Device Region Size Register
<u>FSPI_DEVSIZE0</u>	0x0064	W	0x00000012	Device Size Register for CS0 Device
<u>FSPI_TME0</u>	0x0068	W	0x00000000	Timeout Enable Control Register for CS0 Device
<u>FSPI_POLLDLY_CTRL</u>	0x006c	W	0x0003ffff	Polling Delay Control Register
<u>FSPI_DMATR</u>	0x0080	W	0x00000000	DMA Trigger Register
<u>FSPI_DMAADDR</u>	0x0084	W	0x00000000	DMA Address Register
<u>FSPI_POLL_DATA</u>	0x0090	W	0x00000000	Polling Data Register
<u>FSPI_XMMCSR</u>	0x0094	W	0x00000000	Memory Mapped Status Register
<u>FSPI_CMD</u>	0x0100	W	0x00000000	Indirect Command Register
<u>FSPI_ADDR</u>	0x0104	W	0x00000000	Address Register
<u>FSPI_DATA</u>	0x0108	W	0x00000000	Data Register
<u>FSPI_CTRL1</u>	0x0200	W	0x00000000	Control Register for CS1 Device
<u>FSPI_AX1</u>	0x0214	W	0x000000a5	FSPI Auxiliary Data Value for CS1 Device

<u>FSPI ABIT1</u>	0x0218	W	0x00000000	Extend Address Bits for CS1 Device
<u>FSPI DLL CTRL1</u>	0x023c	W	0x00000001	Delay Line Control Register for CS1 Device
<u>FSPI XMMC WCMD1</u>	0x0250	W	0x00000000	Memory Mapped Control Write Command Register for CS1 Device
<u>FSPI XMMC RCMD1</u>	0x0254	W	0x00000000	Memory-Mapped Command Control Register for CS1 Device
<u>FSPI DEVSIZE1</u>	0x0264	W	0x00000012	Device Size Register for CS1 Device
<u>FSPI TME1</u>	0x0268	W	0x00000000	Timeout Enable Control Register for CS1 Device

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

17.4.3 Detail Register Description

FSPI CTRL0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
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 Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
11:10	RW	0x0	ADRB Address line width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command line width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.

Bit	Attr	Reset Value	Description
7:4	RW	0x0	<p>IDLE_CYCLE Idle cycles when switching IO from output to input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.</p>
3:2	RO	0x0	reserved
1	RW	0x0	<p>SHIFTPHASE Shift phase of data input in controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".</p>
0	RW	0x0	<p>SPIIM Serial peripheral interface mode 1'b0: Mode 0 1'b1: Mode 3 SPIIM is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.</p>

FSPI IMR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x1	<p>STPOLLM Status polling done interrupt mask 1'b0: Status polling done interrupt is not masked 1'b1: Status polling done interrupt is masked Only valid in indirect access mode.</p>
7	RW	0x1	<p>DMAM DMA finish interrupt mask 1'b0: DMA finish interrupt is not masked 1'b1: DMA finish interrupt is masked Only valid in indirect access mode.</p>
6	RW	0x1	<p>NSPIM NSPI interrupt mask 1'b0: NSPI interrupt is not masked 1'b1: NSPI interrupt is masked Valid in indirect access mode and memory mapped mode.</p>

Bit	Attr	Reset Value	Description
5	RW	0x1	AHBM AMBA AHB error interrupt mask 1'b0: AMBA AHB Error interrupt is not masked 1'b1: AMBA AHB Error interrupt is masked Only valid in indirect access mode.
4	RW	0x1	TRANSM Transfer finish interrupt mask 1'b0: Transfer finish interrupt is not masked 1'b1: Transfer finish interrupt is masked Only valid in indirect access mode.
3	RW	0x1	TXEM Transmit FIFO empty interrupt mask 1'b0: Transmit FIFO empty interrupt is not masked 1'b1: Transmit FIFO empty interrupt is masked Only valid in indirect access mode.
2	RW	0x1	TXOM Transmit FIFO overflow interrupt mask 1'b0: Transmit FIFO overflow interrupt is not masked 1'b1: Transmit FIFO overflow interrupt is masked Only valid in indirect access mode.
1	RW	0x1	RXUM Receive FIFO underflow interrupt mask 1'b0: Receive FIFO underflow interrupt is not masked 1'b1: Receive FIFO underflow interrupt is masked Only valid in indirect access mode.
0	RW	0x1	RXFM Receive FIFO full interrupt mask 1'b0: Receive FIFO full interrupt is not masked 1'b1: Receive FIFO full interrupt is masked Only valid in indirect access mode.

FSPI ICLR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	W1C	0x0	STPOLLC Status polling done interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the STPOLL.
7	W1C	0x0	DMAC DMA finish interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the DMAS

Bit	Attr	Reset Value	Description
6	W1 C	0x0	NSPIC NSPI error interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the NSPIS.
5	W1 C	0x0	AHBC AHB error interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the AHBS.
4	W1 C	0x0	TRANSC Transfer finish interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TRANSS.
3	W1 C	0x0	TXEC Transmit FIFO empty interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXES.
2	W1 C	0x0	TXOC Transmit FIFO overflow interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the TXOS.
1	W1 C	0x0	RXUC Receive FIFO underflow interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXUS.
0	W1 C	0x0	RXFC Receive FIFO full interrupt clear 1'b0: No action 1'b1: Clear interrupt Write "1" to clear the RXFS.

FSPI_FTLR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:8	RW	0x10	<p>RXFTLR Receive FIFO threshold level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ...</p> <p>When the number of receive FIFO entries is bigger than or equal to this value, the receive FIFO full interrupt is triggered.</p>
7:0	RW	0x10	<p>TXFTLR Transmit FIFO threshold level 8'h0: 0 entry level 8'h1: 1 entry level ... 8'h10: 16 entry level(default) ...</p> <p>When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.</p>

FSPI_RCVR

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	R/W SC	0x0	<p>RCVR FSPI recover Write any values to trigger the recovery of FSPI NSPI state machine, FIFO state and other logic state.</p>

FSPI_AX0

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>AX Auxiliary data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".</p>

FSPI_ABITO

Address: Operational Base + offset (0x0018)

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

Bit	Attr	Reset Value	Description
4:0	RW	0x00	ABIT Address bits extend 5'h0: 1bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.

FSPI ISR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	STPOLLS Status polling done interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
7	RO	0x0	DMAS DMA finish interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
6	RO	0x0	NSPIS NSPI transaction decode error interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
5	RO	0x0	AHBS AHB error interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
4	RO	0x0	TRANSS Transfer finish interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
3	RO	0x0	TXES Transmit FIFO empty interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
2	RO	0x0	TXOS Transmit FIFO overflow interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated
1	RO	0x0	RXUS Receive FIFO underflow interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated

Bit	Attr	Reset Value	Description
0	RW	0x0	RXFS Receive FIFO full interrupt status 1'b0: No interrupt 1'b1: Active interrupt generated

FSPI_FSR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:21	RO	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
12:8	RO	0x00	TXWLVL TX FIFO water level 5'h0: FIFO is full 5'h1: 1 entry is left ... 5'h10: 16 entry is left, FIFO is empty
7:4	RO	0x0	reserved
3	RO	0x0	RXFS Receive FIFO full status 1'b0: RX FIFO is not full 1'b1: RX FIFO is full
2	RO	0x0	RXES Receive FIFO empty status 1'b0: RX FIFO is not empty 1'b1: RX FIFO is empty
1	RO	0x0	TXES Transmit FIFO empty status 1'b0: TX FIFO is not empty 1'b1: TX FIFO is empty
0	RO	0x1	TXFS Transmit FIFO full status 1'b0: TX FIFO is not full 1'b1: TX FIFO is full

FSPI_SR

Address: Operational Base + offset (0x0024)

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

Bit	Attr	Reset Value	Description
0	RO	0x0	<p>SR Status register 1'b0: NSPI Controller is idle 1'b1: NSPI Controller is busy When controller is busy, don't change the setting of control register. When NSPI is idle, the software can initiate new transaction to external device.</p>

FSPI_RISR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	<p>STPOLLS Status polling done interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
7	RO	0x0	<p>DMAS DMA finish interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
6	RO	0x0	<p>NSPIS NSPI error interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
5	RO	0x0	<p>AHBS AHB error interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
4	RO	0x0	<p>TRANSS Transfer finish interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>

Bit	Attr	Reset Value	Description
3	RO	0x0	<p>TXES Transmit FIFO empty interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
2	RO	0x0	<p>TXOS Transmit FIFO overflow interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
1	RO	0x0	<p>RXUS Receive FIFO underflow interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>
0	RO	0x0	<p>RXFS Receive FIFO full interrupt status 1'b0: No active raw interrupt 1'b1: Active raw interrupt is generated Cleared by writing corresponding ICLR bit to clear raw interrupt status.</p>

FSPI VER

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0003	<p>VER The Version ID of FSPI</p>

FSPI QOP

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	<p>SO123BP SO123 bypass mode 1'b0: Disable bypass 1'b1: Enable bypass Default is enabled.</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	SO123 SO1 SO2 SO3 data value during inactive when CS is active 1'b0: Set to "0" 1'b1: Set to "1" The value of SO1, SO2 and SO3 during command and address bits input.

FSPI EXT CTRL

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x1	SR_GEN_MODE Status register generation mode 1'b0: Compatible mode with old controller 1'b1: Robust generation to indicates the status of controller If set to "1", the controller will only clear the SR bit after operation sequence done and CS is high.
13	RW	0x0	TRANS_INT_MODE Transmit done Interrupt generation mode 1'b0: Trigger NSPI end in data done 1'b1: Trigger NSPI end in CS inactive Default Generation is compatible with old controller.
12	RO	0x0	reserved
11:8	RW	0x0	SWITCH_IO_O2I_CNT Switch IO attribute cycles in O2I Idle phase 4'h0: 1st cycle 4'h1: 2nd cycle 4'h2: 3rd cycle 4'h3: 4th cycle ... 4'hf: 16th cycle The target cycle when switching from output to input in O2I idle phase.
7:4	RW	0x2	SWITCH_IO_DUMM_CNT Switch IO attribute cycles in dummy phase 4'h0: 1st cycle 4'h1: 2nd cycle 4'h2: 3rd cycle 4'h3: 4th cycle ... 4'hf: 16th cycle The target cycle when switching from output to input in Dummy data phase.

Bit	Attr	Reset Value	Description
3:0	RW	0x3	<p>CS_DESEL_CTRL CS inactive control 4'h0: 1 cycle 4'h1: 2 cycles 4'h2: 3 cycles 4'h3: 4 cycles ... 4'hf: 16 cycles</p> <p>The target cycles to hold CS inactive after de-assert the CS. Default value are 4 SCLK cycles that is enough for normal device.</p>

FSPI POLL CTRL

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	<p>ST_POLL_BIT_COMP_EN Status polling bit comparison enable 1'b0: The bit data is don't care 1'b1: The bit data should be equal to the expect data bit before completion Each bit has individual control enable.</p>
23:16	RW	0x00	<p>ST_POLL_EXPECT_DATA Status Polling Expect Data The target status data compared with read from device status register.</p>
15:8	RW	0x00	<p>ST_POLL_CMD PARA Status polling command parameter Target command for matching to do polling status. When the write command data matches with this configuration and ST_POLL_EN is set, the controller will do the polling mode until the expected status data is latched.</p>
7:2	RO	0x0	reserved
1	RW	0x0	<p>POLL_DLY_EN Status polling mode trigger delay enable 1'b0: Disable 1'b1: Enable To control the enable of polling mode trigger delay. If set to "0", the controller will transmit the read status immediately. The delay is controlled by POLLDLY_CTRL register.</p>
0	RW	0x0	<p>ST_POLL_EN Status polling mode enable 1'b0: Disable 1'b1: Enable If enabled, the application will continuous access the device status register within one command cycle.</p>

FSPI DLL CTRL0

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	SCLK_SMP_SEL SCLK sampling selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
7:0	RW	0x01	SMP_DLL_CFG Sample delay line configuration 8'h0: 1 DLL element cell 8'h1: 1 DLL element cell 8'h2: 2 DLL element cells ... 8'h255: 255 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI HRDYMASK

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	HRDYMASK AMBA HREADY mask control register 1'b0: Disable 1'b1: Enable Used to mask the hready low generation.

FSPI EXT AX

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0xf0	AX_SETUP_PAT Auxiliary setup data pattern The AX data pattern that setup the continuous/enhance/XIP read mode
7:0	RW	0xff	AX_CANCEL_PAT Auxiliary cancel data pattern The AX data pattern that cancel the continuous/enhance/XIP read mode.

FSPI SCLK INATM CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	SCLK_INATM_CNT SCLK inactive timeout counter When CS is active and SCLK is hold in high or low due to TX FIFO is empty or RX FIFO is full, if SCLK_INATM_EN is enabled, and timeout occurs, the controller will go back to idle and RX FIFO is flushed.

FSPI AUTO RF CNT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:0	RW	0xffffffff	AUTO_RF_CNT Auto refresh counter When CS is active and AUTO_RF_EN is enabled, if timeout occurs, the controller auto de-assert CS for some time to permit the device do refresh, then will continue the remained transaction.

FSPI XMMC WCMDO

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	WO	0x0	ADDRB Address bits 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy cycles 4'h0: No dummy cycles ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI XMMC RCMDO

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	WO	0x0	ADDRB Address bits 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy cycles 4'h0: No dummy cycles ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI XMMC CTRL

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x1	PFT_EN Prefetch enable 1'b0: Disable 1'b1: Enable Should disable prefetch if controller communicate with pSRAM which need refresh.
5	RW	0x1	DEV_HWEN Device hwrite enable 1'b0: Disable 1'b1: Enable
4:0	RO	0x0	reserved

FSPI MODE

Address: Operational Base + offset (0x005c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	XMMC_MODE_EN Memory-Mapped mode enable 1'b0: Disable, indirect access mode 1'b1: Enable, Memory-Mapped mode Before switching from indirect access mode to Memory-Mapped mode, the application should make sure the controller is in idle state and no pending transaction. If switch from Memory-Mapped to indirect access mode, software should initiate a dummy read by CPU before that.

FSPI DEVREGN

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x17	RSIZE Region size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes 5'd10: 1K bytes 5'd20: 1M bytes 5'd31: 4G bytes In Memory-Mapped mode, the CS is controlled by AHB address bus, region size is used to generate CS.

FSPI DEVSIZE0

Address: Operational Base + offset (0x0064)

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

Bit	Attr	Reset Value	Description
4:0	RW	0x12	<p>DSIZE Device Size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes 5'd10: 1K bytes 5'd20: 1M bytes 5'd31: 4G byte</p> <p>Device size is used to generate slop over status.</p>

FSPI_TME0

Address: Operational Base + offset (0x0068)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	<p>SCLK_INATM_EN SCLK inactive timeout enable 1'b0: Disable 1'b1: Enable</p>
0	RW	0x0	<p>AUTO_RF_EN Auto refresh enable 1'b0: Disable 1'b1: Enable</p>

FSPI_POLLDLY_CTRL

Address: Operational Base + offset (0x006c)

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>POLLDLY_IP Polling delay in progress 1'b0: No action 1'b1: Polling mode delay is in progress The application writes "1" to this bit to trigger the delay counter, when delay time elapsed, this bit is de-asserted.</p>
30:0	RW	0x0003ffff	<p>CNT Counter The target delay counter to trigger the polling command. The unit is based on HCLK.</p>

FSPI_DMATR

Address: Operational Base + offset (0x0080)

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

Bit	Attr	Reset Value	Description
0	W1 C	0x0	DMATR DMA trigger Write "1" to start the DMA transfer.

FSPI DMAADDR

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAADDR DMA address The destination or source data address in current system.

FSPI POLL DATA

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RO	0x0	POLL_STA Polling state 1'b0: Idle state 1'b1: Busy state This bit is set to 1 by controller if a polling command will be executed and cleared after polling done.
7:0	RO	0x00	POLL_DATA Polling data The polling data that read from external device, this data is updated if polling done. This data is valid only when POLL_BUSY is idle.

FSPI XMMCSR

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	R/W SC	0x0	SLOPOVER1 Slop over register for CS1 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.
0	R/W SC	0x0	SLOPOVER0 Slop over register for CS0 1'b0: Normal state 1'b1: Address slop over When the access address in memory map mode is bigger than DEVSIZE, this bit will be set. Write "1" to clear this bit.

FSPI CMD

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:30	WO	0x0	CS Device chip select. 2'b00: Chip select 0 2'b01: Chip select 1 2'b10: Reserved 2'b11: Reserved
29:16	WO	0x0000	TRB Total transfer bytes 14'd0: No data 14'd1: 1 Byte 14'd2: 2 Bytes ... Total Data Bytes number that will write to /read from the device.
15:14	WO	0x0	ADDRB Address bits 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register Address bits number select in indirect access mode. if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	WO	0x0	WR Write or read 1'b0: Read 1'b1: Write
11:8	WO	0x0	DUMM Dummy cycles 4'h0: No dummy cycles ... 4'h8: 8 cycles ... Dummy bit cycles in indirect access mode.
7:0	WO	0x00	CMD Command Command data in indirect access mode.

FSPI ADDR

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	ADDR Address register Indirect access start address data for current command sequence.

FSPI DATA

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DATA Data register Device data read or write from/to device.

FSPI CTRL1

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
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 Set this DATB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
11:10	RW	0x0	ADRB Address line width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: Reserved Set this ADRB to match the CMD sequence before doing indirect access mode and memory mapped access mode.
9:8	RW	0x0	CMDB Command line width 2'b00: 1bit, x1 mode 2'b01: 2bits, x2 mode 2'b10: 4bits, x4 mode 2'b11: Reserved Set this CMDB to match the CMD sequence before doing indirect access mode and memory mapped access mode.

Bit	Attr	Reset Value	Description
7:4	RW	0x0	<p>IDLE_CYCLE Idle cycles when switching IO from output to input 4'h0: Idle hold is disable 4'h1: Hold the SCLK in idle for 2 cycles when switch to shift in ... 4'hf: Hold the SCLK in idle for 16 cycles when switch to shift in To improve the transform IO timing, the application can set this register to hold the SCLK in low state or high state.</p>
3:2	RO	0x0	reserved
1	RW	0x0	<p>SHIFTPHASE Shift phase of data input in controller 1'b0: Shift input data at posedge SCLK 1'b1: Shift input data at negedge SCLK The application can select the input data captured by posedge of SCLK when "0" or negedge of SCLK when "1".</p>
0	RW	0x0	<p>SPI MODE Serial peripheral interface mode 1'b0: Mode 0 1'b1: Mode 3 SPI MODE is used to control the serial mode (CPOL and CPHA). CPOL indicates clock polarity of Serial master, CPOL=1 for SCLK high while idle, CPOL=0 for SCLK low while not transmitting. CPHA indicates clock phase. The combination of CPOL bit and CPHA bit decides which Serial mode is supported.</p>

FSPI_AX1

Address: Operational Base + offset (0x0214)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>AX Auxiliary data The AX value when doing the continuous read (enhance mode or XIP mode). That is M7-M0 in "Continuous Read Mode".</p>

FSPI_AB1T

Address: Operational Base + offset (0x0218)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>ABIT Address bits extend 5'h0: 1bit 5'h1: 2 bits ... 5'h1f: 32 bits Only valid when ADDRB is set to 2'b11.</p>

FSPI DLL CTRL1

Address: Operational Base + offset (0x023c)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	SCLK_SMP_SEL SCLK sampling selection 1'b0: Bypass DLL 1'b1: From DLL The sampling SCLK source selection.
7:0	RW	0x01	SMP_DLL_CFG Sample delay line configuration 8'h0: 1 DLL element cell 8'h1: 1 DLL element cell 8'h2: 2 DLL element cells ... 8'h255: 255 DLL element cells This register to control the sampling delay line cell used. The maximum DLL element cells is decided by process.

FSPI XMMC WCMD1

Address: Operational Base + offset (0x0250)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	WO	0x0	ADDRB Address bits 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy cycles 4'h0: No dummy cycles ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.

Bit	Attr	Reset Value	Description
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI XMMC RCMD1

Address: Operational Base + offset (0x0254)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:14	WO	0x0	ADDRB Address bits 2'b00: 0bits 2'b01: 24bits 2'b10: 32bits 2'b11: From the ABIT register Address bits number select in memory mapped mode, if there is not address command to send, set to zero.
13	WO	0x0	CONT Continuous 1'b0: Disable continuous mode 1'b1: Enable continuous mode AX mode or Continuous mode or XIP mode for device which begins with address.
12	RO	0x0	reserved
11:8	WO	0x0	DUMM Dummy cycles 4'h0: No dummy cycles ... 4'h8: 8 cycles ... Dummy bit cycles in memory mapped mode.
7:0	WO	0x00	CMD Command Command data in memory mapped access mode.

FSPI DEVSIZE1

Address: Operational Base + offset (0x0264)

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

Bit	Attr	Reset Value	Description
4:0	RW	0x12	<p>DSIZE Device size 5'd0: 1 byte 5'd1: 2 bytes 5'd2: 4 bytes 5'd10: 1K bytes 5'd20: 1M bytes 5'd31: 4G bytes</p> <p>Device size is used to generate slop over status.</p>

FSPI_TME1

Address: Operational Base + offset (0x0268)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	<p>SCLK_INATM_EN SCLK inactive timeout enable 1'b0: Disable 1'b1: Enable</p>
0	RW	0x0	<p>AUTO_RF_EN Auto refresh enable 1'b0: Disable 1'b1: Enable</p>

17.5 Interface Description

Table 17-2 FSPI0 interface description

Module Pin	Direction	Pin Name	IOMUX Setting
SCLK	O	FSPI0_CLK/GPIO1_A4	GRF_GPIO1A_IOMUX_H[3:0]=4'h1
CS#	O	FSPI0_CSn/GPIO1_A5	GRF_GPIO1A_IOMUX_H[7:4]=4'h1
SIO0	I/O	FSPI0_D0/GPIO1_A0	GRF_GPIO1A_IOMUX_L[3:0]=4'h1
SIO1	I/O	FSPI0_D1/GPIO1_A1	GRF_GPIO1A_IOMUX_L[7:4]=4'h1
SIO2	I/O	FSPI0_D2/GPIO1_A2	GRF_GPIO1A_IOMUX_L[11:8]=4'h1
SIO3	I/O	FSPI0_D3/GPIO1_A3	GRF_GPIO1A_IOMUX_L[15:12]=4'h1

Table 17-3 FSPI1 interface description

Module Pin	Direction	Pin Name	IOMUX Setting
SCLK	O	HyperBus_CKP/FSPI1_CLK(GPIO1_B0)	GRF_GPIO1B_IOMUX_L[3:0]=4'h2
CS#	O	HyperBus_CSn/FSPI1_CSn(GPIO1_B2)	GRF_GPIO1B_IOMUX_L[11:8]=4'h2
SIO0	I/O	HyperBus_D0/FSPI1_D0(GPIO1_B4)	GRF_GPIO1B_IOMUX_H[3:0]=4'h2
SIO1	I/O	HyperBus_D3/FSPI1_D1(GPIO1_B7)	GRF_GPIO1B_IOMUX_H[15:12]=4'h2
SIO2	I/O	HyperBus_D2/FSPI1_D2(GPIO1_B6)	GRF_GPIO1B_IOMUX_H[11:8]=4'h2
SIO3	I/O	HyperBus_D4/FSPI1_D3(GPIO1_C0)	GRF_GPIO1C_IOMUX_L[3:0]=4'h2

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

17.6 Application Notes

17.6.1 Typical Program Flow Without DMA

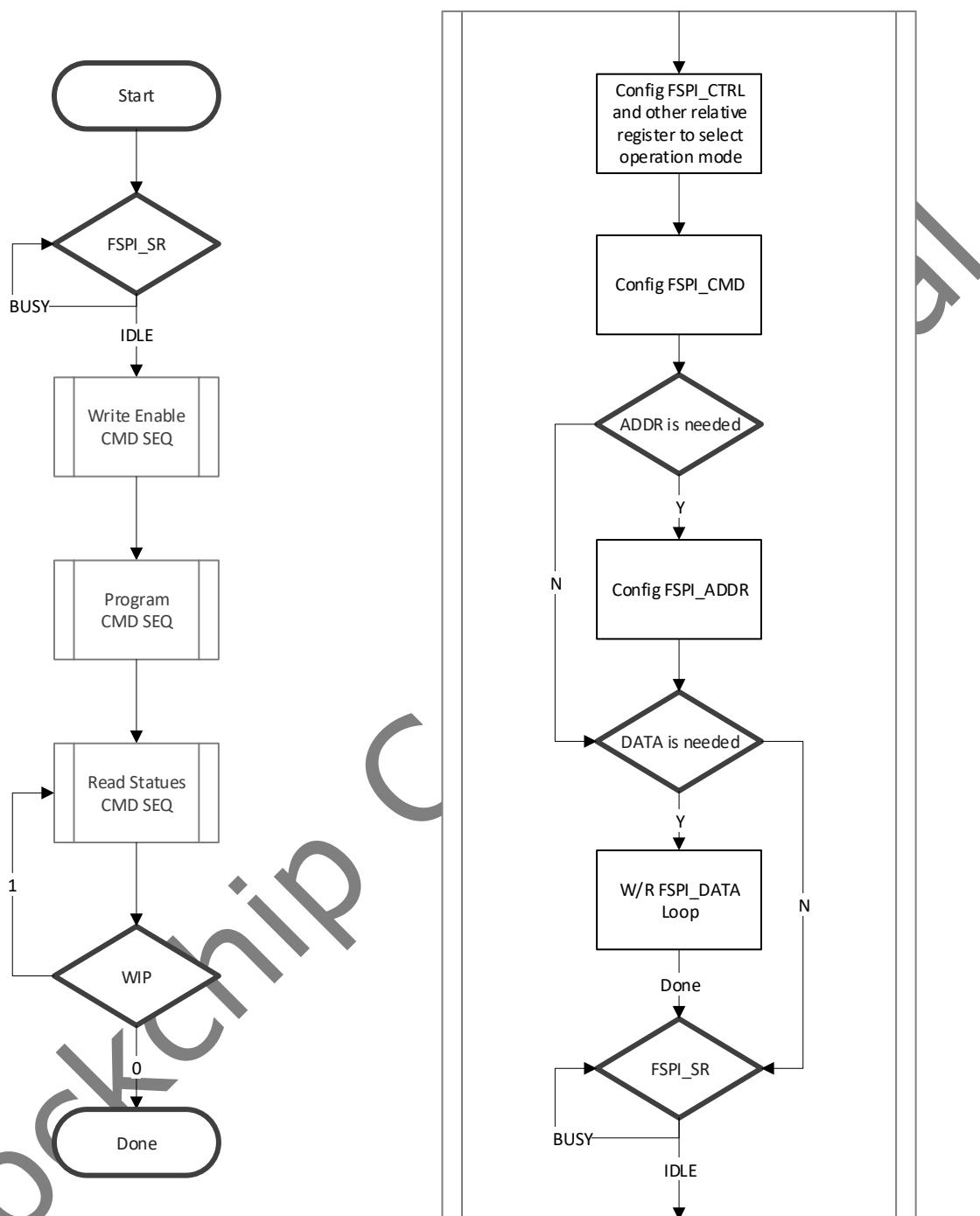


Fig. 17-2 Program Flow

All the AHB bus write data to FSPI_CMD, FSPI_ADDR and FSPI_DATA will be marked with different header and then pushed into transmit FIFO by writing order.

17.6.2 Typical READ Flow Without DMA

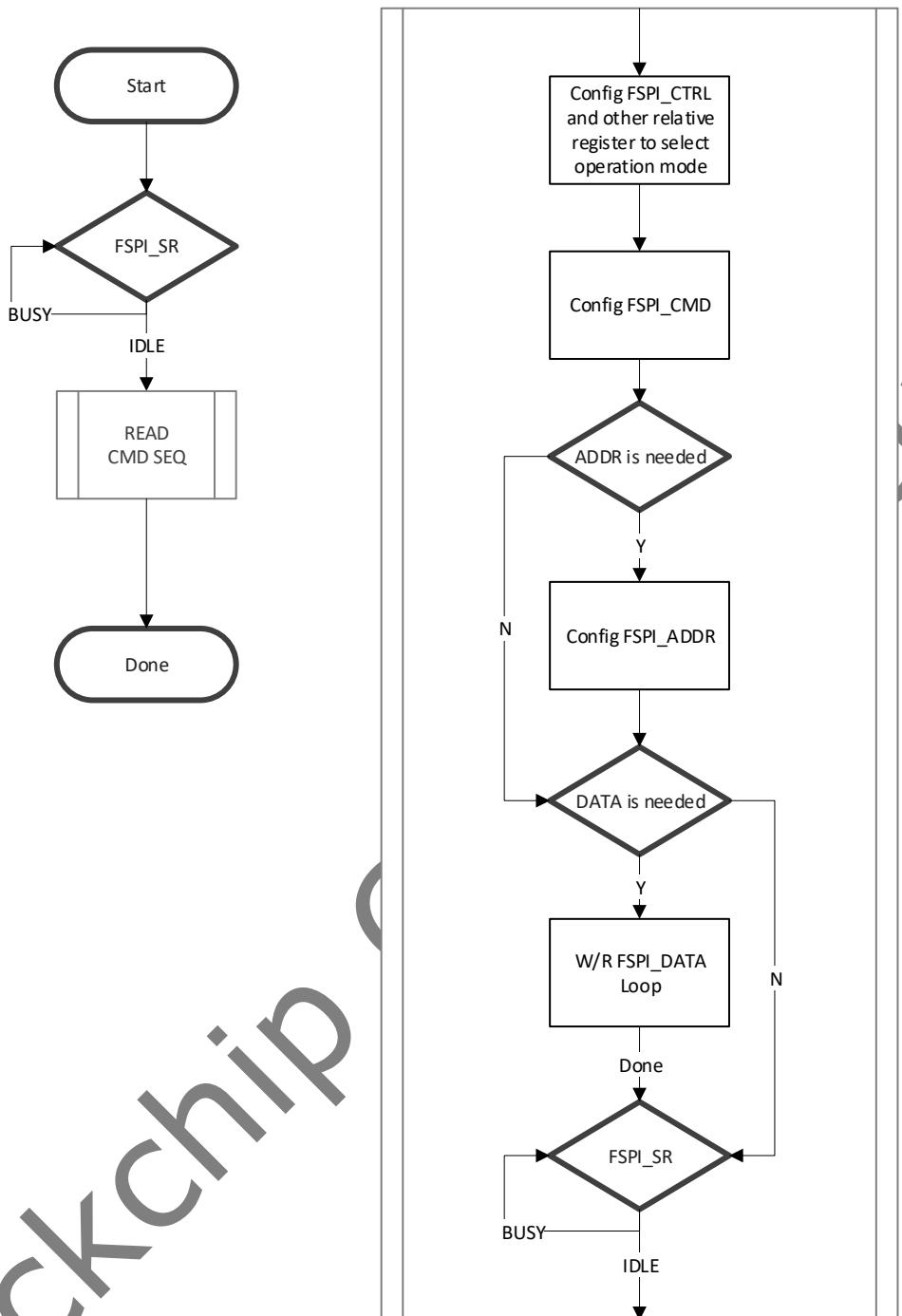


Fig. 17-3 Read Flow

17.6.3 Command Flow with DMA

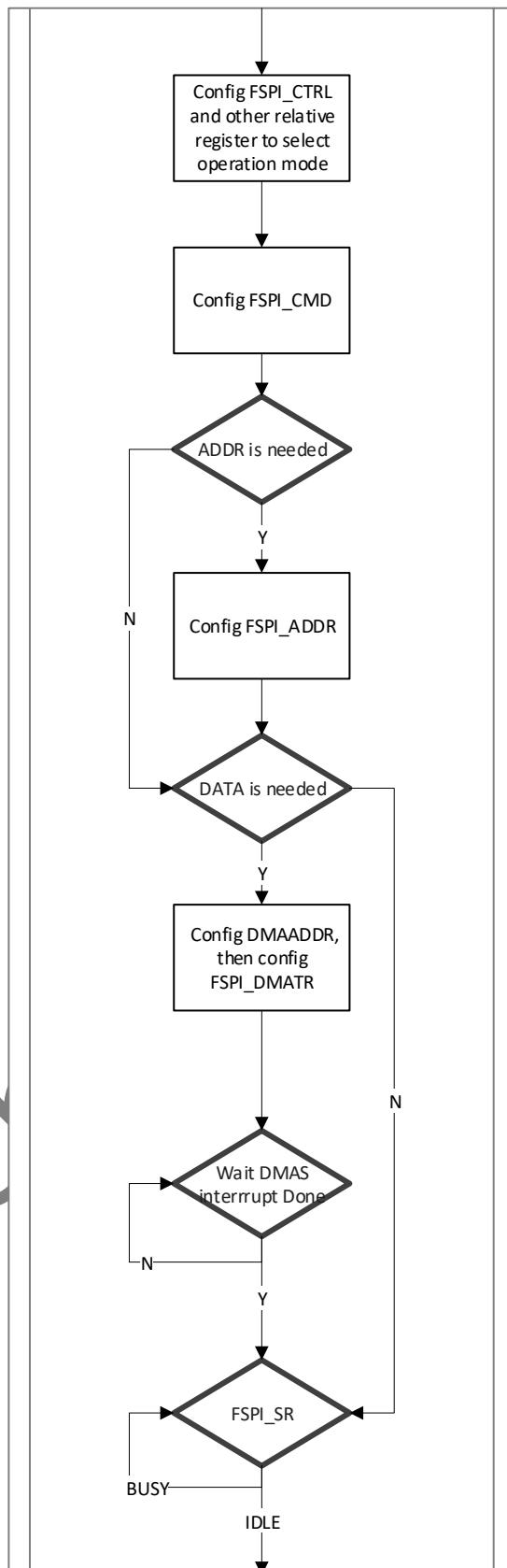


Fig. 17-4 Command with DMA Flow

The total transfer bytes is decided by TRB register in FSPI_CMD and must be aligned to 2 bytes.

17.6.4 SPI Mode and Shift Phase

The register SPIM in FSPI_CTRL will decide the default value of SCLK when CS# is inactive.

When SPIM=0, the default value is 0, means Mode 0. When SPIM=1, the default value is 1, means SPI Mode 3.

The register SHIFTPAHSE in FSPI_CTRL will decide when to sample the SIO data. If SHIFTPAHSE=0, it will sample the data at the posedge of sclk_out. If SHIFTPHASE=1, it will sample the data at the negedge of sclk_out.

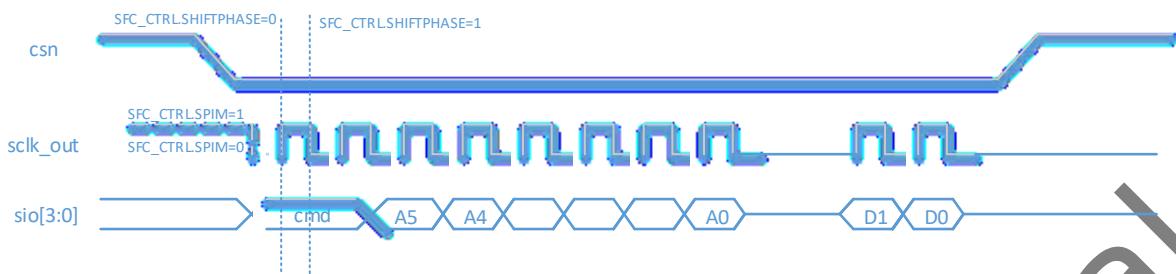


Fig. 17-5 SPI mode

17.6.5 Idle Cycles

The FSPI_CTRL register is a global control register, when the controller is in busy state (FSPI_SR), FSPI_CTRL cannot be set. The field IDLE_CYCLE (FSPI_CTRL[7:4]) of this register are used to configure the idle level cycles of FSPI core clock (sclk) before reading the first bit of the read command.

Like the following picture shows: the red line of the sclk is the idle cycles, during these cycles, the chip pad is switched to output. When IDLE_CYCLE =0, it means there will be not idle level cycles.

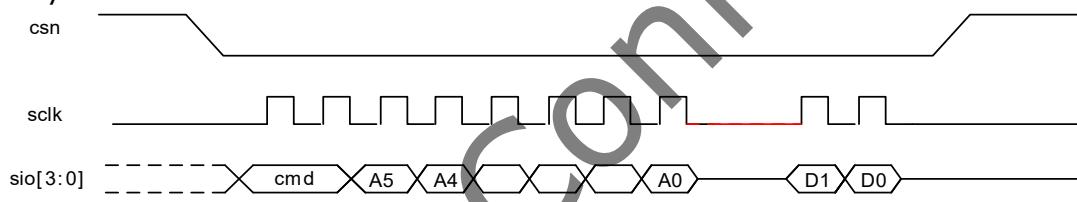


Fig. 17-6 Idle cycles

17.6.6 Memory-Mapped Mode

After the Controller is configured as Memory-Mapped mode, the normal operation mode is not allowed which means indirect command transaction is forbidden unless software configures the FSPI_MODE back to indirect access mode.

Before switching into Memory-Mapped mode, the software should initiate some transactions to configure the external device, such as Quad IO enable and IO Drive Strength.

In Memory-Mapped mode, it supports read transaction from serial NOR Flash and serial pSRAM, and the write transaction is only support by serial pSRAM.

Chapter 18 HyperBus

18.1 Overview

The HyperBus module provides function and operation for interfacing to the HyperBus memory devices. The HyperBus achieves high speed read/write throughput by double data rate interface

The main feature is shown below

- Support operational frequency up to 166 MHz
- Achieve maximum 333 MB/s data throughput by 8 bits bus and few timing signals only
- Support the double data rate interface
- Support maximum 128MB address space
- Support one slave devices
- Support XIP operation
- Support configurable RWDS input clock delay line

18.2 Block Diagram

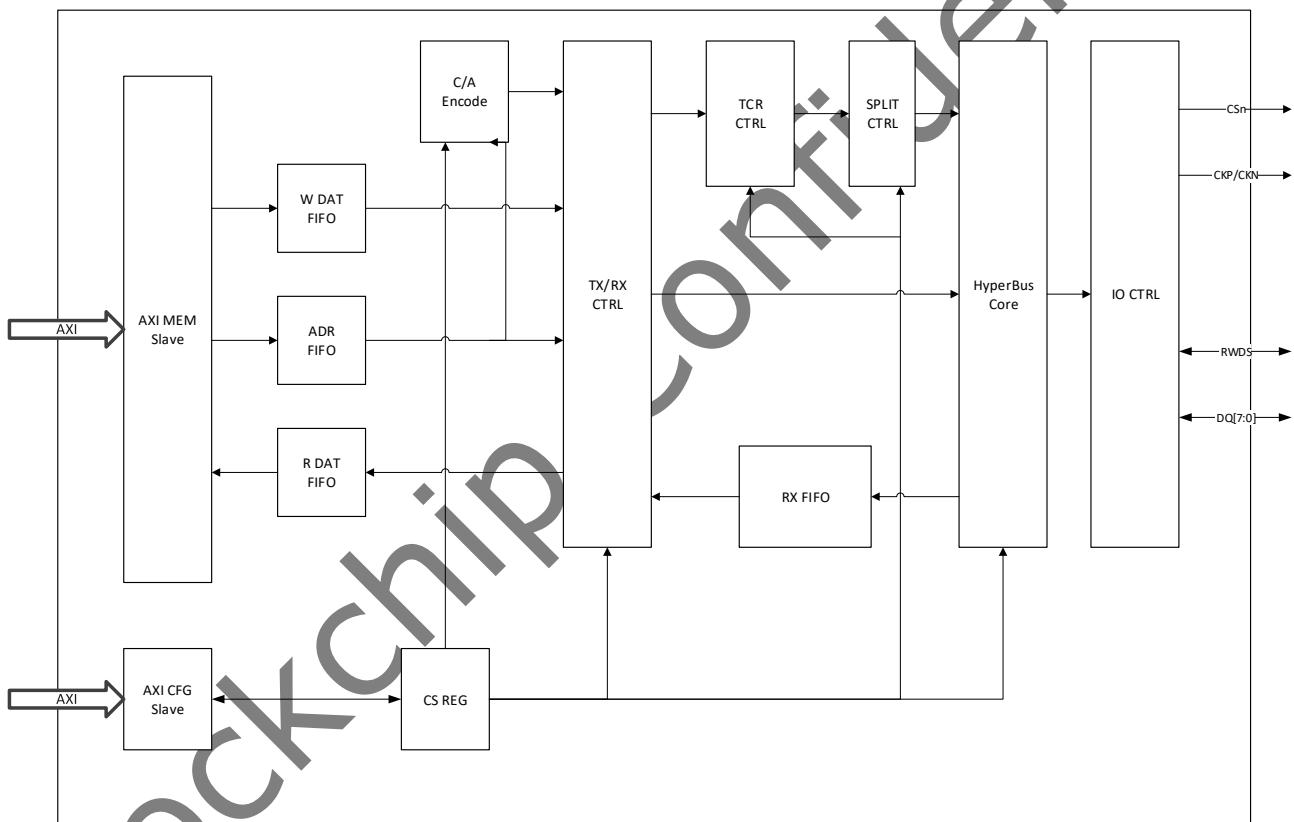


Fig. 18-1 HyperBus Architecture

18.3 Function Description

The HyperBus module is high throughput memory interface. It is a part of the device FSS and provides execute in place (XIP) and block copy access to HyperBus memory devices (HyperRAM and HyperFlash). The HyperBus module is compliant to the Cypress HyperBus™ Specification

18.3.1 HyperBus Core

This section describes physical interface of HyperBus module. The HyperBus core generates HyperBus timing from control signals

18.3.1.1 Read

HyperBus Core module asserts CSn, CKP/CKN, and DQ. Then, it outputs 3 words command/address, and reads 1 or more data by the timing of RWDS. If both RX FIFO and

R DAT FIFO become full, HyperBus Core ends the access. Then, when only RX FIFO becomes empty, HyperBus Core re-initiates the access. At this time, R DAT FIFO is transferring read data continuously to AXI bus. The HyperBus Core puts the transaction request from AXI bus, such as the burst type and the burst length, to the C/A cycle on HyperBus as is shown below

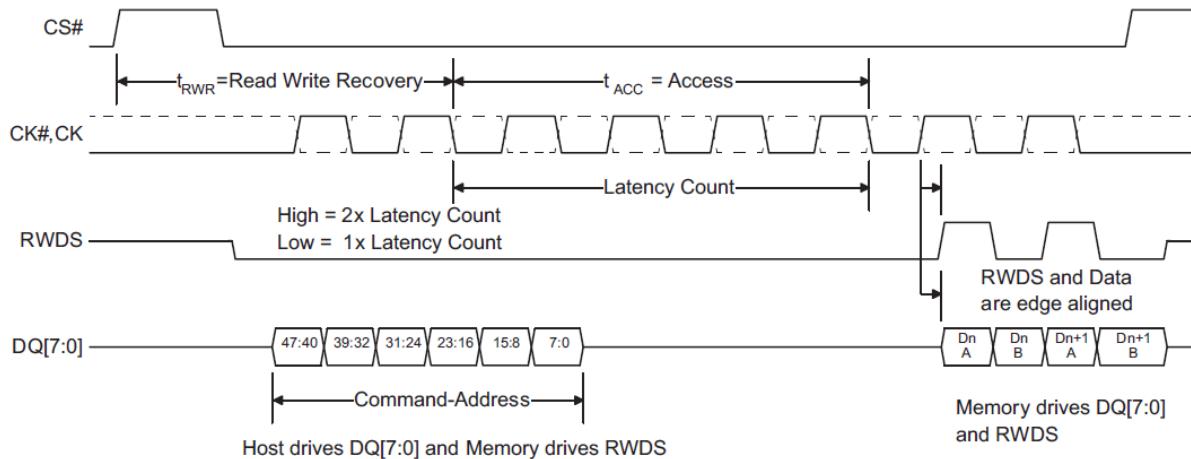


Fig. 18-2 HyperRAM Read Waveform

18.3.1.2 Write

This section describes the write operation of HyperBus Core. The HyperBus Core asserts CSn, CKP/CKN, and DQ. Then, it outputs 3 words command/address, and writes 1 or more data. For HyperRAM write operation, there is initial clock latency before writing data.

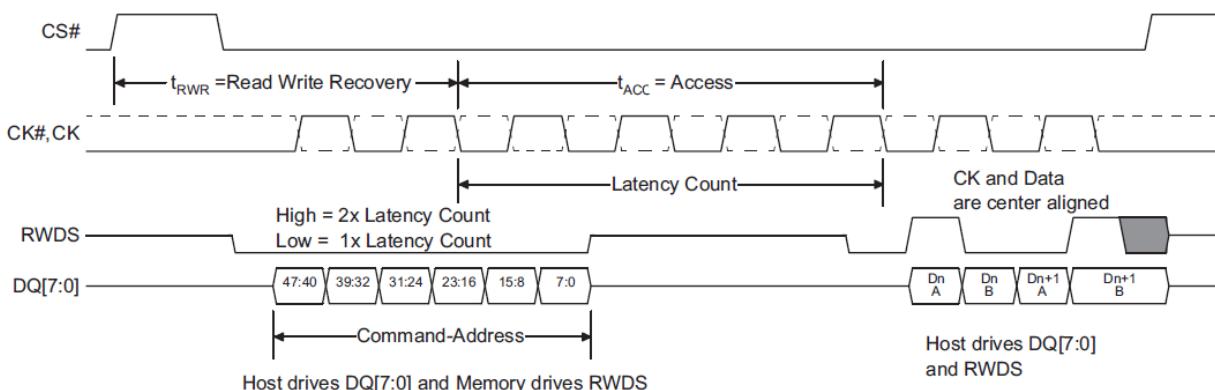


Fig. 18-3 HyperRAM Write Waveform

18.3.1.3 Byte Mask

This byte mask is function for only HyperRAM device (changed by MBRx.DEVTYPE). HyperBus Memory Interface Controller outputs RWDS signal as a byte mask during write operation. When byte access is requested from Main Controller, not write data is masked by RWDS=HIGH. When target device is HyperFlash, byte masked data is written to device because of only supporting 16-bit access (byte masked data is fill in all 1 bit). In the HyperRAM Write Waveform, the D_n A and D_{n+1} B is invalid data which is masked.

18.3.2 TX/RX Controller

The TX/RX Controller performs operation control by state machine, and flow control of data between AXI bus and HyperBus. In addition, when next access address is accepted by multiple outstanding address on AXI bus, it passes the transaction to the TCR CTRL if the address of subsequent access is continuous address to the present access.

18.3.3 C/A Encoder

HyperBus memory device uses 6 bytes of command/address information to define the transactions characteristics. Table below shows the assignments of CA bit in HyperBus Memory Interface Controller

Table 18-1 C/A Format of HyperBus

C/A Bit	Name	Assignment
47	R/W#	0: Write, 1: Read
46	Target	CRT in HYPERBUS_MCRn
45	Burst Type	0: WRAP, 1: INCR
44	RFU	0
43-16	Page Address	Address [31:4]
15	RFU	0
14-13	RFU	3
12-3	RFU	0
2-0	Column Address	Address [3:1]

18.3.4 FIFO

This section describes the FIFO for processing transaction request

18.3.4.1 ADR FIFO

ADR FIFO is a synchronizer module from AXI clock domain to HyperBus clock domain. This FIFO has 46-bit width. This FIFO stores the control data which includes address, length, burst type and r/w flag. From these data, the control signals are generated for Main Controller.

18.3.4.2 W DAT FIFO

The W DAT FIFO is composed of 40-bit width x 128 steps. This FIFO is used to store the write data, valid information and strobe information written from the master in order to receive the command sequence of write buffer programming without a wait on AXI interface. The valid information indicates the 8-bit data is valid or not. The strobe information is used for strobe data from AXI interface to HyperBus Core. When valid and strobe is 1, the 8-bit data is written data

18.3.4.3 R DAT FIFO

The R DAT FIFO is composed of 40-bit width x 128 steps. This FIFO is used to store the data read from RX FIFO and the error detection information, and data stored is outputted according to timing of ACLK on AXI interface. This error information is transferred to HOST by RRESP signals.

18.3.4.4 RX FIFO

The RX FIFO is composed of 20-bit width x 256 steps. This FIFO is used to store the data which reads from DQIN FIFO

18.4 Register Description

18.4.1 Internal Address Mapping

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

Table 18-2 HyperBus Address Mapping Table

Name	Address Base	Size
HyperBus CFG	0x40300000	64KB
HyperBus MEM	0x18000000/0x38000000	128MB

For HyperBus MEM regions access, it's exclusive with FSPI1 MEM regions access. The Application should configure the GRF_SOC_CON0.grf_con_hyperx8_sfc1_sel to select these regions mapped as HyperBus MEM address.

18.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
HYPERBUS_CSR	0x0000	W	0x00000000	Controller Status Register
HYPERBUS_IEN	0x0004	W	0x80000000	Interrupt Enable
HYPERBUS_ISR	0x0008	W	0x00000000	Interrupt Status Register
HYPERBUS_MBR0	0x0010	W	0x00000000	Reserved (CS0# Memory Base Address)
HYPERBUS_MBR1	0x0014	W	0x00000000	Reserved (CS1# Memory Base Address)
HYPERBUS_MCR0	0x0020	W	0x00000003	CS0# Memory Configuration Register
HYPERBUS_MCR1	0x0024	W	0x00000003	CS1# Memory Configuration Register
HYPERBUS_MTR0	0x0030	W	0x00000001	CS0# Memory Base Address
HYPERBUS_MTR1	0x0034	W	0x00000001	CS1# Memory Base Address
HYPERBUS_GPOR	0x0040	W	0x00000000	General Purpose Output Register
HYPERBUS_WPR	0x0044	W	0x00000000	Write Protection
HYPERBUS_LBR	0x0048	W	0x00000000	Loopback Register
HYPERBUS_TAR	0x004c	W	0x00000000	Transaction Allocation Register
HYPERBUS_RWDSIC	0x0050	W	0x00000000	RWDS Input Control Register
HYPERBUS_CA2RSVD	0x0054	W	0x00000000	CA2 Data Control Register
HYPERBUS_SPCSR	0x0058	W	0x00000000	Special Control Register

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

18.4.3 Detail Register Description

HYPERBUS_CSR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RO	0x0	<p>WRSTOERR Write RSTO error. This bit indicates whether HyperBus memory is under reset state in the latest write operation.</p> <p>1'b0: Normal operation 1'b1: HyperBus memory is under reset</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI SLVERR.</p>
25	RO	0x0	<p>WTRSERR Write Transaction Error. This bit indicates whether AXI protocol is acceptable by HyperBus Memory Controller IP in the latest write transaction.</p> <p>1'b0: Normal operation 1'b1: This protocol is not supported</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI SLVERR.</p>

Bit	Attr	Reset Value	Description
24	RO	0x0	<p>WDECERR Write Decode Error. This bit indicates whether access address is acceptable in the latest write transaction.</p> <p>1'b0: Normal operation 1'b1: Access address is not reachable</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI DECERR.</p>
23:17	RO	0x0	reserved
16	RO	0x0	<p>WACT Write is Active. This bit indicates whether write transaction is in progress or not.</p> <p>1'b0: Write is idle 1'b1: Write is active</p> <p>When receiving write request on write address channel, this bit becomes 1. When retrieving response signaling on write response channel, this bit becomes 0.</p>
15:12	RO	0x0	reserved
11	RO	0x0	<p>RDSSTALL RDS Stall. This bit indicates whether read data transfer from HyperBus memory is stalled (RDS remains LOW) in the latest read transaction.</p> <p>1'b0: Normal operation 1'b1: RDS is stalled</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI SLVERR.</p>
10	RO	0x0	<p>RRSTOERR Read RSTO error. This bit indicates whether HyperBus memory is under reset state in the latest read operation.</p> <p>1'b0: Normal operation 1'b1: HyperBus memory is under reset</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI SLVERR.</p>
9	RO	0x0	<p>RTRSERR Read Transaction Error. This bit indicates whether AXI protocol is acceptable by HyperBus Memory Controller IP in the latest read transaction.</p> <p>1'b0: Normal operation 1'b1: This protocol is not supported</p> <p>When this bit is set, HyperBus Memory Controller IP responds by AXI SLVERR.</p>

Bit	Attr	Reset Value	Description
8	RO	0x0	RDECERR Read Decode Error. This bit indicates whether access address is acceptable in the latest read transaction. 1'b0: Normal operation 1'b1: Access address is not reachable When this bit is set, HyperBus Memory Controller IP responds by AXI DECERR.
7:1	RO	0x0	reserved
0	RO	0x0	RACT Read is Active. This bit indicates whether read transaction is in progress or not. 1'b0: Read is idle 1'b1: Read is active When receiving read request on read address channel, this bit becomes 1. When retrieving all requested data on read data channel, this bit becomes 0.

HYPERBUS IEN

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31	RW	0x1	NTP Interrupt Polarity Control. This bit is used to choose the polarity of optional interrupt signal (IENOn). 1'b0: IENOn signal is active low 1'b1: IENOn signal is active high
30:1	RO	0x0	reserved
0	RW	0x0	RPCINTE HyperBus Memory Interrupt Enable. 1'b0: Disable interrupt 1'b1: Enable interrupt by INT# signal of HyperBus memory

HYPERBUS ISR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	RPCINTS HyperBus Memory Interrupt. 1'b0: No interrupt. 1'b1: This bit displays interrupt from INT# signal of HyperBus memory.

HYPERBUS MBR0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	BADDR The base address of addressable region to HyperBus memory is set up. Since register can be set in 16M bytes boundary, lower 24 bit is fixed to 0.

HYPERBUS MBR1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	BADDR The base address of addressable region to HyperBus memory is set up. Since register can be set in 16M bytes boundary, lower 24 bit is fixed to 0.

HYPERBUS MCRO

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31	RW	0x0	MAXEN Maximum length Enable 1'b0: No configurable CS# low time 1'b1: Configurable CS# low time When this bit 1, CS# low time can be configurable by MAXLEN bit.
30:27	RO	0x0	reserved
26:18	RW	0x000	MAXLEN Maximum Length This bit indicates maximum read/write transaction length to memory. This bit is ignored when MAXEN bit is 0. 9'h000: 2 Byte (1 HyperBus CK) 9'h001: 4 Byte (2 HyperBus CK) 9'h002: 6 Byte (3 HyperBus CK) 9'h1ff: 1024 Byte (512 HyperBus CK)
17	RW	0x0	TCMO True Continuous Merge Option. 1'b0: No merging WRAP and INCR 1'b1: Merging WRAP and INCR Note that this function can be used with the HyperBus memory with specific function. Please confirm whether it is corresponding HyperBus memory before enabling this function. When HyperBus memory doesn't accept the 8-bit boundary address, and a wrapping burst access with ARSIZE=0 and ARADDR0=1 is used, this bit must not be set to 1.

Bit	Attr	Reset Value	Description
16	RW	0x0	ACS Asymmetry Cache Support. 1'b0: No asymmetry cache system support. 1'b1: Asymmetry cache system support. This function should be disabled if the HyperBus memory itself supports the asymmetry cache system.
15:6	RO	0x0	reserved
5	RW	0x0	CRT Configuration Register Target. 1'b0: Memory space 1'b1: CR space This bit indicates whether the read or write operation accesses the memory or CR space. This bit is mapped to CA[46] bit in HyperRAM. When using HyperFlash, this bit should be set to 0.
4	RW	0x0	DEVTYPE Device Type. 1'b0: HyperFlash 1'b1: HyperRAM Device type for control target.
3:2	RO	0x0	reserved
1:0	RW	0x3	WRAPSIZE Wrap Size. 2'b00: Reserved 2'b01: 64 Bytes 2'b10: 16 Bytes 2'b11: 32 Bytes The wrap burst length of HyperBus memory. This bit is ignored when the asymmetry cache support bit is 0. When the asymmetry cache support is 1, this bit should be set the same as wrap size of configuration register in HyperBus memory.

HYPERBUS MCR1

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31	RW	0x0	MAXEN Maximum length Enable 1'b0: No configurable CS# low time 1'b1: Configurable CS# low time When this bit 1, CS# low time can be configurable by MAXLEN bit.
30:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26:18	RW	0x000	<p>MAXLEN Maximum Length This bit indicates maximum read/write transaction length to memory.</p> <p>This bit is ignored when MAXEN bit is 0.</p> <p>9'h000: 2 Byte (1 HyperBus CK) 9'h001: 4 Byte (2 HyperBus CK) 9'h002: 6 Byte (3 HyperBus CK) 9'h1ff: 1024 Byte (512 HyperBus CK)</p>
17	RW	0x0	<p>TCMO True Continuous Merge Option. 1'b0: No merging WRAP and INCR 1'b1: Merging WRAP and INCR</p> <p>Note that this function can be used with the HyperBus memory with specific function. Please confirm whether it is corresponding HyperBus memory before enabling this function.</p> <p>When HyperBus memory doesn't accept the 8-bit boundary address, and a wrapping burst access with ARSIZE=0 and ARADDR0=1 is used, this bit must not be set to 1.</p>
16	RW	0x0	<p>ACS Asymmetry Cache Support. 1'b0: No asymmetry cache system support. 1'b1: Asymmetry cache system support.</p> <p>This function should be disabled if the HyperBus memory itself supports the asymmetry cache system.</p>
15:6	RO	0x0	reserved
5	RW	0x0	<p>CRT Configuration Register Target. 1'b0: Memory space 1'b1: CR space</p> <p>This bit indicates whether the read or write operation accesses the memory or CR space. This bit is mapped to CA[46] bit in HyperRAM. When using HyperFlash, this bit should be set to 0.</p>
4	RW	0x0	<p>DEVTYPE Device Type. 1'b0: HyperFlash 1'b1: HyperRAM</p> <p>Device type for control target.</p>
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x3	<p>WRAPSIZE Wrap Size. 2'b00: Reserved 2'b01: 64 Bytes 2'b10: 16 Bytes 2'b11: 32 Bytes</p> <p>The wrap burst length of HyperBus memory. This bit is ignored when the asymmetry cache support bit is 0. When the asymmetry cache support is 1, this bit should be set the same as wrap size of configuration register in HyperBus memory.</p>

HYPERBUS_MTR0

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	<p>RCSHI Read Chip Select High Between Operations. This bit indicates CS# high time for read between operations. 0x0 corresponds to 1.5 clock cycle, 0xF corresponds to 16.5 clock cycle.</p>
27:24	RW	0x0	<p>WCShI Write Chip Select High Between Operations. This bit indicates CS# high time for write between operations. 0x0 corresponds to 1.5 clock cycle, 0xF corresponds to 16.5 clock cycle.</p>
23:20	RW	0x0	<p>RCSS Read Chip Select Setup to next CK rising edge. This bit indicates CS# setup time for read from CS# assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
19:16	RW	0x0	<p>WCSS Write Chip Select Setup to next CK rising edge. This bit indicates CS# setup time for write from CS# assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
15:12	RW	0x0	<p>RCSH Read Chip Select Hold after CK falling edge. This bit indicates CS# hold time for read to CS# de-assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
11:8	RW	0x0	<p>WCSh Write Chip Select Hold after CK falling edge. This bit indicates CS# hold time for write to CS# de-assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	<p>LTCY Latency Cycle. Only uses in HyperRAM.</p> <p>This bit indicates initial latency code for read/write access. This bit is ignored when MCRX.DEVTYPE is 0 (HyperFlash).</p> <p>4'h0: 5 clock latency 4'h1: 6 clock latency 4'h2: 7 clock latency 4'h3~4'hd: Reserved 4'he: 3 clock latency 4'hf: 4 clock latency</p>

HYPERBUS MTR1

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	<p>RCSHI Read Chip Select High Between Operations.</p> <p>This bit indicates CS# high time for read between operations. 0x0 corresponds to 1.5 clock cycle, 0xF corresponds to 16.5 clock cycle.</p>
27:24	RW	0x0	<p>WCSHI Write Chip Select High Between Operations.</p> <p>This bit indicates CS# high time for write between operations. 0x0 corresponds to 1.5 clock cycle, 0xF corresponds to 16.5 clock cycle.</p>
23:20	RW	0x0	<p>RCSS Read Chip Select Setup to next CK rising edge.</p> <p>This bit indicates CS# setup time for read from CS# assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
19:16	RW	0x0	<p>WCSS Write Chip Select Setup to next CK rising edge.</p> <p>This bit indicates CS# setup time for write from CS# assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
15:12	RW	0x0	<p>RCSH Read Chip Select Hold after CK falling edge.</p> <p>This bit indicates CS# hold time for read to CS# de-assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
11:8	RW	0x0	<p>WCSH Write Chip Select Hold after CK falling edge.</p> <p>This bit indicates CS# hold time for write to CS# de-assertion. 0x0 corresponds to 1 clock cycle, 0xF corresponds to 16 clock cycle.</p>
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x1	<p>LTCY Latency Cycle. Only uses in HyperRAM.</p> <p>This bit indicates initial latency code for read/write access. This bit is ignored when MCRX.DEVTYPE is 0 (HyperFlash).</p> <p>4'h0: 5 clock latency 4'h1: 6 clock latency 4'h2: 7 clock latency 4'h3~4'hd: Reserved 4'he: 3 clock latency 4'hf: 4 clock latency</p>

HYPERBUS GPOR

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1:0	RW	0x0	<p>GPO General Purpose Output interface.</p> <p>1'b0: Output signal polarity is LOW 1'b1: Output signal polarity is HIGH</p>

HYPERBUS WPR

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	<p>WP Write Protection Control.</p> <p>1'b0: Not Protected. WP# signal is HIGH. 1'b1: Protected. WP# signal is LOW.</p>

HYPERBUS LBR

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	<p>LOOPBACK The write transaction data written on AXI bus is looped back as the read data from RPC bus. The loopback is performed between W DAT FIFO and R DAT FIFO in AXI interface controller.</p> <p>1'b0: Disable loopback. 1'b1: Enable loopback.</p>

HYPERBUS TAR

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	RTA Read Transaction Allocation. 2'b00: 1 read transaction 2'b01: 2 read transactions 2'b10: 3 read transactions 2'b11: 4 read transactions
3:2	RO	0x0	reserved
1:0	RW	0x0	WTA Write Transaction Allocation. 2'b00: 1 write transaction 2'b01: 2 write transactions 2'b10: 3 write transactions 2'b11: 4 write transactions

HYPERBUS RWDSIC

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	rxend_ctrl RWDS input clk end timing ctrl. 1'b0: Standard control mode 1'b1: Enhance control mode
0	RW	0x0	rxstart_ctrl RWDS input clk start timing ctrl. 1'b0: Standard control mode 1'b1: Enhance control mode

HYPERBUS CA2RSVD

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:3	RW	0x0000	ca2_data Reserved data for CA2[15:3]. Default is all "0".
2:0	RO	0x0	reserved

HYPERBUS SPCSR

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>W955D8_CON</p> <p>If Winbond W955D8 Device is used. Below configuration is for compatible operation</p> <p>2'b00: Standard operation</p> <p>2'h11: W955D8 specific</p> <p>Others: Reserved</p>

18.5 Interface Description

Table 18-3 FSPI1 interface description

Module Pin	Direction	Pin Name	IOMUX Setting
CKP	O	HyperBus_CKP/FSPI1_CLK(GPIO1_B0)	GRF_GPIO1B_IOMUX_L[3:0]=4'h1
CKN	O	HyperBus_CKN(GPIO1_B1)	GRF_GPIO1B_IOMUX_L[7:4]=4'h1
CS#	O	HyperBus_CSn/FSPI1_CSn(GPIO1_B2)	GRF_GPIO1B_IOMUX_L[11:8]=4'h1
RWDS	I/O	HyperBus_RWDS(GPIO1_B3)	GRF_GPIO1B_IOMUX_L[15:12]=4'h1
DQ0	I/O	HyperBus_D0/FSPI1_D0(GPIO1_B4)	GRF_GPIO1B_IOMUX_H[3:0]=4'h1
DQ1	I/O	HyperBus_D1(GPIO1_B5)	GRF_GPIO1B_IOMUX_H[3:0]=4'h1
DQ2	I/O	HyperBus_D2/FSPI1_D2(GPIO1_B6)	GRF_GPIO1B_IOMUX_H[11:8]=4'h1
DQ3	I/O	HyperBus_D3/FSPI1_D1(GPIO1_B7)	GRF_GPIO1B_IOMUX_H[15:12]=4'h1
DQ4	I/O	HyperBus_D4/FSPI1_D3(GPIO1_C0)	GRF_GPIO1C_IOMUX_L[3:0]=4'h1
DQ5	I/O	HyperBus_D5(GPIO1_C1)	GRF_GPIO1C_IOMUX_L[7:4]=4'h1
DQ6	I/O	HyperBus_D6(GPIO1_C2)	GRF_GPIO1C_IOMUX_L[11:8]=4'h1
DQ7	I/O	HyperBus_D7(GPIO1_C3)	GRF_GPIO1C_IOMUX_L[15:12]=4'h1

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

18.6 Application Notes

18.6.1 Transaction Arbitration

When read and write request from AXI interface stacks, the transaction arbitration is executed. The allocation of read and write transaction can be varied by the TAR.RTA and TAR.WTA register settings. If there is a read request and there is no write request, the read request passes through this arbiter on any TAR.RTA and TAR.WTA conditions. If there are four read requests and there is three write requests on TAR.RTA is set to "1" (Allocation of read is "2") and TAR.WTA is set to "0" (Allocation of write is "1"), these requests align to two read requests, one write request, two read requests and remaining write requests from the front. It might select the write request for first transaction because the read/write

selection derives from the previous transaction. In this case, it alights to one write request, two read requests, one write requests and remaining (one) write request.

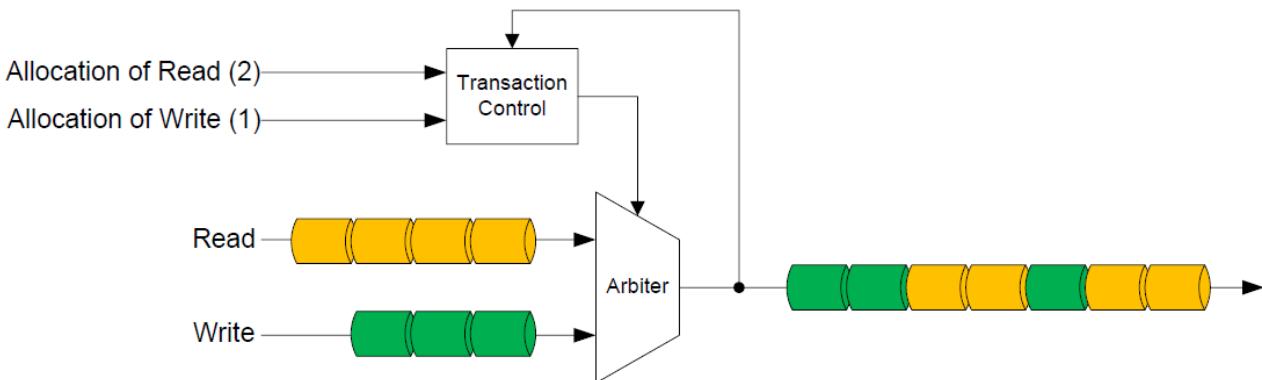


Fig. 18-4 Transaction Arbitration

18.6.2 Chip Select selection

This section describes how to select Chip Select (CS0# or CS#1). This controller can't assert two Chip Select at the same time.

- CS0# select condition:

MBR0 address <= AxADDR < MBR1 address

Address for Main Controller = AxADDR - MBR0 address

- CS1# select condition:

MBR1 address <= AxADDR

Address for Main Controller = AxADDR - MBR1 address

When MBR1 address is equal to or lower than MBR0 address, CS1# is not available.

18.6.3 Configurable CS# Low Time

This section describes configurable CS# low time. HyperRAM has a regulation for CS# low time (tCSM). To meet CS# low time, a read/write transaction can be regulated by setting MCRx.MAXLEN bit. By MCRx.MAXLEN bit, one transaction is split multi transactions.

MAXEN=0



MAXEN=1 & MAXLEN=127

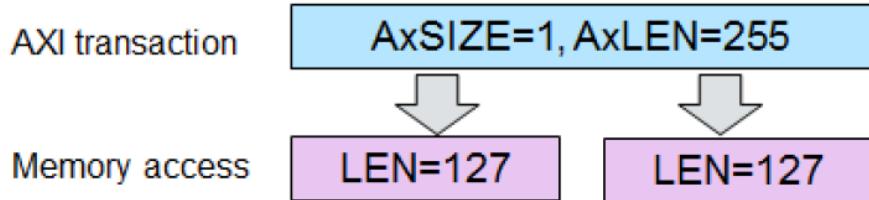


Fig. 18-5 Separated Transaction

18.6.4 True Continuous Operation

This section describes operation of the true continuous read. This function supports both HyperFlash and HyperRAM.

- True continuous operation removes address output period and latency cycles for

- subsequent transaction.
- When HyperBus Memory Controller IP accepts subsequent read transaction with continuous address during read transaction operation, it is merged to one CS# period on HyperBus memory interface.
- If both RX FIFO in HyperBus memory interface controller and R DAT FIFO in AXI interface controller become full, HyperBus memory interface controller ends the access (CS# on HyperBus memory interface is de-asserted). Then, when only RX FIFO becomes empty, HyperBus memory interface controller initiates the access. (CS# is asserted again.) At this time, AXI interface controller is transferring read data continuously in R DAT FIFO.
- HyperBus memory interface controller merges some read transactions by the following condition.
 - Between the sequential read transactions which have INCR as the type of the burst and have the continuous address are merged.
 - Between the read transaction with WRAP and the subsequent read transaction with INCR which have an address following the final address of wrap boundary are merged. This function is optional

18.6.5 Timing Adjustment

This section describes the HyperBus timing adjustment. HyperBus core has timing adjustment circuit and timing adjustment is controlled by HYPERBUS_MTRn. RCSHI, RCSS, and RCSH in HYPERBUS_MTRn are used for the read timing adjustment, and WCSHI, WCSS, and WCSH in HYPERBUS_MTRn are used for the write timing adjustment.

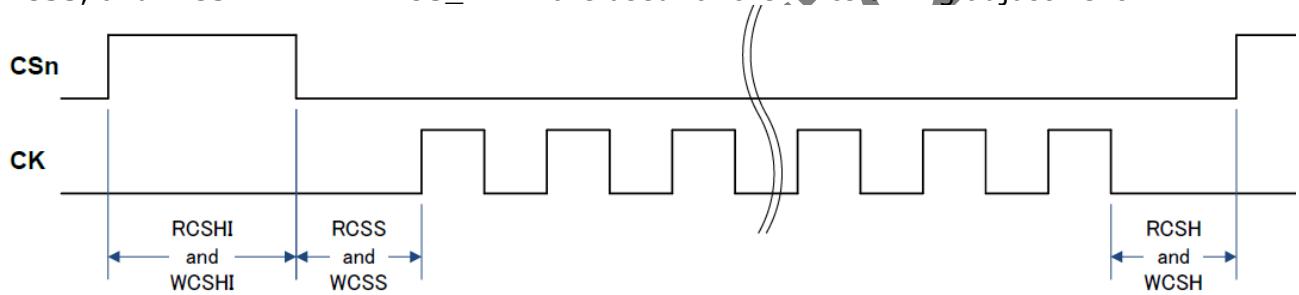


Fig. 18-6 Timing Adjustment

Rockchip

Chapter 19 SPI2APB

19.1 Overview

The SPI2APB module supports the following features:

- External SPI master can access the whole address space in the chip through SPI2APB convertor.
- A transaction states register and a message register can be read by external SPI master by query operation.
- Two 32 bit registers can be written by external SPI master, and one of them can generate an interrupt after been written. This two register can be read by masters such as MCU.

19.2 Block Diagram

The figure below shows SPI2APB block structure.

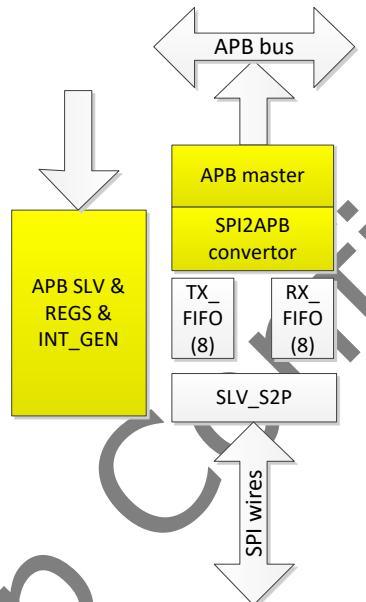


Fig.19-1SPI2APB Block Diagram

19.3 Function Description

19.3.1 Overview

SPI2APB convertor receives data from SPI wires, analysis the command and operates according to the protocol.

19.3.2 Protocol Description

SPI2APB supports four types of operation: Write, Read, Query and Write Message. By default, the command, address and data are all transferred in 32 bits packets in little endian, and the first bit to be transferred is the LSB. The operations begin at the negative edge of CS and end at the positive edge of CS. Between two operation, CS should keep high for a time longer than one SCLK period.

Write

External SPI master should dessert CS firstly. The Write command, address and data packets should be transferred according to the order of the figure. The first data (data0 in the figure) will be written to the address ADDR, the second data will be written to the address ADDR+4, and so on. External SPI master can terminate the write operation by setting the CS to high level.

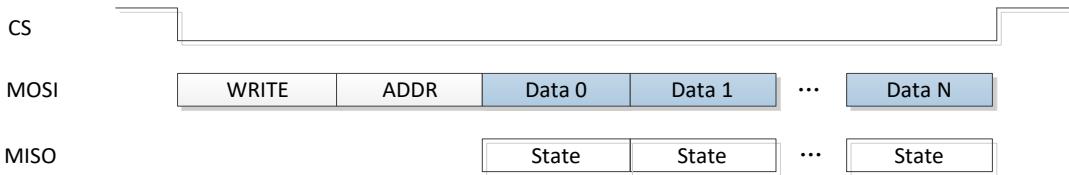


Fig.19-2 Write operation

Read

After receiving READ command and address, the convertor will read N data from the address and push the data into the RD_FIFO (N is a parameter in READ command). Dummy packets can be inserted to ensure there is available data in the RD_FIFO. When SPI master is ready to receive data, it should send a RD_BEGIN command packet. The first packets following RD_BEGIN packets (data0 in the figure) is the data read from address ADDR, the second packets is the data read from address ADDR+ 4, and so on. After N data packets have been transferred, SPI2APB will begin to transfer state information.

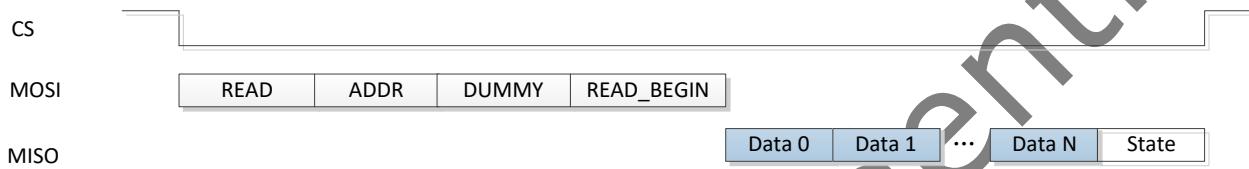


Fig.19-3 Read operation

Query

There are two registers can be queried: State Register and Message Register 2. State register contains transfer state information such as whether the transfer is complete or not, detail is showed in the table. Message Register 2 can be written by DSP. The query result is immediately following the query command without delay.

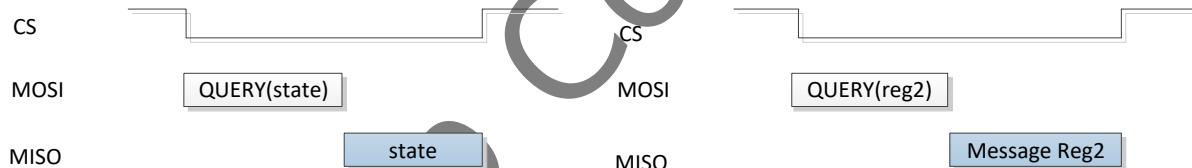


Fig.19-4 Query operation

Bit	Attr	Reset Value	Description
[31:16]	RW	0x1608	Chip ID
[15:11]	RW	0x0	reserved
10	RW	0x0	pre_err Error response is received when APB master is doing pre-read transaction
9	RW	0x0	rd_udflw Read FIFO is underflow
8	RW	0x0	rd_err Data with error response has been transmitted
[7:4]	RW	0x0	reserved
3	RW	0x0	trans_cfl Transaction conflict. Previous transaction is unfinished and new transaction is launched by SPI master

Bit	Attr	Reset Value	Description
2	RW	0x0	trans_unfi Transaction is unfinished
1	RW	0x0	wr_ovfl Write FIFO is overflow
0	RW	0x0	wr_err Error response is received when APB master is writing

Write Message

There are two message registers can be written by external SPI master: Message Register 0 and Message Register 1. These two registers can be read by DSP. After Message Register 1 has been written, an interrupt will be generated.

The control register (CTRL0) can be written by using this command too.



Fig.19-5 Write message operation

Command encoding

All commands are 32bits.

command	format	description
READ	0x0077+(PRE_NUM <<16)	PRE_NUM is pre_read number. For example, 0x00100077 indicates pre_read number is 16. If PRE_NUM is 0, converter will read data when RD_FIFO is not full and read transaction will not be terminated until ss_in_n is pulled high.
READ_BEGIN	0x000000aa	Valid data is the next frame following READ_BEGIN command.
WRITE	0x00000011	Write command, followed by write address.
WRITE MESSAGE (message reg0)	0x00010011	Write message register 0.
WRITE MESSAGE (message reg1)	0x00020011	Write message register 1, and generate an interrupt.
WRITE MESSAGE (ctrl0)	0x00030011	Write ctrl0 register.
QUERY (state)	0x000000ff	Query SPI2APB status.
QUERY (Message register 2)	0x000001ff	Query message in Message register2, contents in which is written by MCU or CPU.

19.4 Register Description

19.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SPI2APB_CTRL0	0x0000	W	0x00000000	Control Register 0
SPI2APB_SR	0x0024	W	0x00000014	SPI2APB status
SPI2APB_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI2APB_RISR	0x0034	W	0x00000000	Raw Interrupt Status
SPI2APB_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI2APB_VERSION	0x0048	W	0x063b0002	Version
SPI2APB_QUICK_REG0	0x0050	W	0x00000000	Quick Write Register0
SPI2APB_QUICK_REG1	0x0054	W	0x00000000	Quick Write Register1
SPI2APB_QUICK_REG2	0x0058	W	0x00000000	Quick Read Register2

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

19.4.2 Detail Register Description

SPI2APB_CTRL0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	TXCP 1'b0: TX clock is not inverted 1'b1: TX clock is inverted
2	RW	0x0	RXCP 1'b0: RX clock is not inverted 1'b1: RX clock is inverted
1	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
0	RW	0x0	FBM 1'b0: first bit is LSB 1'b1: first bit is MSB

SPI2APB_SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x1	RFE 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
3	RO	0x0	RFF 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full

Bit	Attr	Reset Value	Description
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 the ss_in signal is active low. 1'b0: ss_in is not active 1'b1: ss_in is active low

SPI2APB IMR

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	QWIM 1'b0: spi2apb reg1 quick write interrupt is masked 1'b1: spi2apb reg1 quick write interrupt is not masked

SPI2APB RISR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	QWRIS 1'b0: spi2apb reg1 quick write interrupt is not active 1'b1: spi2apb reg1 quick write interrupt is active

SPI2APB ICR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1 C	0x0	CQWI Write 1 to Clear SPI2APB quick write reg1 Interrupt

SPI2APB VERSION

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:0	RO	0x063b0002	VER version

SPI2APB QUICK REG0

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	QWV0 SPI master write to this register, and CPU read it.

SPI2APB QUICK REG1

Address: Operational Base + offset (0x0054)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	QWV1 SPI master write to this register, generate an interrupt .and CPU read it.

SPI2APB QUICK REG2

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	QRV CPU writes a value to this register, and spi master read it.

19.5 Interface Description

SPI2APB's relative IO is default muxed to SPI2APB interface, user don't need to configure the IO mux.

Table 19-1SPI2APB interface description

Module Pin	Direction	Pin Name	IOMUX Setting
spislv0_csn	I	LCD_D4/CIF_D4/UART2_CTSN_M1/SPI1_CS0n_M1/SPI_SLV_CSn/GPIO0_B0_u	GPIO0B_IOMUX_L[2:0]=3'h5
spislv0_clk	I	LCD_D5/CIF_D5/UART2_RTSN_M1/SPI1_CLK_M1/SPI_SLV_CLK/GPIO0_B1_u	GPIO0B_IOMUX_L[6:4]=3'h5
spislv0_mosi	I	LCD_D6/CIF_D6/UART2_RX_M1/SPI1_MOSI_M1/SPI_SLV_MOSI/GPIO0_B2_u	GPIO0B_IOMUX_L[10:8]=3'h5
spislv0_miso	O	LCD_D7/CIF_D7/UART2_TX_M1/SPI1_MISO_M1/SPI_SLV_MISO/GPIO0_B3_u	GPIO0B_IOMUX_L[14:12]=3'h5

19.6 Application Notes

External SPI master should query the transaction state after every write and read operation to ensure the operation is completed correctly.

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 in either master or slave mode.

SPI Controller supports the following features:

- Support Motorola SPI, TI Synchronous Serial Protocol and National Semiconductor Micro wire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 64-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bits 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
- Support timeout mechanism in slave mode
- Support BYPASS slave mode, in which RX and TX logic is driven by SCLK_IN directly instead of spi_clk

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

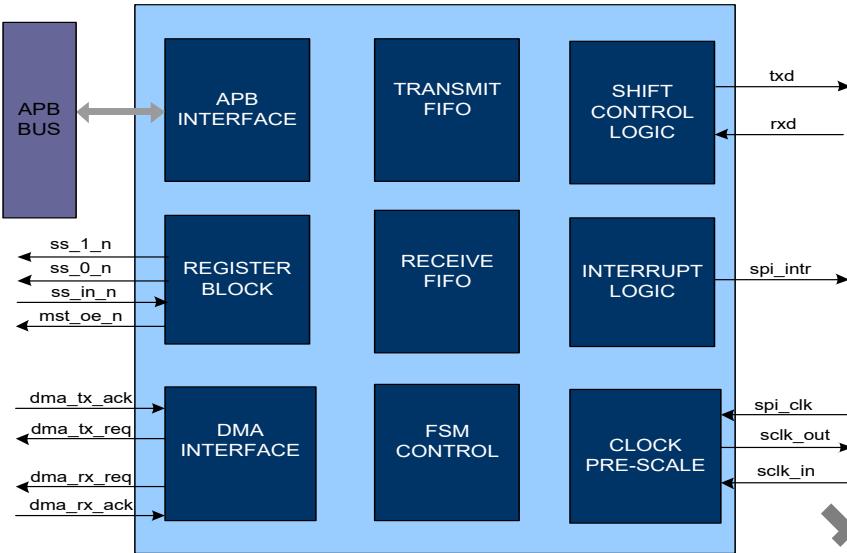


Fig. 20-1 SPI Controller Block Diagram

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 64x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the APB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit FIFO or to the receive FIFO. This logic automatically right-justifies receive data in the receive FIFO buffer.

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

20.3 Function Description

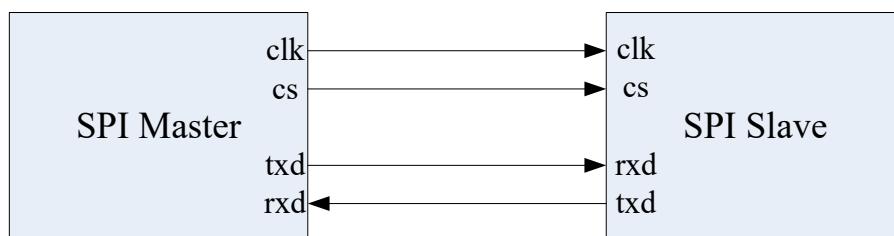


Fig. 20-2 SPI Master and Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram shows how the SPI controller connects with other SPI devices.

Operation Modes

The SPI can be configured in the following two fundamental modes of operation: Master

Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

Transfer Modes

The SPI operates in the following three modes when transferring data on the serial bus.

1). Transmit and Receive

When SPI_CTRLR0 [19:18] == 2'b00, both transmit and receive logic are valid.

2). Transmit Only

When SPI_CTRLR0 [19:18] == 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3). Receive Only

When SPI_CTRLR0 [19:18] == 2'b10, the transmit data are invalid.

Clock Ratios

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as,

When SPI Controller works as master, the $F_{spi_clk} \geq 2 \times (\text{maximum } F_{sclk_out})$

When SPI Controller works as slave, the $F_{spi_clk} \geq 6 \times (\text{maximum } F_{sclk_in})$

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.

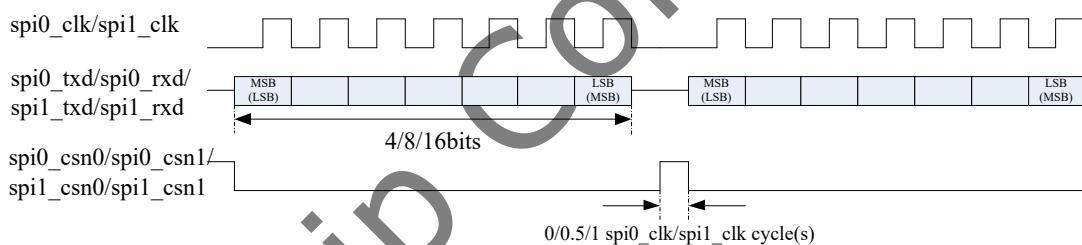


Fig. 20-3 SPI Format (SCPH=0 SCPOL=0)

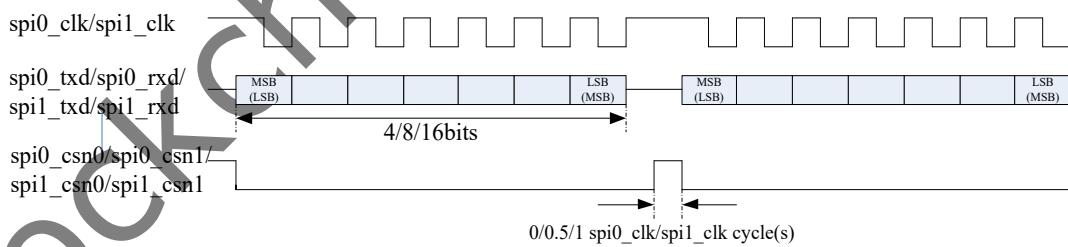


Fig. 20-4 SPI Format (SCPH=0 SCPOL=1)

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.

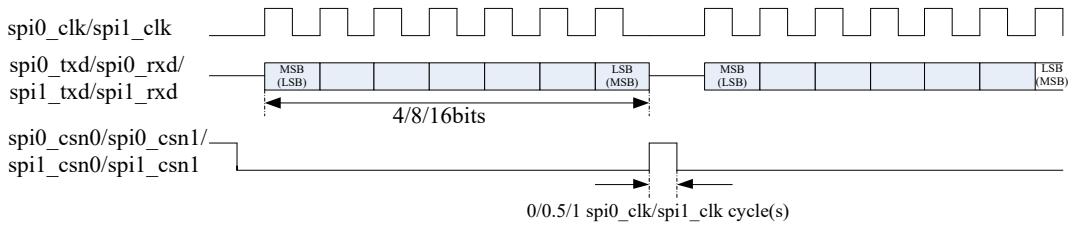


Fig. 20-5 SPI Format (SCPH=1 SCPOL=0)

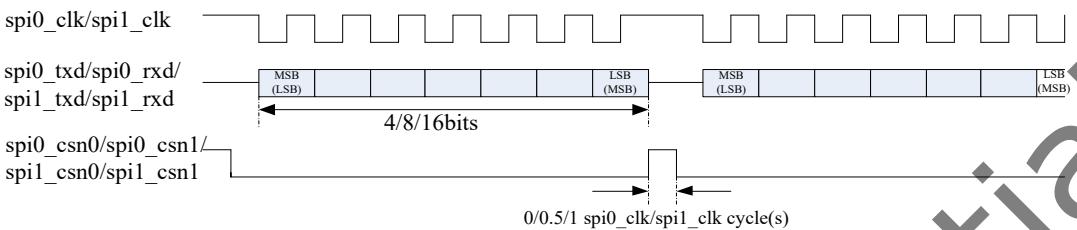


Fig. 20-6 SPI Format (SCPH=1 SCPOL=1)

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	0x00000000	SPI Status
SPI_IPR	0x0028	W	0x00000000	Interrupt Polarity
SPI_IMR	0x002c	W	0x00000000	Interrupt Mask
SPI_ISR	0x0030	W	0x00000000	Interrupt Status
SPI_RISR	0x0034	W	0x00000001	Raw Interrupt Status
SPI_ICR	0x0038	W	0x00000000	Interrupt Clear
SPI_DMACR	0x003c	W	0x00000000	DMA Control
SPI_DMATDLR	0x0040	W	0x00000000	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x00000000	DMA Receive Data Level
SPI_TIMEOUT	0x004c	W	0x00000000	Timeout control register
SPI_BYPASS	0x0050	W	0x00000000	BYPASS control register
SPI_TXDR	0x0400	W	0x00000000	Transmit FIFO Data
SPI_RXDR	0x0800	W	0x00000000	Receive FIFO Data

Notes: **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
31:26	RO	0x0	reserved
25	RW	0x0	LBK Loop back 1'b0: Normal mode 1'b1: Loop back mode, rxd is connected to txd
24:23	RW	0x0	SOI SS_N output inverted 1'b0: Corresponding bit of ss_in is not inverted 1'b1: Corresponding bit of ss_in is inverted
22	RO	0x0	reserved
21	RW	0x0	MTM Valid when frame format is set to National Semiconductors Microwire. 1'b0: Non-sequential transfer 1'b1: Sequential transfer
20	RW	0x0	OPM 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
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

Bit	Attr	Reset Value	Description
11	RW	0x0	<p>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</p>
10	RW	0x0	<p>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.</p>
9:8	RW	0x0	<p>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</p>
7	RW	0x0	<p>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.</p>
6	RW	0x0	<p>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.</p>
5:2	RW	0x0	<p>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</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x2	DFS Selects the data frame length. 2'b00: 4bit data 2'b01: 8bit data 2'b10: 16bit data 2'b11: Reserved

SPI CTRLR1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	NDM When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 4GB of data in a continuous transfer.

SPI ENR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:1	RO	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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>BAUDR SPI Clock Divider.</p> <p>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:</p> $Fsclk_{out} = Fspi_{clk} / SCKDV$ <p>Where SCKDV is any even value between 2 and 65534.</p> <p>For example: for Fspi_clk = 3.6864MHz and SCKDV = 2 $Fsclk_{out} = 3.6864/2 = 1.8432MHz$</p>

SPI TXFTLR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	<p>TXFTLR</p> <p>When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.</p>

SPI RXFTLR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	<p>RXFTLR</p> <p>When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.</p>

SPI TXFLR

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RO	0x00	<p>TXFLR</p> <p>Contains the number of valid data entries in the transmit FIFO</p>

SPI RXFLR

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RO	0x00	<p>RXFLR</p> <p>Contains the number of valid data entries in the receive FIFO</p>

SPI SR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x1	SSI 1'b0: SS_IN_N is low 1'b1: SS_IN_N is high
5	RW	0x0	STB 1'b0: Slave tx not busy 1'b1: Slave tx busy
4	RW	0x0	RFF 1'b0: Receive FIFO is not full 1'b1: Receive FIFO is full
3	RW	0x1	RFE 1'b0: Receive FIFO is not empty 1'b1: Receive FIFO is empty
2	RW	0x1	TFE 1'b0: Transmit FIFO is not empty 1'b1: Transmit FIFO is empty
1	RW	0x0	TFF 1'b0: Transmit FIFO is not full 1'b1: Transmit FIFO is full
0	RW	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
31:8	RO	0x0	reserved
7	RW	0x0	TXFIM 1'b0: Tx finish interrupt is masked. 1'b1: Tx finish interrupt is not masked.

Bit	Attr	Reset Value	Description
6	RW	0x0	SSPIM 1'b0: SS_IN_N posedge interrupt is masked. 1'b1: SS_IN_N posedge interrupt is not masked.
5	RW	0x0	TOIM 1'b0: SPI timeout interrupt is masked. 1'b1: SPI timeout interrupt is not masked.
4	RW	0x0	RFFIM 1'b0: spi_rxf_intr interrupt is masked. 1'b1: spi_rxf_intr interrupt is not masked.
3	RW	0x0	RFOIM 1'b0: spi_rxo_intr interrupt is masked. 1'b1: spi_rxo_intr interrupt is not masked.
2	RW	0x0	RFUIM 1'b0: spi_rxu_intr interrupt is masked. 1'b1: spi_rxu_intr interrupt is not masked.
1	RW	0x0	TFOIM 1'b0: spi_txo_intr interrupt is masked. 1'b1: spi_txo_intr interrupt is not masked.
0	RW	0x0	TFEIM 1'b0: spi_txe_intr interrupt is masked. 1'b1: spi_txe_intr interrupt is not masked.

SPI ISR

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	TXFIS 1'b0: TX finish interrupt is not active after masking. 1'b1: TX finish interrupt is active after masking.
6	RW	0x0	SSPIS 1'b0: ss_in_nposedege interrupt is not active after masking. 1'b1: ss_in_nposedege interrupt is active after masking.
5	RW	0x0	TOIS 1'b0: spi timeout interrupt is not active after masking. 1'b1: spi timeout interrupt is active after masking.
4	RO	0x0	RFFIS 1'b0: spi_rxf_intr interrupt is not active after masking. 1'b1: spi_rxf_intr interrupt is full after masking.
3	RO	0x0	RFOIS 1'b0: spi_rxo_intr interrupt is not active after masking. 1'b1: spi_rxo_intr interrupt is active after masking.
2	RO	0x0	RFUIS 1'b0: spi_rxu_intr interrupt is not active after masking. 1'b1: spi_rxu_intr interrupt is active after masking.

Bit	Attr	Reset Value	Description
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
31:8	RO	0x0	reserved
7	RW	0x0	TXFRIS 1'b0: TX finish interrupt is not active prior to masking. 1'b1: TX finish interrupt is active prior to masking.
6	RW	0x0	SSPRIS 1'b0: ss_in_nposedege interrupt is not active prior to masking. 1'b1: ss_in_nposedege interrupt is active prior to masking.
5	RW	0x0	TORIS 1'b0: spi timeout interrupt is not active prior to masking. 1'b1: spi timeout interrupt is active prior to masking.
4	RO	0x0	RFFRIS 1'b0: spi_rxf_intr interrupt is not active prior to masking. 1'b1: spi_rxf_intr interrupt is full prior to masking.
3	RO	0x0	RFORIS 1'b0: spi_rxo_intr interrupt is not active prior to masking. 1'b1: spi_rxo_intr interrupt is active prior to masking.
2	RO	0x0	RFURIS 1'b0: spi_rxu_intr interrupt is not active prior to masking. 1'b1: spi_rxu_intr interrupt is active prior to masking.
1	RO	0x0	TFORIS 1'b0: spi_txo_intr interrupt is not active prior to masking. 1'b1: spi_txo_intr interrupt is active prior to masking.
0	RO	0x1	TFERIS 1'b0: spi_txe_intr interrupt is not active prior to masking. 1'b1: spi_txe_intr interrupt is active prior to masking.

SPI ICR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	W1C	0x0	CTXFI Write 1 to Clear tx finish Interrupt

Bit	Attr	Reset Value	Description
5	W1C	0x0	CSSPI Write 1 to Clear ss_in_nposdege Interrupt
4	W1C	0x0	CTOI Write 1 to Clear Timeout Interrupt
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
31:6	RO	0x0	reserved
5:0	RW	0x00	TDL This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI DMARDLR

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	RDL This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI TIMEOUT

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	TOE Timeoutenable 1'b0: Timeout counter is inactive 1'b1: Timeout counter is active
15:0	RW	0x0000	TOV Timeout threshold value If SCLK_IN keep inactive for a time, timeout interrupt will be triggered. The timeout threshold time is TOV*pclk_perid*16.

SPI BYPASS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RW	0x0	TXCP TX clock polarity 1'b0: TX logic use raw SCLK 1'b1: TX logic use inverted SCLK
3	RW	0x0	RXCP RX clock polarity 1'b0: RX logic use raw SCLK 1'b1: RX logic use inverted SCLK
2	RW	0x0	END Endian mode 1'b0: Work in little endian mode 1'b1: Work in big endian mode
1	RW	0x0	FBM First bit mode 1'b0: First bit is LSB 1'b1: First bit is MSB

Bit	Attr	Reset Value	Description
0	RW	0x0	BYEN Bypass enable 1'b0: Normal mode 1'b1: Bypass mode, SPI serial/parallel convert logic is drive by SCLK

SPI TXDR

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RO	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

There are two SPIs in device, SPI0 can be muxed to two IO groups, SPI1 can be muxed to three IO groups. SPI0's IO group can be set by configuring GRF_SOC_CON1[1], SPI1's IO group can be set by configuring GRF_SOC_CON1[3:2].

Table 20-1 SPI interface description

Module Pin	Direction	Pin Name	IOMUX Setting
spi0_m0_csn	I/O	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GPIO0B_IOMUX_H[3:0]=4'h3
spi0_m0_clk	I/O	PWM1_M1/UART0_RTSN_M0/SPI0_CLK_M0/I2C0_SCL_M0/TKEY1_M0/PWM_AUDIO_R_M0/I2C2_SCL_M1/GPIO0_B5_d	GPIO0B_IOMUX_H[6:4]=3'h3
spi0_m0_miso	I/O	PWM3_M1/UART0_TX_M0/SPI0_MISO_M0/I2C1_SCL_M0/TKEY3_M0/TKEY_DRIVE_M2/GPIO0_B7_d	GPIO0B_IOMUX_H[14:12]=3'h3
spi0_m0_mosi	I/O	PWM2_M1/UART0_RX_M0/SPI0_MOSI_M0/I2C1_SDA_M0/TKEY2_M0/GPIO0_B6_d	GPIO0B_IOMUX_H[10:8]=3'h3
spi0_csn1/spi1_cs_n1	I/O	PWM7_M1/SPI0_CS1n/SPI1_CS1n/GPIO1_D0_u	GPIO0D_IOMUX_H[3:0]=3'h2/ GPIO0D_IOMUX_H[3:0]=3'h3
spi0_m1_csn	I/O	SRADC0/PWM0_M0/UART1_CTSN_M0/SPI0_CS0n_M1/I2S_SDO1_M0/SDMMC_CLKOUT/TKEY9/GPIO0_C0_u	GPIO0C_IOMUX_L[3:0]=3'h4
spi0_m1_clk	I/O	SRADC1/PWM1_M0/UART1_RTSN_M0/SPI0_CLK_M1/I2C1_SDA_M1/I2S_SCLK_RX_M0/S	GPIO0C_IOMUX_L[7:4]=4'h4

Module Pin	Direction	Pin Name	IOMUX Setting
		DMMC_CMD/TKEY10/GPIO0_C1_u	
spi0_m1_mosi	I/O	SRADC2/PWM2_M0/UART1_RX_M0/SPI0_MO SI_M1/I2C1_SCL_M1/I2S_LRCK_RX_M0/SD MMC_D0/TKEY11/GPIO0_C2_u	GPIO0C_IOMUX_L[11:8]=4'h4
spi1_m1_miso	I/O	SRADC3/PWM3_M0/UART1_TX_M0/SPI0_MI SO_M1/PDM_CLK_S_M1/I2S_MCLK_M0/TKE Y4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GPIO0C_IOMUX_L[15:12]=4'h4
SPI1 Interface			
spi1_m0_csn0	I/O	SRADC4/PWM4_M0/UART0_CTSN_M1/SPI1_CS0n_M0/PDM_CLK_M1/I2S_SCLK_TX_M0/T KEY5_M0/GPIO0_C4_u	GPIO0C_IOMUX_H[2:0]=4'h4
spi1_m0_clk	I/O	SRADC5/PWM5_M0/UART0_RTSN_M1/SPI1_CLK_M0/PDM_SDI_M1/I2S_LRCK_TX_M0/T KEY6_M0/GPIO0_C5_u	GPIO0C_IOMUX_H[6:4]=4'h4
spi1_m0_miso	I/O	SRADC7/PWM7_M0/UART0_TX_M1/SPI1_MI SO_M0/I2C0_SCL_M1/I2S_SDI_M0/TKEY8/P WM_AUDIO_R_M1/PMIC_INT_M1/GPIO0_C7_d	GPIO0C_IOMUX_H[15:12]=4'h4
spi1_m0_mosi	I/O	SRADC6/PWM6_M0/UART0_RX_M1/SPI1_MO SI_M0/I2C0_SDA_M1/I2S_SDO_M0/TKEY7/P WM_AUDIO_L_M1/GPIO0_C6_u	GPIO0C_IOMUX_H[11:8]=4'h4
spi1_m1_csn	I/O	LCD_D4/CIF_D4/UART2_CTSN_M1/SPI1_CS On_M1/SPI_SLV_CS_n/GPIO0_B0_u	GPIO0B_IOMUX_L[2:0]=3'h4
spi1_m1_clk	I/O	LCD_D5/CIF_D5/UART2_RTSN_M1/SPI1_CLK _M1/SPI_SLV_CLK/GPIO0_B1_u	GPIO0B_IOMUX_L[6:4]=3'h4
spi1_m1_mosi	I/O	LCD_D6/CIF_D6/UART2_RX_M1/SPI1_MOSI _M1/SPI_SLV_MOSI/GPIO0_B2_u	GPIO0B_IOMUX_L[10:8]=3'h4
spi1_m1_miso	I/O	LCD_D7/CIF_D7/UART2_TX_M1/SPI1_MISO_ M1/SPI_SLV_MISO/GPIO0_B3_u	GPIO0B_IOMUX_L[14:12]=3'h4
spi1_m2_csn	I/O	CODEC_SYNC_M0/PWM5_M1/TKEY1_M1/UA RT1_TX_M2/SPI1_CS0n_M2/I2C0_SCL_M3/P WM_AUDIO_R_M2/GPIO0_D1_u	GPIO0D_IOMUX_L[6:4]=3'h5
spi1_m2_clk	I/O	CODEC_ADC_D_M0/PWM6_M1/TKEY2_M1/U ART2_CTSN_M0/SPI1_CLK_M2/I2S_MCLK_M 1/GPIO0_D2_u	GPIO0D_IOMUX_L[10:8]=3'h5
spi1_m2_mosi	I/O	CODEC_DAC_DL_M0/PWM8/TKEY3_M1/UART 2_RTSN_M0/SPI1_MOSI_M2/I2S_SCLK_TX_ M1/GPIO0_D3_u	GPIO0D_IOMUX_L[14:12]=3'h5
spi1_m2_miso	I/O	PMIC_INT_M0/PWM9/TKEY4_M1/UART2_RX_ M0/SPI1_MISO_M2/I2S_LRCK_TX_M1/GPIO 0_D4_u	GPIO0D_IOMUX_H[2:0]=3'h5

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 $F_{spi_clk} \geq 2 \times (\text{maximum } F_{sclk_out})$

When SPI Controller works as slave, the $F_{spi_clk} \geq 6 \times (\text{maximum } F_{sclk_in})$

Master Transfer Flow

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the `sclk_out` line. When the SPI is disabled (`SPI_ENR = 0`), no serial transfers can occur and `sclk_out` is held in "inactive" state, as defined by the serial protocol under which it operates.

Slave Transfer Flow

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its `txd` data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (`sclk_in`), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

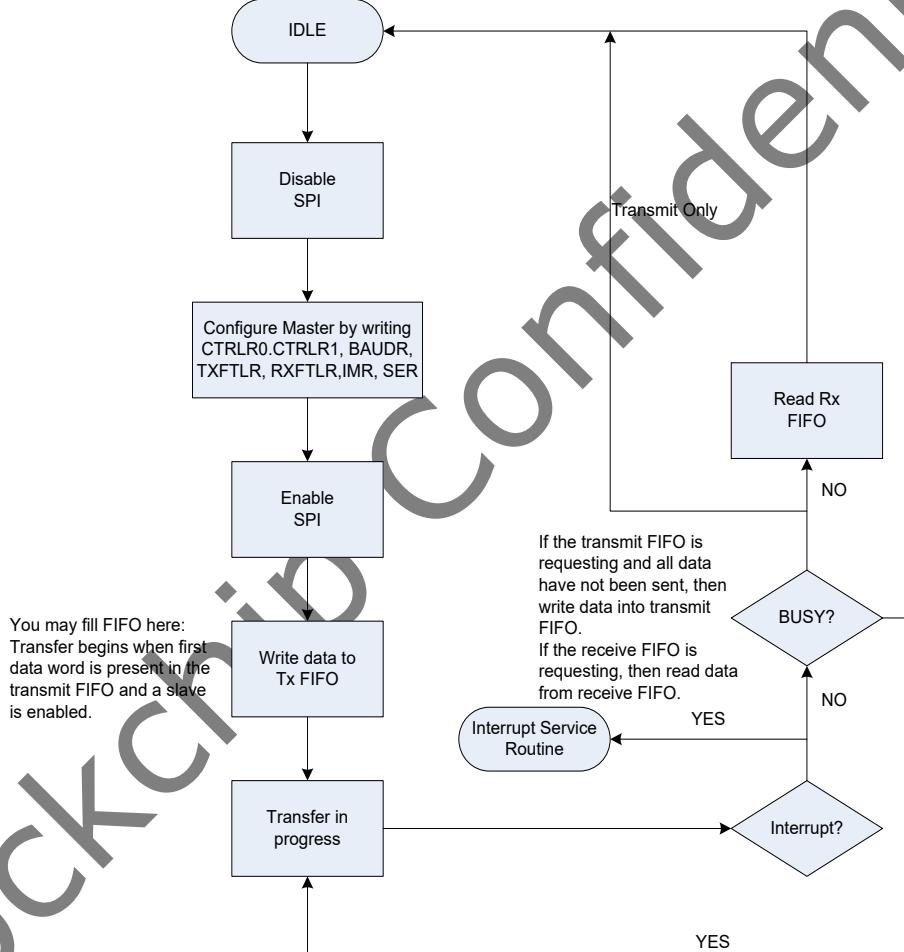


Fig. 20-7 SPI Master transfer flow diagram

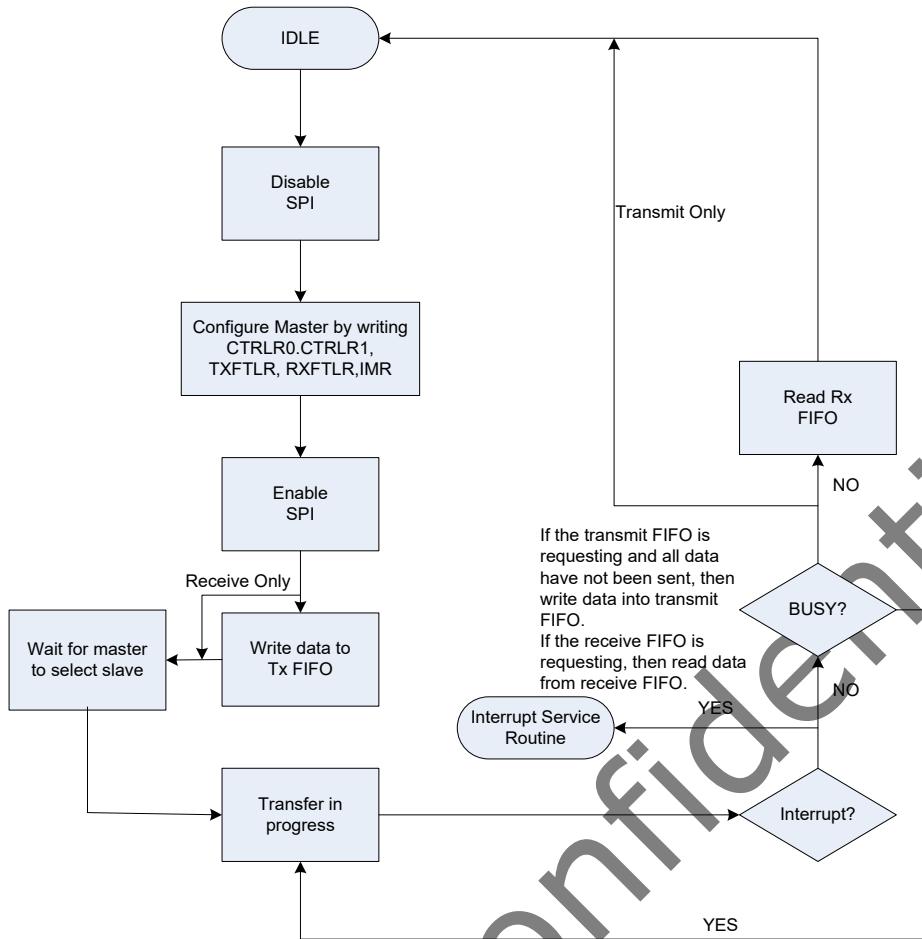


Fig. 20-8 SPI Slave transfer flow diagram

Rockchip Confidential

Chapter 21 UART

21.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

UART Controller supports the following features:

- Support 3 independent UART controllers: UART0, UART1, UART2
- UART0/UART1/UART2 all contain two 64Bytes FIFOs for data receive and transmit
- UART0/UART2 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

21.2 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

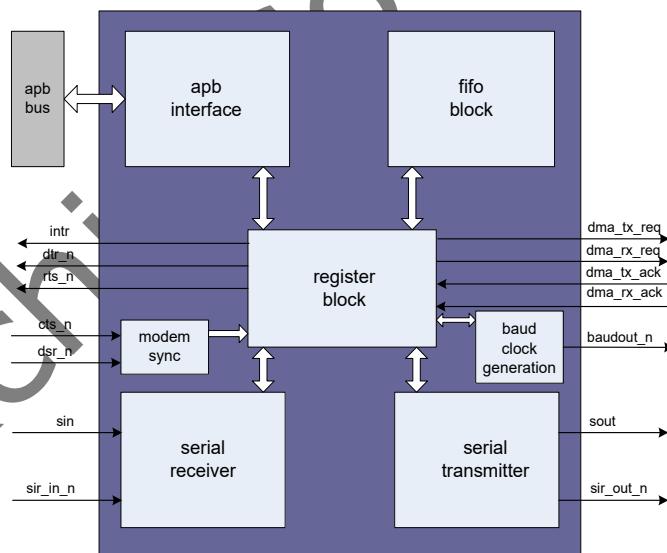


Fig. 21-1 UART Architecture

APB INTERFACE

The host processor accesses data, control, and status information on the UART through the APB interface. The UART supports APB data bus width of 8, 16, and 32 bits.

Register block

Be responsible for the main UART functionality including control, status and interrupt generation.

Modem Synchronization block

Synchronizes the modem input signal.

FIFO block

Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).

Baud Clock Generator

Generates the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. 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.

21.3 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.

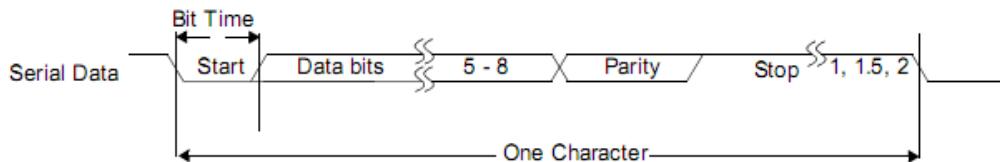


Fig. 21-2 UART Serial protocol

IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

Transmitting a single infrared pulse signals a logic zero, while a logic one is represented by not sending a pulse. The width of each pulse is 3/16ths of a normal serial bit time. Data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled.

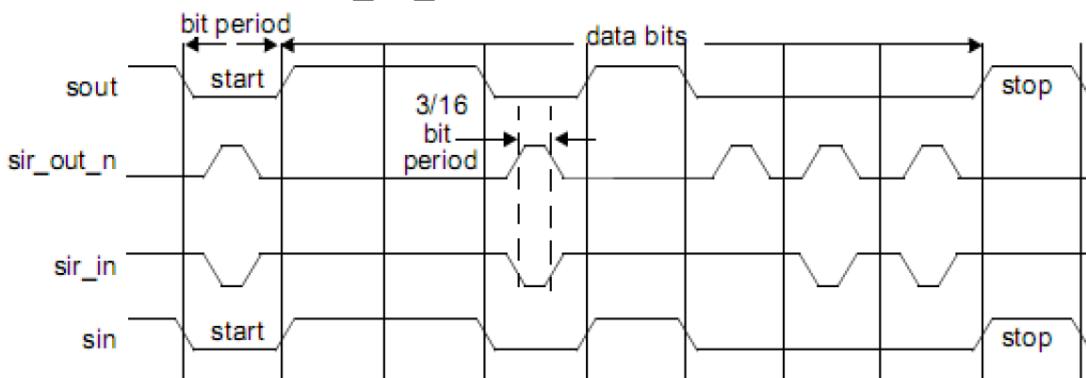


Fig. 21-3 IrDA 1.0

Baud Clock

The baud rate is controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid-point for sampling is not difficult, that is every 16 baud clocks after the mid-point sample of the start bit.

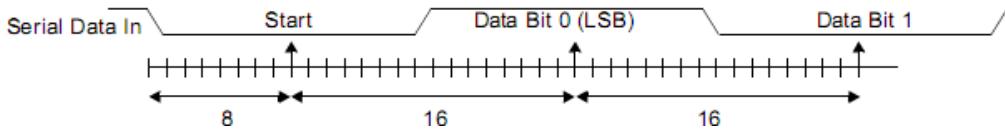


Fig. 21-4 UART baud rate

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 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.

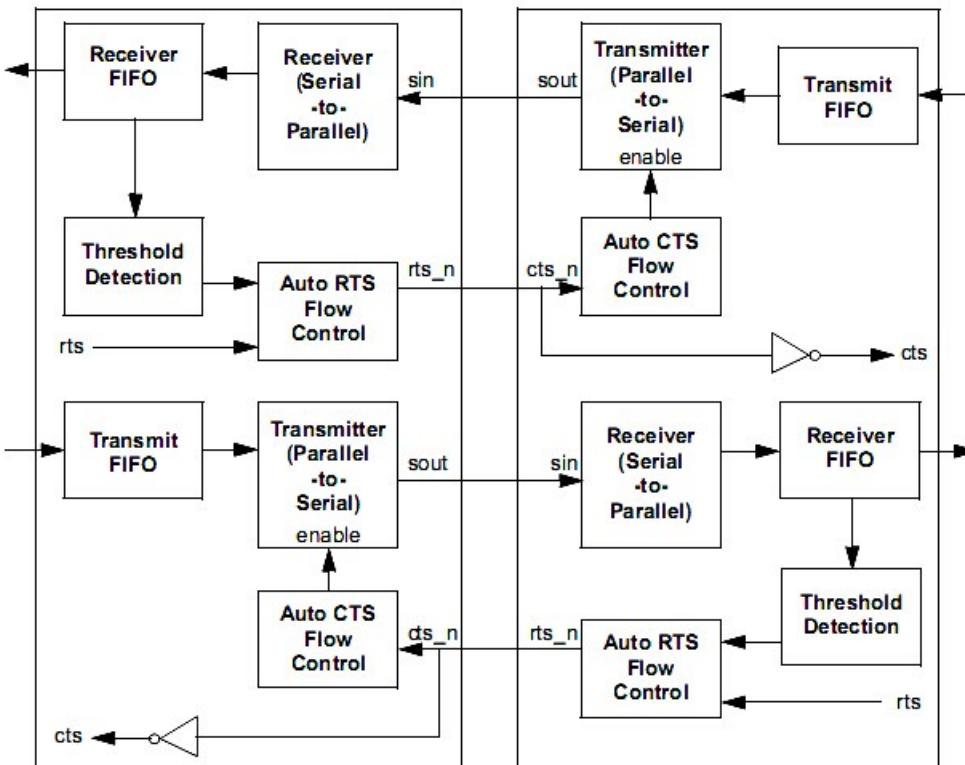


Fig. 21-5 UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0] bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

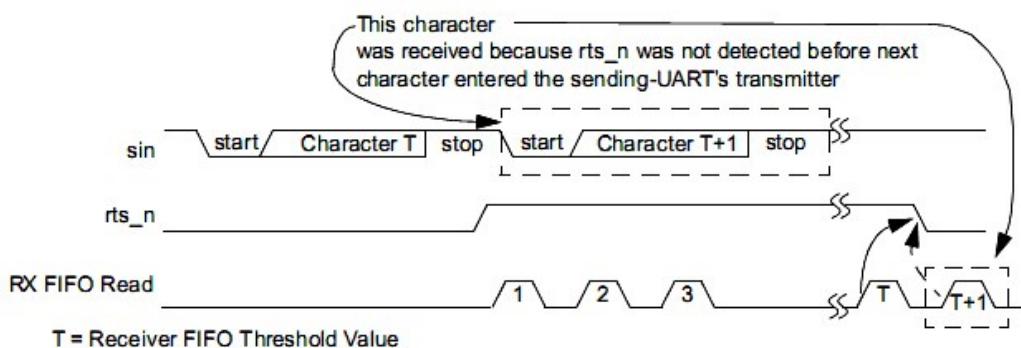


Fig. 21-6 UART AUTO RTS TIMING

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)

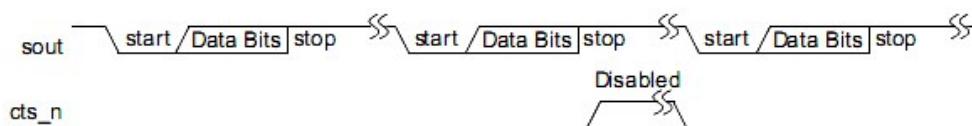


Fig. 21-7 UART AUTO CTS TIMING

21.4 Register Description

This section describes the control/status registers of the design. There are 3 UARTs, and each one has its own base address.

21.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x00000000	Receive Buffer Register
UART_THR	0x0000	W	0x00000000	Transmit Holding Register
UART_DLL	0x0000	W	0x00000000	Divisor Latch (Low)
UART_DLH	0x0004	W	0x00000000	Divisor Latch (High)
UART_IER	0x0004	W	0x00000000	Interrupt Enable Register
UART_IIR	0x0008	W	0x00000000	Interrupt Identification Register
UART_FCR	0x0008	W	0x00000000	FIFO Control Register
UART_LCR	0x000c	W	0x00000000	Line Control Register
UART_MCR	0x0010	W	0x00000000	Modem Control Register
UART_LSR	0x0014	W	0x00000000	Line Status Register
UART_MSR	0x0018	W	0x00000000	Modem Status Register
UART_SCR	0x001c	W	0x00000000	Scratchpad Register
UART_SRBR	0x0030-0x006c	W	0x00000000	Shadow Receive Buffer Register
UART_STHR	0x0030-0x006c	W	0x00000000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x00000000	FIFO Access Register
UART_TFR	0x0074	W	0x00000000	Transmit FIFO Read
UART_RFW	0x0078	W	0x00000000	Receive FIFO Write
UART_USR	0x007c	W	0x00000000	UART Status Register
UART_TFL	0x0080	W	0x00000000	Transmit FIFO Level
UART_RFL	0x0084	W	0x00000000	Receive FIFO Level
UART_SRR	0x0088	W	0x00000000	Software Reset Register
UART_SRTS	0x008c	W	0x00000000	Shadow Request to Send
UART_SBCR	0x0090	W	0x00000000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x00000000	Shadow DMA Mode
UART_SFE	0x0098	W	0x00000000	Shadow FIFO Enable
UART_SRT	0x009c	W	0x00000000	Shadow RCVR Trigger
UART_STET	0x00a0	W	0x00000000	Shadow TX Empty Trigger
UART_HTX	0x00a4	W	0x00000000	Halt TX
UART_DMASA	0x00a8	W	0x00000000	DMA Software Acknowledge
UART_CPR	0x00f4	W	0x00000000	Component Parameter Register
UART_UCV	0x00f8	W	0x0330372a	UART Component Version
UART_CTR	0x00fc	W	0x44570110	Component Type Register

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

21.4.2 Detail Register Description

UART_RBR

Address: Operational Base + offset (0x0000)

Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>data_input</p> <p>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.</p>

UART_THR

Address: Operational Base + offset (0x0000)

Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>data_output</p> <p>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.</p> <p>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.</p> <p>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.</p>

UART_DLL

Address: Operational Base + offset (0x0000)

Divisor Latch (Low)

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

Bit	Attr	Reset Value	Description
7:0	RW	0x00	<p>baud_rate_divisor_L</p> <p>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.</p>

UART_DLH

Address: Operational Base + offset (0x0004)

Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	<p>baud_rate_divisor_H</p> <p>Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.</p>

UART_IER

Address: Operational Base + offset (0x0004)

Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	<p>prog_thre_int_en</p> <p>Programmable THRE Interrupt Mode Enable</p> <p>This is used to enable/disable the generation of THRE Interrupt.</p> <p>1'b0: Disabled 1'b1: Enabled</p>
6:4	RO	0x0	reserved
3	RW	0x0	<p>modem_status_int_en</p> <p>Enable Modem Status Interrupt</p> <p>This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt.</p> <p>1'b0: Disabled 1'b1: Enabled</p>
2	RW	0x0	<p>receive_line_status_int_en</p> <p>Enable Receiver Line Status Interrupt</p> <p>This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt.</p> <p>1'b0: Disabled 1'b1: Enabled</p>

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)

Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	fifos_en 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	0x0	int_id Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types: 4'b0000: Modem status 4'b0001: No interrupt pending 4'b0010: THR empty 4'b0100: Received data available 4'b0110: Receiver line status 4'b0111: Busy detect 4'b1100: Character timeout

UART_FCR

Address: Operational Base + offset (0x0008)

FIFO Control Register

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

Bit	Attr	Reset Value	Description
7:6	WO	0x0	<p>rcvr_trigger</p> <p>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:</p> <ul style="list-style-type: none"> 2'b00: 1 character in the FIFO 2'b01: FIFO 1/4 full 2'b10: FIFO 1/2 full 2'b11: FIFO 2 less than full
5:4	WO	0x0	<p>tx_empty_trigger</p> <p>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:</p> <ul style="list-style-type: none"> 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	<p>dma_mode</p> <p>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.</p> <ul style="list-style-type: none"> 1'b0: Mode 0 1'b1: Mode 11100 = character timeout
2	WO	0x0	<p>xmit_fifo_reset</p> <p>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.</p>
1	WO	0x0	<p>rcvr_fifo_reset</p> <p>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.</p>
0	WO	0x0	<p>fifo_en</p> <p>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.</p>

UART_LCR

Address: Operational Base + offset (0x000c)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	<p>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.</p>
6	RW	0x0	<p>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.</p>
5	RO	0x0	reserved
4	RW	0x0	<p>even_parity_sel 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.</p>
3	RW	0x0	<p>parity_en 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</p>
2	RW	0x0	<p>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, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 1'b0: 1 stop bit 1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p>

Bit	Attr	Reset Value	Description
1:0	RW	0x0	<p>data_length_sel 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:</p> <p>2'b00: 5 bits 2'b01: 6 bits 2'b10: 7 bits 2'b11: 8 bits</p>

UART_MCR

Address: Operational Base + offset (0x0010)

Modem Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	<p>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</p>
5	RW	0x0	<p>auto_flow_ctrl_en Auto Flow Control Enable. 1'b0: Auto Flow Control Mode disabled 1'b1: Auto Flow Control Mode enabled</p>
4	RW	0x0	<p>loopback LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.</p>
3	RW	0x0	<p>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)</p>
2	RW	0x0	<p>out1 This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 1'b0: out2_n de-asserted (logic 1) 1'b1: out2_n asserted (logic 0)</p>
1	RW	0x0	<p>req_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.</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>data_terminal_ready</p> <p>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:</p> <p>1'b0: dtr_n de-asserted (logic 1)</p> <p>1'b1: dtr_n asserted (logic 0)</p>

UART_LSR

Address: Operational Base + offset (0x0014)

Line Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	<p>receiver_fifo_error</p> <p>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.</p> <p>1'b0: No error in RX FIFO</p> <p>1'b1: Error in RX FIFO</p>
6	RO	0x0	<p>trans_empty</p> <p>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.</p>
5	RO	0x0	<p>trans_hold_reg_empty</p> <p>Transmit Holding Register Empty bit.</p> <p>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.</p> <p>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.</p>
4	RO	0x0	<p>break_int</p> <p>This is used to indicate the detection of a break sequence on the serial input data.</p>
3	RO	0x0	<p>framing_error</p> <p>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.</p>

Bit	Attr	Reset Value	Description
2	RO	0x0	parity_error 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 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 This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 1'b0: No data ready 1'b1: Data ready

UART_MSR

Address: Operational Base + offset (0x0018)

Modem Status Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RO	0x0	data_carrier_detect This is used to indicate the current state of the modem control line dcd_n.
6	RO	0x0	ring_indicator This is used to indicate the current state of the modem control line ri_n.
5	RO	0x0	data_set_ready This is used to indicate the current state of the modem control line dsr_n.
4	RO	0x0	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 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 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 This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR

Address: Operational Base + offset (0x001c)
 Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	temp_store_space This register is for programmers to use as a temporary storage space.

UART_SRBR

Address: Operational Base + offset (0x0030)
 Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_rbr This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs.

UART_STHR

Address: Operational Base + offset (0x006c)
 Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	shadow_thr This is a shadow register for the THR.

UART_FAR

Address: Operational Base + offset (0x0070)
 FIFO Access Register

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

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>fifo_access_test_en</p> <p>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.</p> <p>1'b0: FIFO access mode disabled 1'b1: FIFO access mode enabled</p>

UART_TFR

Address: Operational Base + offset (0x0074)

Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	<p>trans_fifo_read</p> <p>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.</p>

UART_RFW

Address: Operational Base + offset (0x0078)

Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9	WO	0x0	<p>receive_fifo_framing_error</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).</p>
8	WO	0x0	<p>receive_fifo_parity_error</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).</p>
7:0	WO	0x00	<p>receive_fifo_write</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).</p> <p>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.</p> <p>When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.</p>

UART_USR

Address: Operational Base + offset (0x007c)

UART Status Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4	RO	0x0	<p>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.</p>
3	RO	0x0	<p>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.</p>
2	RO	0x0	<p>transn_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</p>
1	RO	0x0	<p>trans_fifo_not_full This is used to indicate that the transmit FIFO is 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.</p>
0	RO	0x0	<p>uart_busy UART Busy. This 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)</p>

UART_TFL

Address: Operational Base + offset (0x0080)

Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	<p>trans_fifo_level This indicates the number of data entries in the transmit FIFO.</p>

UART_RFL

Address: Operational Base + offset (0x0084)

Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RO	0x00	<p>receive_fifo_level This indicates the number of data entries in the receive FIFO.</p>

UART_SRR

Address: Operational Base + offset (0x0088)

Software Reset Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	xmit_fifo_reset This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	WO	0x0	rcvr_fifo_reset This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	WO	0x0	uart_reset This asynchronously resets the UART and synchronously removes the reset assertion. For a two-clock implementation both pclk and sclk domains are reset.

UART_SRTS

Address: Operational Base + offset (0x008c)

Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_req_to_send This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR

Address: Operational Base + offset (0x0090)

Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_break_ctrl This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM

Address: Operational Base + offset (0x0094)

Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_dma_mode This is a shadow register for the DMA mode bit (FCR[3]).

UART_SFE

Address: Operational Base + offset (0x0098)

Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_fifo_en Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT

Address: Operational Base + offset (0x009c)

Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_rcvr_trigger This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET

Address: Operational Base + offset (0x00a0)

Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	shadow_tx_empty_trigger This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX

Address: Operational Base + offset (0x00a4)

Halt TX

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	halt_tx_en This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 1'b0: Halt TX disabled 1'b1: Halt TX enabled

UART_DMASA

Address: Operational Base + offset (0x00a8)

DMA Software Acknowledge

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	dma_software_ack This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_CPR

Address: Operational Base + offset (0x00f4)

Component Parameter Register

UART_CPR is UART0's own unique register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x00	FIFO_MODE 8'h0: 0 8'h0: 16 8'h0: 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

Bit	Attr	Reset Value	Description
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)

UART Component Version

Bit	Attr	Reset Value	Description
31:0	RO	0x0330372a	ver ASCII value for each number in the version

UART_CTR

Address: Operational Base + offset (0x00fc)

Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44570110	peripheral_id This register contains the peripherals identification code.

21.5 Interface Description

The I/O interface of UART0 can be chosen by setting GRF_SOC_CON16[0](uart0sel) bit.
 The I/O interface of UART1 can be chosen by setting GRF_SOC_CON16[2:1](uart1sel) bits.
 The I/O interface of UART2 can be chosen by setting GRF_SOC_CON16[3](uart2sel) bit.

Table 21-1 UART Interface Description

Module Pin	Di r	Pad name	IOMUX
UART0 Interface			
uart0_m0_ctsn	I	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GRF_GPIO0B_IOMUX_H[3:0]=4'h2
uart0_m0_rtsn	O	PWM1_M1/UART0_RTSN_M0/SPI0CLK_M0/I2C0_SCL_M0/TKEY1_M0/PWM_AUDIO_R_M0/I2C2_SCL_M1/GPIO0_B5_d	GRF_GPIO0B_IOMUX_H[6:4]=3'h2
uart0_m0_rx	I	PWM2_M1/UART0_RX_M0/SPI0_MOSI_M0/I2C1_SDA_M0/TKEY2_M0/GPIO0_B6_d	GRF_GPIO0B_IOMUX_H[10:8]=3'h2
uart0_m0_tx	O	PWM3_M1/UART0_TX_M0/SPI0_MISO_M0/I2C1_SCL_M0/TKEY3_M0/TKEY_DRIVE_M2/GPIO0_B7_d	GRF_GPIO0B_IOMUX_H[14:12]=3'h2
uart0_m1_ctsn	I	SRADC4/PWM4_M0/UART0_CTSN_M1/SPI1_CS0n_M0/PDM_CLK_M1/I2S_SCLK_TX_M0/TKEY5_M0/GPIO0_C4_u	GRF_GPIO0C_IOMUX_H[2:0]=3'h3

Module Pin	Dir	Pad name	IOMUX
uart0_m1_rtsn	O	SRADC5/PWM5_M0/UART0_RTSN_M1/SPI1_CLK_M0/PDM_SDI_M1/I2S_LRCK_TX_M0/TKEY6_M0/GPIO0_C5_u	GRF_GPIO0C_IOMUX_H[6:4]=3'h3
uart0_m1_rx	I	SRADC6/PWM6_M0/UART0_RX_M1/SPI1_MOSI_M0/I2C0_SDA_M1/I2S_SDO_M0/TKEY7/PWM_AUDIO_L_M1/GPIO0_C6_u	GRF_GPIO0C_IOMUX_H[11:8]=4'h3
uart0_m1_tx	O	SRADC7/PWM7_M0/UART0_TX_M1/SPI1_MISO_M0/I2C0_SCL_M1/I2S_SDI_M0/TKEY8/PWM_AUDIO_R_M1/PMIC_INT_M1/GPIO0_C7_d	GRF_GPIO0C_IOMUX_H[15:12]=4'h3
UART1 Interface			
uart1_m0_ctsn	I	SRADC0/PWM0_M0/UART1_CTSN_M0/SPI0_CS0n_M1/I2S_SDO1_M0/SDMMC_CLKOUT/TKEY9/GPIO0_C0_u	GRF_GPIO0C_IOMUX_L[2:0]=3'h3
uart1_m0_rtsn	O	SRADC1/PWM1_M0/UART1_RTSN_M0/SPI0_CLK_M1/I2C1_SDA_M1/I2S_SCLK_RX_M0/SDMMC_CMD/TKEY10/GPIO0_C1_u	GRF_GPIO0C_IOMUX_L[7:4]=4'h3
uart1_m0_rx	I	SRADC2/PWM2_M0/UART1_RX_M0/SPI0_MOSI_M1/I2C1_SCL_M1/I2S_LRCK_RX_M0/SDMMC_D0/TKEY11/GPIO0_C2_u	GRF_GPIO0C_IOMUX_L[11:8]=4'h3
uart1_m0_tx	O	SRADC3/PWM3_M0/UART1_TX_M0/SPI0_MISO_M1/PDM_CLK_S_M1/I2S_MCLK_M0/TKEY4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GRF_GPIO0C_IOMUX_L[15:12]=4'h3
uart1_m1_ctsn	I	LCD_RDN/CIF_CLKOUT/UART1_CTS_N_M1/TKEY16/PMU_SLEEP/PMU_STATE2/CODEC_CLK_M1/AONJTAG_TRSTn/DSPJTAG_TRSTn/GPIO0_A4_u	GRF_GPIO0A_IOMUX_H[3:0]=4'h3
uart1_m1_rtsn	O	LCD_WRN/CIF_CLKIN/UART1_RTSN_M1/PDM_CLK_M0/TKEY17/PMU_STATE3/CODEC_SYNC_M1/GPIO0_A5_d	GRF_GPIO0A_IOMUX_H[7:4]=4'h3
uart1_m1_rx	I	LCD_D2/CIF_D2/UART1_RX_M1/PDM_SDI_M0/TKEY18/PMU_STATE4/CODEC_ADC_D_M1/GPIO0_A6_d	GRF_GPIO0A_IOMUX_H[10:8]=3'h3
uart1_m1_tx	O	LCD_D3/CIF_D3/UART1_TX_M1/PDM_CLK_S_M0/TKEY19/TEST_CLKOUT/CODEC_DAC_DL_M1/GPIO0_A7_d	GRF_GPIO0A_IOMUX_H[14:12]=3'h3
uart1_m2_rx	I	CODEC_CLK_M0/PWM4_M1/TKEY0_M1/UART1_RX_M2/TKEY_DRIVE_M4/I2C0_SDA_M3/PWM_AUDIO_L_M2/	GRF_GPIO0D_IOMUX_L[2:0]=3'h4

Module Pin	Di r	Pad name	IOMUX
		GPIO0_D0_u	
uart1_m2_tx	O	CODEC_SYNC_M0/PWM5_M1/TKEY1_M1/UART1_TX_M2/SPI1_CS0n_M2/I2C0_SCL_M3/PWM_AUDIO_R_M2/GPIO0_D1_u	GRF_GPIO0D_IOMUX_L[6:4]=3'h4
UART2 Interface			
uart2_m0_ctsn	I	CODEC_ADC_D_M0/PWM6_M1/TKEY2_M1/UART2_CTSN_M0/SPI1_CLK_M2/I2S_MCLK_M1/GPIO0_D2_u	GRF_GPIO0D_IOMUX_L[10:8]=3'h4
uart2_m0_rtsn	O	CODEC_DAC_DL_M0/PWM8/TKEY3_M1/UART2_RTSN_M0/SPI1_MOSI_M2/I2S_SCLK_TX_M1/GPIO0_D3_u	GRF_GPIO0D_IOMUX_L[14:12]=3'h4
uart2_m0_rx	I	PMIC_INT_M0/PWM9/TKEY4_M1/UART2_RX_M0/SPI1_MISO_M2/I2S_LRCK_TX_M1/GPIO0_D4_u	GRF_GPIO0D_IOMUX_H[2:0]=3'h4
uart2_m0_tx	O	I2C2_SDA_M0/PWM10/TKEY5_M1/UART2_TX_M0/I2C1_SDA_M3/I2S_SD00_M1/PWM_AUDIO_L_M3/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[6:4]=3'h4
uart2_m1_ctsn	I	LCD_D4/CIF_D4/UART2_CTSN_M1/SPI1_CS0n_M1/SPI_SLV_CSn/GPIO0_B0_u	GRF_GPIO0B_IOMUX_L[2:0]=3'h4
uart2_m1_rtsn	O	LCD_D5/CIF_D5/UART2_RTSN_M1/SPI1_CLK_M1/SPI_SLV_CLK/GPIO0_B1_u	GRF_GPIO0B_IOMUX_L[6:4]=3'h4
uart2_m1_rx	I	LCD_D6/CIF_D6/UART2_RX_M1/SPI1_MOSI_M1/SPI_SLV_MOSI/GPIO0_B2_u	GRF_GPIO0B_IOMUX_L[10:8]=3'h4
uart2_m1_tx	O	LCD_D7/CIF_D7/UART2_TX_M1/SPI1_MISO_M1/SPI_SLV_MISO/GPIO0_B3_u	GRF_GPIO0B_IOMUX_L[14:12]=3'h4

21.6 Application Notes

21.6.1 None FIFO Mode Transfer Flow

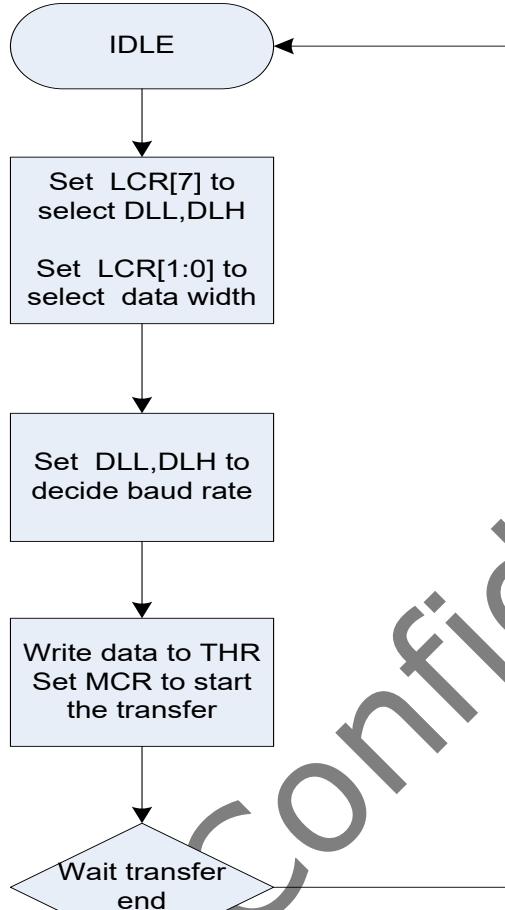


Fig. 21-8 UART none fifo mode

21.6.2 FIFO Mode Transfer Flow

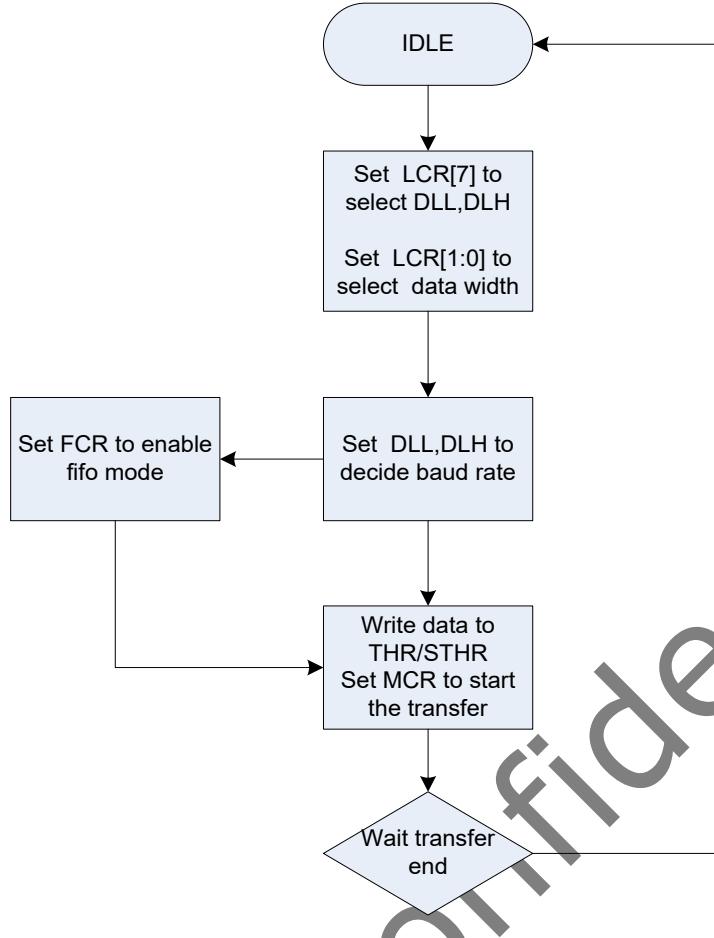


Fig. 21-9 UART fifo mode

The UART is an APB slave performing:

Serial-to-parallel conversion on data received from a peripheral device.

Parallel-to-serial conversion on data transmitted to the peripheral device.

The CPU reads and writes data and control/status information through the APB interface.

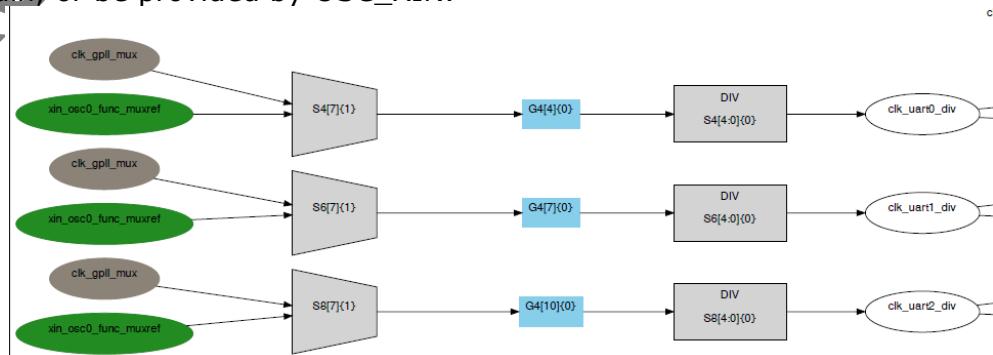
The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 64-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

21.6.3 Baud Rate Calculation

UART clock generation

The following figures shows the UART clock generation.

UART0, UART1 and UART2 source clocks can be selected from two PLL outputs (GPLL/CPLL). UART clocks can be generated by 1 to 64 division of its source clock, or can be fractionally divided again, or be provided by OSC_XIN.



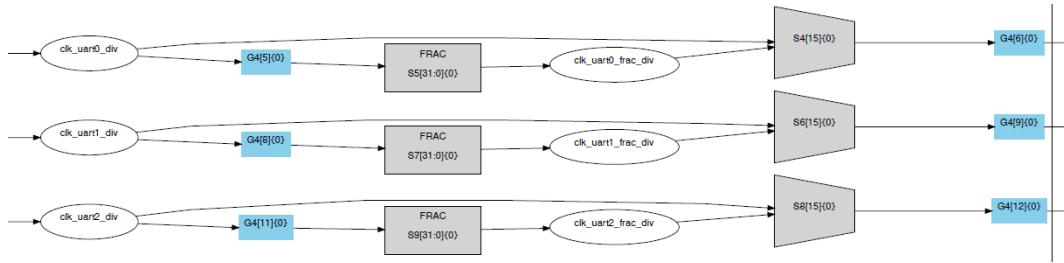


Fig. 21-10 UART clock generation

UART baud rate configuration

The following table provides some reference configuration for different UART baud rates.

Table 21-2 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Configure GPLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 8.
460.8 Kbps	Configure GPLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 2.
921.6 Kbps	Configure GPLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Configure UART_DLL to 1.
1.5 Mbps	Choose GPLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Configure UART_DLL to 1.
3 Mbps	Choose GPLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Configure UART_DLL to 1.
4 Mbps	Configure GPLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Configure UART_DLL to 1.

CTS_n and RTS_n Polarity Configurable:

The polarity of cts_n and rts_n ports can be configured by GRF registers.

GRF_SOC_CON2[2] (grf_uart0_rts_sel) used to configure the polarity of uart0 rts_n.

GRF_SOC_CON2[0] (grf_uart0_cts_sel) used to configure the polarity of uart0 cts_n.

GRF_SOC_CON2[3] (grf_uart2_rts_sel) used to configure the polarity of uart1rts_n.

GRF_SOC_CON2[1] (grf_uart2_cts_sel) used to configure the polarity of uart1cts_n.

GRF_SOC_CON1[15] (grf_uart2_rts_sel) used to configure the polarity of uart2 rts_n.

GRF_SOC_CON1[14] (grf_uart2_cts_sel) used to configure the polarity of uart2 cts_n.

When grf_uartX_cts_sel (X=0,1,2) is configured as 1'b1, cts_n is high active. Otherwise, low active.

When grf_uartX_rts_sel (X=0,1,2) is configured as 1'b1, rts_n is high active. Otherwise, low active.

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 can also read back the data on external pads using memory-mapped registers.

GPIO supports the following features:

- 32 bits APB data bus width
- Up to 32 independently configurable signals
- Software control registers with write mask for each bit of each signal
- Configurable debounce logic with a slow clock to debounce interrupts
- Configurable interrupt mode

22.2 Block Diagram

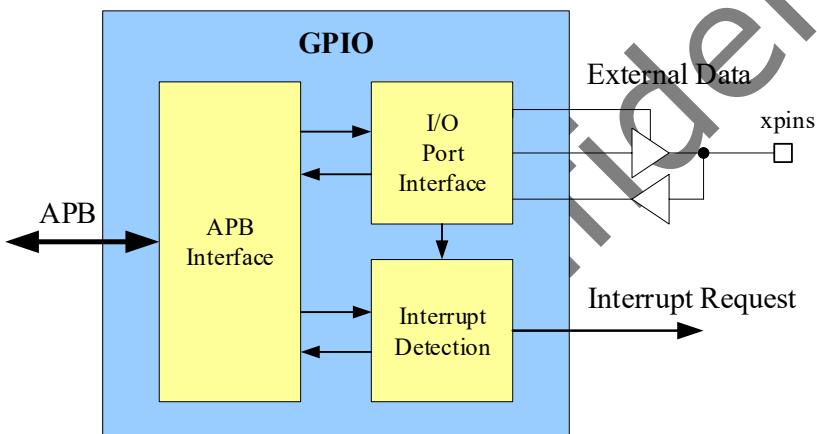


Fig. 22-1 GPIO Block Diagram

GPIO is comprised of:

- APB Interface

The APB Interface implements the APB slave operation. Its data bus width is 32 bits.

- I/O Port Interface

External data interface to or from I/O pads.

- Interrupt Detection

Interrupt interface to or from interrupt controller.

22.3 Function Description

22.3.1 Data Control

Under software control, the data and direction control for the signal are sourced from the port data registers (GPIO_SWPORT_DR_L/GPIO_SWPORT_DR_H) and direction control registers (GPIO_SWPORT_DRR_L/GPIO_SWPORT_DRR_H).

The direction of the external I/O pad is controlled by the value of the port data direction registers. The data written to these memory-mapped registers gets mapped onto an output signal (gpio_port_ddr) of the GPIO peripheral. This output signal controls the direction of an external I/O pad. The default data direction is Input.

The data written to the port data registers drives the output buffer (gpio_port_dr) of the I/O pad.

External data are input on the external data signal (gpio_ext_port). Reading the external signal register (GPIO_EXT_PORT) shows the value of this signal, regardless of the direction.

This register is read-only, meaning that it cannot be written from the APB software interface.

22.3.2 Interrupts

I/O port can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge
- Both the rising edge and the falling edge

The interrupts can be masked by programming the GPIO_INT_MASK_L/GPIO_INT_MASK_H registers. The interrupt status can be read before masking (GPIO_INT_RAWSTATUS) and after masking (GPIO_INT_STATUS).

For edge-sensitive interrupts, the Interrupt Service Routine (ISR) can clear the interrupt by writing a 1 to the corresponding bit of the GPIO_PORT_EOI_L/GPIO_PORT_EOI_H registers. This write operation also clears the interrupt status and raw status registers. Writing to the interrupt clear registers has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the interrupt raw status register until the interrupt source disappears, or it can write to the interrupt mask register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

The interrupts are combined into an active-high interrupt output signal. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever I/O port is configured for interrupts, the data direction must be set to Input. If the data direction is reprogrammed to Output, then any pending edge-sensitive interrupts are not lost. However, no new interrupts are generated, and level-sensitive interrupts are lost.

Interrupt signals are internally synchronized to a system clock pclk_intr, which is connected to the APB bus clock pclk. Therefore, the pclk needs to be running for interrupt detection.

22.3.3 Debounce Operation

The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When an input interrupt signal is debounced using a slow debounce clock (external input clock dbclk or internal divided clock dbclk_div), the signal must be active for a minimum of two cycles of the debounce clock to guarantee that it is registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one rising edge, it is not registered.

The debounce function can be controlled by programming the debounce enable registers (GPIO_DEBOUNCE_L/GPIO_DEBOUNCE_H), debounce clock divide enable registers (GPIO_DBCLK_DIV_EN_L/GPIO_DBCLK_DIV_EN_H) and debounce clock divide control register (GPIO_DBCLK_DIV_CON).

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 two GPIOs (GPIO0 ~ GPIO1), and each of them has same register group. Therefore, two GPIOs' register groups have two different base addresses.

22.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORT_DR_L	0x0000	W	0x00000000	Port Data Register (Low)
GPIO_SWPORT_DR_H	0x0004	W	0x00000000	Port Data Register (High)
GPIO_SWPORT_DDR_L	0x0008	W	0x00000000	Port Data Direction Register (Low)
GPIO_SWPORT_DDR_H	0x000c	W	0x00000000	Port Data Direction Register (High)
GPIO_INT_EN_L	0x0010	W	0x00000000	Interrupt Enable Register (Low)
GPIO_INT_EN_H	0x0014	W	0x00000000	Interrupt Enable register (High)
GPIO_INT_MASK_L	0x0018	W	0x00000000	Interrupt Mask Register (Low)
GPIO_INT_MASK_H	0x001c	W	0x00000000	Interrupt Mask Register (High)
GPIO_INT_TYPE_L	0x0020	W	0x00000000	Interrupt Level Register (Low)
GPIO_INT_TYPE_H	0x0024	W	0x00000000	Interrupt Level Register (High)
GPIO_INT_POLARITY_L	0x0028	W	0x00000000	Interrupt Polarity Register (Low)
GPIO_INT_POLARITY_H	0x002c	W	0x00000000	Interrupt Polarity Register (High)
GPIO_INT_BOTHEDGE_L	0x0030	W	0x00000000	Interrupt Both Edge Type Register (Low)
GPIO_INT_BOTHEDGE_H	0x0034	W	0x00000000	Interrupt Both Edge Type Register (High)
GPIO_DEBOUNCE_L	0x0038	W	0x00000000	Debounce Enable Register (Low)
GPIO_DEBOUNCE_H	0x003c	W	0x00000000	Debounce Enable Register (High)
GPIO_DBCLK_DIV_EN_L	0x0040	W	0x00000000	DBCLK Divide Enable Register (Low)
GPIO_DBCLK_DIV_EN_H	0x0044	W	0x00000000	DBCLK Divide Enable Register (High)
GPIO_DBCLK_DIV_CON	0x0048	W	0x00000001	DBCLK Divide Control Register
GPIO_INT_STATUS	0x0050	W	0x00000000	Interrupt Status Register
GPIO_INT_RAWSTATUS	0x0058	W	0x00000000	Interrupt Raw Status Register
GPIO_PORT_EOI_L	0x0060	W	0x00000000	Interrupt Clear Register (Low)
GPIO_PORT_EOI_H	0x0064	W	0x00000000	Interrupt Clear Register (High)
GPIO_EXT_PORT	0x0070	W	0x00000000	External Port Data Register
GPIO_VER_ID	0x0078	W	0x01000c2b	Version ID Register

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

22.4.2 Detail Register Description

GPIO_SWPORT_DR_L

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_swport_dr_low Values written to this register are output on the I/O signals for the lower 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DR_H

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_swport_dr_high Values written to this register are output on the I/O signals for the upper 16 bits of I/O Port if the corresponding data direction bits for I/O Port are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORT_DDR_L

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_swport_ddr_low Values written to this register independently control the direction of the corresponding data bit in the lower 16 bits of I/O Port. 1'b0: Input (default) 1'b1: Output

GPIO_SWPORT_DDR_H

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_swport_ddr_high Values written to this register independently control the direction of the corresponding data bit in the upper 16 bits of I/O Port. 1'b0: Input (default) 1'b1: Output

GPIO_INT_EN_L

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_en_low Allows each bit of the lower 16 bits of I/O Port to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output. 1'b0: Configure I/O Port bit as normal GPIO signal (default) 1'b1: Configure I/O Port bit as interrupt

GPIO_INT_EN_H

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_en_high Allows each bit of the upper 16 bits of I/O Port to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on I/O Port to become an interrupt; otherwise, I/O Port operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of I/O Port if the corresponding data direction register is set to Output. 1'b0: Configure I/O Port bit as normal GPIO signal (default) 1'b1: Configure I/O Port bit as interrupt

GPIO INT MASK L

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_mask_low Controls whether an interrupt on the lower 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 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 is unmasked (default) 1'b1: Interrupt is masked

GPIO INT MASK H

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_mask_high Controls whether an interrupt on the upper 16 bits of I/O Port can create an interrupt for the interrupt controller by not masking it. 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 is unmasked (default) 1'b1: Interrupt is masked

GPIO INT TYPE L

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_type_low Controls the type of interrupt that can occur on the lower 16 bits of I/O Port. 1'b0: Level-sensitive (default) 1'b1: Edge-sensitive

GPIO INT TYPE H

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_type_high Controls the type of interrupt that can occur on the upper 16 bits of I/O Port. 1'b0: Level-sensitive (default) 1'b1: Edge-sensitive

GPIO INT POLARITY L

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_polarity_low Controls the polarity of edge or level sensitivity that can occur on the lower 16 bits of I/O Port. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 1'b0: Active-low (default) 1'b1: Active-high

GPIO INT POLARITY H

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_int_polarity_high Controls the polarity of edge or level sensitivity that can occur on the upper 16 bits of I/O Port. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 1'b0: Active-low (default) 1'b1: Active-high

GPIO INT BOTHEdge L

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable</p>
15:0	RW	0x0000	<p>gpio_int_bothedge_low Controls the edge type of interrupt that can occur on the lower 16 bits of I/O Port. Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on I/O Port.</p> <p>The values programmed in the registers gpio_int_type 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_int_type and gpio_int_polarity registers.</p> <p>1'b0: No both-edge detection (default) 1'b1: Enable both-edge detection</p>

GPIO INT BOTEDGE H

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable</p>
15:0	RW	0x0000	<p>gpio_int_bothedge_high Controls the edge type of interrupt that can occur on the upper 16 bits of I/O Port. Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on I/O Port.</p> <p>The values programmed in the registers gpio_int_type 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_int_type and gpio_int_polarity registers.</p> <p>1'b0: No both-edge detection (default) 1'b1: Enable both-edge detection</p>

GPIO DEBOUNCE L

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_debounce_low Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 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 DEBOUNCE_H

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	gpio_debounce_high Controls whether an external signal of the lower 16 bits of I/O Port that is the source of an interrupt needs to be debounced to remove any spurious glitches. 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 DBCLK DIV EN_L

Address: Operational Base + offset (0x0040)

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

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	<p>gpio_dbclk_div_en_low</p> <p>Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the lower 16 bits of I/O Port. Whenever a 0 is written to a bit of this register, input clock dbclk is used as debounce clock; otherwise, the clock divided from dbclk is used. The clock divide factor depends on register gpio_dbclk_div_con.</p> <p>The values programmed in this register for this particular bit are not considered when the corresponding bit of register gpio_debounce is set to 0.</p> <p>1'b0: No divider for debounce clock (default) 1'b1: Enable divider for debounce clock</p>

GPIO DBCLK DIV EN H

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	<p>write_mask</p> <p>Write enable for lower 16 bits, each bit is individual.</p> <p>1'b0: Write access disable 1'b1: Write access enable</p>
15:0	RW	0x0000	<p>gpio_dbclk_div_en_high</p> <p>Controls whether to use the internal divided clock when debounce function is enabled for an external signal of the upper 16 bits of I/O Port. Whenever a 0 is written to a bit of this register, input clock dbclk is used as debounce clock; otherwise, the clock divided from dbclk is used. The clock divide factor depends on register gpio_dbclk_div_con.</p> <p>The values programmed in this register for this particular bit are not considered when the corresponding bit of register gpio_debounce is set to 0.</p> <p>1'b0: No divider for debounce clock (default) 1'b1: Enable divider for debounce clock</p>

GPIO DBCLK DIV CON

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000001	<p>gpio_dbclk_div_con</p> <p>dbclk_div = dbclk / (gpio_dbclk_div_con + 1)</p>

GPIO INT STATUS

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>gpio_int_status</p> <p>Interrupt status of I/O Port</p>

GPIO INT RAWSTATUS

Address: Operational Base + offset (0x0058)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_int_rawstatus Interrupt raw status of I/O Port (premasking bits)

GPIO PORT EOI L

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	W1C	0x0000	gpio_port_eoi_low Controls the clearing of edge type interrupts from the lower 16 bits of I/O Port. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts. 1'b0: No edge-sensitive interrupt clear (default) 1'b1: Clear edge-sensitive interrupt

GPIO PORT EOI H

Address: Operational Base + offset (0x0064)

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_mask Write enable for lower 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	W1C	0x0000	gpio_port_eoi_high Controls the clearing of edge type interrupts from the upper 16 bits of I/O Port. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. Writing to this register has no effect on level-sensitive interrupts. All interrupts are cleared when I/O Port is not configured for interrupts. 1'b0: No edge-sensitive interrupt clear (default) 1'b1: Clear edge-sensitive interrupt

GPIO EXT PORT

Address: Operational Base + offset (0x0070)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	gpio_ext_port This register always reflects the value of the signals on the external I/O Port.

GPIO VER ID

Address: Operational Base + offset (0x0078)

Bit	Attr	Reset Value	Description
31:0	RO	0x01000c2b	gpio_ver_id Version ID

22.5 Interface Description

Table 22-1 GPIO Interface Description

Module Pin	Direction	Pad Name	IOMUX Setting
GPIO0 Interface			
gpio0_port[3:0]	I/O	GPIO0_A[3:0]	GRF_GPIO0A_IOMUX_L[15:0]=16'h0
gpio0_port[7:4]	I/O	GPIO0_A[7:4]	GRF_GPIO0A_IOMUX_H[15:0]=16'h0
gpio0_port[11:8]	I/O	GPIO0_B[3:0]	GRF_GPIO0B_IOMUX_L[15:0]=16'h0
gpio0_port[15:12]	I/O	GPIO0_B[7:4]	GRF_GPIO0B_IOMUX_H[15:0]=16'h0
gpio0_port[19:16]	I/O	GPIO0_C[3:0]	GRF_GPIO0C_IOMUX_L[15:0]=16'h0
gpio0_port[23:20]	I/O	GPIO0_C[7:4]	GRF_GPIO0C_IOMUX_H[15:0]=16'h0
gpio0_port[27:24]	I/O	GPIO0_D[3:0]	GRF_GPIO0D_IOMUX_L[15:0]=16'h0
gpio0_port[30:28]	I/O	GPIO0_D[6:4]	GRF_GPIO0D_IOMUX_H[11:0]=12'h0
GPIO1 Interface			
gpio1_port[3:0]	I/O	GPIO1_A[3:0]	GRF_GPIO1A_IOMUX_L[15:0]=16'h0
gpio1_port[5:4]	I/O	GPIO1_A[5:4]	GRF_GPIO1A_IOMUX_H[7:0]=8'h0
gpio1_port[11:8]	I/O	GPIO1_B[3:0]	GRF_GPIO1B_IOMUX_L[15:0]=16'h0
gpio1_port[15:12]	I/O	GPIO1_B[7:4]	GRF_GPIO1B_IOMUX_H[15:0]=16'h0
gpio1_port[19:16]	I/O	GPIO1_C[3:0]	GRF_GPIO1C_IOMUX_L[15:0]=16'h0
gpio1_port[24]	I/O	GPIO1_D[0]	GRF_GPIO1D_IOMUX_L[3:0]=4'h0

Notes: Unused Module Pin is tied to zero! I=input, O=output, I/O=input/output, bidirectional

22.6 Application Notes

- Reading from an unused location or unused bits in a particular register always returns zeros. There is no error mechanism in the APB.
- Programming the GPIO registers for interrupt detection should be completed prior to enabling the interrupts in order to prevent spurious glitches on the interrupt output signal to the interrupt controller.

Chapter 23 Interrupt Controller

23.1 Overview

The INTC is a configurable, vectored interrupt controller for AMBA-based systems. It is an AMBA 2.0-compliant Advanced High-speed Bus (AHB) slave device.

The INTC has the following main features:

- Multiple IRQ normal interrupt sources
- Software interrupts
- Priority filtering
- Programmable interrupt priorities

23.2 Block Diagram

The figure below shows INTC block diagram.

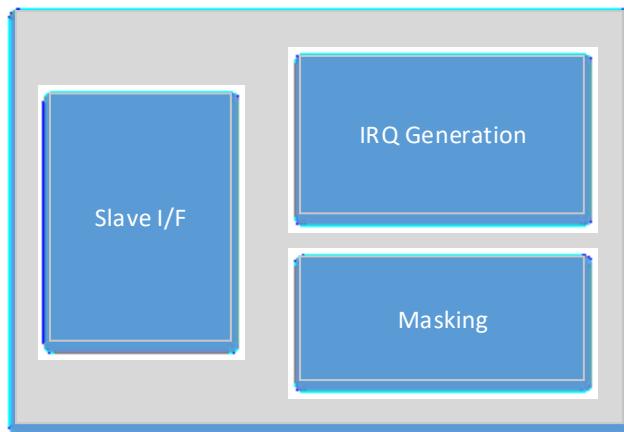


Fig. 23-1 INTC Block Diagram

23.3 Function Description

23.3.1 Overview

The INTC supports 27 IRQ sources that are processed to produce a single IRQ interrupt to the processor. All interrupt processing is combinational so that interrupts are propagated if the bus interface of the INTC is powered down. This means that reading any of the interrupt status registers (raw, status, or final_status) is simply returning the status of the combinational logic, since there are no flip-flops associated with these registers. It is the user's responsibility to ensure that the interrupts stay asserted until they are serviced.

23.3.2 Detailed Description

IRQ Interrupt Processing

The INTC processes these interrupt sources to produce a single IRQ interrupt to the processor; irq or irq_n. The processing of the interrupt sources is shown as followed.

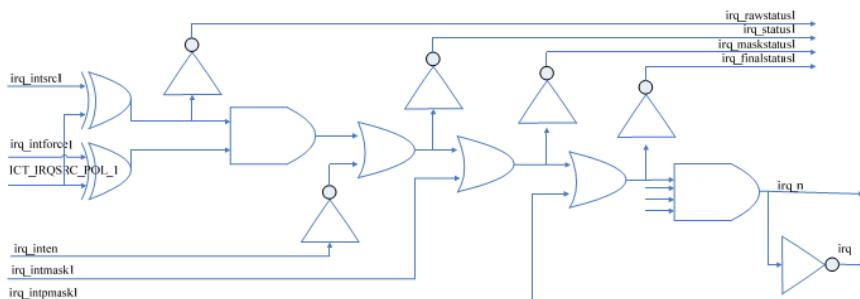


Fig. 23-2 IRQ interrupt processing for INTR

IRQ Software-Programmable Interrupts

The INTC supports forcing interrupts from software. To force an interrupt to be active, write to the corresponding bit in the irq_intforce registers (INTC_IRQ_INTFORCE_L or

INTC_IRQ_INTFORCE_H).

IRQ Enable and Masking

To enable each interrupt source independently, write a 1 to the corresponding bit of the irq_inten registers (INTC_IRQ_INTEN_L or INTC_IRQ_INTEN_H). To mask each interrupt source independently, write a 1 to the corresponding bit of the interrupt mask register (INTC_IRQ_MASKSTATUS_L/INTC_IRQ_MASKSTATUS_H). The reset value for each mask bit is 0 (unmasked).

IRQ Software-Programmable Priority Levels

The INTC supports optional software programmable priority levels. To change the priority level of an interrupt, you write the priority value to the corresponding priority level register in the memory map. There is a priority register for each of the interrupt sources, which can be programmed to one of 16 values from 0x0 to 0xf. Priority registers only exist for available interrupt sources.

IRQ Priority Filter

The INTC supports optional priority filtering. The function of the priority filtering logic is described as follows:

- Each interrupt source is configured to one of 16 priority levels, where 0 is the lowest priority.
- A system priority level can be programmed into the INTC_IRQ_PLEVEL REGISTER, which holds values from 0 to 15.
- The INTC filters out any interrupt source with a configured priority level less than the priority currently programmed in the irq_plevel register.

IRQ Interrupt Status Registers

The INTC includes up to four status registers used for querying the current status of any interrupt at various stages of the processing. All of the following status registers have the same polarity; a 1 indicates that an interrupt is active, a 0 indicates it is inactive:

- irq_rawstatus

The irq_rawstatus register (INTC_IRQ_RAWSTATUS_L/INTC_IRQ_RAWSTATUS_H) contains the state of the interrupt sources after being adjusted for input polarity. Each bit of this register is set to 1 if the corresponding interrupt source bit is active and is set to 0 if it is inactive.

- irq_status

The irq_status register (INTC_IRQ_STATUS_L/INTC_IRQ_STATUS_H) contains the state of all interrupts after the enabling stage, meaning that an active-high bit indicates that particular interrupt source is active and enabled.

- irq_maskstatus

The irq_maskstatus register (INTC_IRQ_MASKSTATUS_L/INTC_IRQ_MASKSTATUS_H) contains the state of all interrupts after the masking stage, meaning that an active-high bit indicates that particular interrupt source is active, enabled, and not masked.

- irq_finalstatus

This register (INTC_IRQ_FINALSTATUS_L/INTC_IRQ_FINALSTATUS_H) contains the state of all interrupts after the priority filtering stage, meaning an active-high bit indicates that particular interrupt source is active, enabled, not masked, and its configured priority level is greater or equal to the value programmed in the irq_plevel register. If priority filtering has not been selected, this register will contain the same value as the irq_maskstatus register (the final stage of processing).

23.4 Register Description

23.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
INTC_IRQ_INTEN_L	0x0000	W	0x00000000	IRQ interrupt source enable register (low)
INTC_IRQ_INTEN_H	0x0004	W	0x00000000	IRQ interrupt source enable register (high).

Name	Offset	Size	Reset Value	Description
INTC_IRQ_INTMASK_L	0x0008	W	0x00000000	IRQ interrupt source mask register (low).
INTC_IRQ_INTMASK_H	0x000c	W	0x00000000	IRQ interrupt source mask register (high).
INTC_IRQ_INTFORCE_L	0x0010	W	0x00000000	IRQ interrupt force register (low).
INTC_IRQ_INTFORCE_H	0x0014	W	0x00000000	IRQ interrupt force register (high).
INTC_IRQ_RAWSTATUS_L	0x0018	W	0x00000000	IRQ raw status register (low).
INTC_IRQ_RAWSTATUS_H	0x001c	W	0x00000000	IRQ raw status register (high)
INTC_IRQ_STATUS_L	0x0020	W	0x00000000	IRQ status register (low)
INTC_IRQ_STATUS_H	0x0024	W	0x00000000	IRQ status register (high)
INTC_IRQ_MASKSTATUS_L	0x0028	W	0x00000000	IRQ interrupt mask status register (low)
INTC_IRQ_MASKSTATUS_H	0x002c	W	0x00000000	IRQ interrupt mask status register (high)
INTC_IRQ_FINALSTATUS_L	0x0030	W	0x00000000	IRQ interrupt final status (low)
INTC_IRQ_FINALSTATUS_H	0x0034	W	0x00000000	IRQ interrupt final status (high)
INTC_IRQ_PLEVEL	0x00d8	W	0x00000000	IRQ System Priority Level Register
INTC_IRQ_PR_N	0x00e8	W	0x00000000	Interrupt priority level register
INTC_AHB_ICTL_COMP_VERSION	0x03f8	W	0x3230342a	Component Version Register
INTC_ICTL_COMP_TYPE	0x03fc	W	0x44571120	Component Type Register

Notes: **S**-ize: **B**- Byte (8 bits) access, **H**W- Half WORD (16 bits) access, **W**-WORD (32 bits) access

23.4.2 Detail Register Description

INTC_IRQ_INTEN_L

Address: Operational Base + offset (0x0000)

IRQ interrupt source enable register (low)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INTEN_L Interrupt enable bits for lower 32 interrupt sources. A 1 in any bit position enables the corresponding interrupt. 1'b0: disable interrupt 1'b1: enable interrupt

INTC_IRQ_INTEN_H

Address: Operational Base + offset (0x0004)

IRQ interrupt source enable register (high).

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INTEN_H Interrupt enable bit for upper 32 interrupt sources. A 1 in any bit position enables the corresponding interrupt. 1'b0: Disable interrupt 1'b1: Enable interrupt

INTC_IRQ_INTMASK_L

Address: Operational Base + offset (0x0008)

IRQ interrupt source mask register (low).

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INTMASK_L Interrupt mask bits for the lower 32 interrupt sources. A 1 in any bit position masks (disables) the corresponding interrupt. By default, all bits are unmasks. 1'b0: Unmask interrupt 1'b1: Mask interrupt

INTC_IRQ_INTMASK_H

Address: Operational Base + offset (0x000c)

IRQ interrupt source mask register (high).

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INTMASK_H Interrupt mask bits for the upper 32 interrupt sources. 1'b0: Unmask interrupt 1'b1: Mask interrupt

INTC_IRQ_INFORCE_L

Address: Operational Base + offset (0x0010)

IRQ interrupt force register (low).

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INFORCE_L Interrupt force bits for the lower 32 interrupt sources. Each bit in this register corresponds to one bit of the irq_intsrc input. The polarity of the bits in the register correspond to the polarity of the associated irq_intsrc input. If the interrupt input is configured to be active high, the corresponding bit in the register is also active high. 1'b0: Active low 1'b1: Active high

INTC_IRQ_INFORCE_H

Address: Operational Base + offset (0x0014)

IRQ interrupt force register (high).

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	IRQ_INTFORCE_H Interrupt force bits for the upper 32 interrupt sources. Each bit in this register corresponds to one bit of the irq_intsrcinput. The polarity of the bits in the register correspond to the polarity of the associated irq_intsrc input. If the interrupt input is configured to be active high, the corresponding bit in the register is also active high. The reset state of the force bits is always inactive. 1'b0: Active low 1'b1: Active high

INTC_IRQ_RAWSTATUS_L

Address: Operational Base + offset (0x0018)

IRQ raw status register (low).

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_RAWSTATUS_L Actual interrupt source.

INTC_IRQ_RAWSTATUS_H

Address: Operational Base + offset (0x001c)

IRQ raw status register (high)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_RAWSTATUS_H Actual interrupt source. These are the upper 32 interrupt sources.

INTC_IRQ_STATUS_L

Address: Operational Base + offset (0x0020)

IRQ status register (low)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_STATUS_L Interrupt status after the forcing and interrupt enabling stage. These are the interrupt status signals for the lower 32 interrupt sources.

INTC_IRQ_STATUS_H

Address: Operational Base + offset (0x0024)

IRQ status register (high)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_STATUS_H Interrupt status after the forcing and interrupt enabling stage. These are the interrupt status signals for the upper 32 interrupt sources.

INTC_IRQ_MASKSTATUS_L

Address: Operational Base + offset (0x0028)

IRQ interrupt mask status register (low)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_MASKSTATUS_L Interrupt status after the masking stage. These are the interrupt status signals for the lower 32 interrupt sources.

INTC_IRQ_MASKSTATUS_H

Address: Operational Base + offset (0x002c)

IRQ interrupt mask status register (high)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_MASKSTATUS_H Interrupt status after the masking stage. These are the interrupt status signals for the upper 32 interrupt sources.

INTC_IRQ_FINALSTATUS_L

Address: Operational Base + offset (0x0030)

IRQ interrupt final status (low)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_FINALSTATUS_L Interrupt status after the priority level filtering stage. These are the interrupt status signals for the lower 32 interrupt sources.

INTC_IRQ_FINALSTATUS_H

Address: Operational Base + offset (0x0034)

IRQ interrupt final status (high)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	IRQ_FINALSTATUS_H Interrupt status after the priority level filtering stage. These are the interrupt status signals for the upper 32 interrupt sources.

INTC_IRQ_PLEVEL

Address: Operational Base + offset (0x00d8)

IRQ System Priority Level Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	IRQ_PLEVEL Individual interrupt priority level. A register's value must be an integer from 0x0 to 0xf

INTC_IRQ_PR_N

Address: Operational Base + offset (0x00e8+4*N)

Interrupt priority level register

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

Bit	Attr	Reset Value	Description
3:0	RW	0x0	IRQ_PR_N Individual interrupt priority level. The range of N [number of registers] is from 0 to 63. A register's value must be an integer from 0x0 to 0xf.

INTC_AHB_ICTL_COMP_VERSION

Address: Operational Base + offset (0x03f8)

Component Version Register

Bit	Attr	Reset Value	Description
31:0	RO	0x3230342a	AHB_ICTL_COMP_VERSION Specific values for this register are described in the Releases Table

INTC_ICTL_COMP_TYPE

Address: Operational Base + offset (0x03fc)

Component Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44571120	ICCTL_COMP_TYPE Type number = 0x44_57_11_20

23.5 Application Notes

IRQ Source ID

Table 23-1 IRQ Source ID

IRQ ID	Source	Polarity
0	DMAC_CH0	High level
1	NA	High level
2	NA	High level
3	WDT2	High level
4	TIMER0_6CH_0(0)	High level
5	TIMER0_6CH_0(1)	High level
6	TIMER0_6CH_0(2)	High level
7	TIMER0_6CH_0(3)	High level
8	TIMER0_6CH_0(4)	High level
9	TIMER0_6CH_0(5)	High level
10	TIMER_1CH	High level
11	UART0	High level
12	UART1	High level
13	UART2	High level
14	A2B1_irq0	High level
15	A2B1_irq1	High level
16	A2B1_irq2	High level
17	A2B1_irq3	High level
18	XIPSFC0	High level
19	XIPSFC1	High level
20	HYPERX8_IRQ	High level
21	PMU_IRQ	High level

IRQ ID	Source	Polarity
22	DMAC_CH1	High level
23	DMAC_CH2	High level
24	DMAC_CH3	High level
25	DMAC_CH4	High level
26	DMAC_CH5	High level

Rockchip Confidential

Chapter 24 Timer

24.1 Overview

Timer is a programmable timer peripheral. This component is an APB slave device. In RK2206 there is one 6 channel timer named timer0 and one single channel timer named timer1. All timers count up from zero to a programmed value and generate an interrupt when the counter reaches the programmed value.

Timer supports the following features:

- Two operation modes: free-running and user-defined count.
- Maskable for each individual interrupt.

24.2 Block Diagram

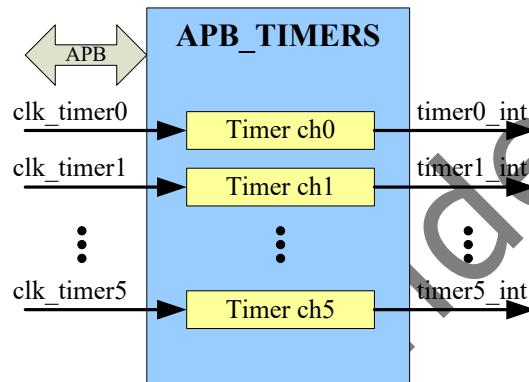


Fig. 24-1 Timer Block Diagram

The above figure shows the architecture of the APB timers (including six programmable timer channels, for single channel timer, there are not ch1 to ch5).

24.3 Function Description

24.3.1 Timer clock

The timer clock source is 24MHz.

24.3.2 Programming sequence

1. Initialize the timer by the TIMERn_CONTROLREG ($0 \leq n \leq 5$) 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—free-running or user-defined—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 ($0 \leq n \leq 5$) and TIMERn_LOAD_COUNT0 ($0 \leq n \leq 5$) register.
3. Enable the timer by writing a "1" to bit 0 of TIMERn_CONTROLREG ($0 \leq n \leq 5$).

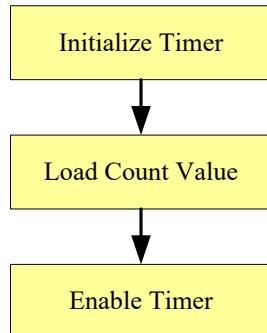


Fig. 24-2 Timer Usage Flow

24.3.3 Loading a timer count value

The initial value for each timer channel 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.

24.3.4 Timer mode selection

- **User-defined count mode** – Timer loads TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 registers (for descending timers) or zero (for incremental timers) as initial value. When the timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for incremental timers), it will not automatically reload the counter register. User need to disable timer firstly and follow the programming sequence to make timer work again.
- **Free-running mode** – Timer loads the TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for descending timers) or zero (for incremental timers) register as initial value. Timer will automatically reload the counter register, when timer counts down to 0 (for descending timers) or counts up to the value in TIMERn_LOAD_COUNT1 and TIMERn_LOAD_COUNT0 (for incremental timers).

24.4 Register Description

24.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER_TIMER_CHn_LOAD_COUNT0	0x0000	W	0x00000000	Low 32 bits value to be loaded into Timer channel n
TIMER_TIMER_CHn_LOAD_COUNT1	0x0004	W	0x00000000	High 32 bits value to be loaded into channel n
TIMER_TIMER_CHn_CURR_VALUE0	0x0008	W	0x00000000	Low 32 bits of current value of channel n
TIMER_TIMER_CHn_CURR_VALUE1	0x000c	W	0x00000000	High 32 bits of current value of channel n
TIMER_TIMER_CHn_CONTROL	0x0010	W	0x00000000	Timer Control Register

Name	Offset	Size	Reset Value	Description
TIMER TIMER CHn INTS STATUS	0x0018	W	0x00000000	Timer Interrupt Stauts Register
TIMER REVISION	0x00f0	W	0x0ca10006	Timer version

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

24.4.2 Detail Register Description

TIMER TIMER CHn LOAD COUNT0

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	count0 Load count low bits

TIMER TIMER CHn LOAD COUNT1

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	count1 Load count high bits

TIMER TIMER CHn CURR VALUE0

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value0 Current_cnt_low bits

TIMER TIMER CHn CURR VALUE1

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	current_value1 Current_cnt_high bits

TIMER TIMER CHn CONTROL

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	int_en Timer interrupt enable 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
1	RW	0x0	timer_mode Timer mode 0: free-running mode 1: user-defined count mode
0	RW	0x0	timer_en Timer enable 0: Disable 1: Enable

TIMER TIMER CHn INTSTATUS

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1 C	0x0	int_pd This register contains the interrupt status for Timer. Write 1 to this register will clear the interrupt.

TIMER REVISION

Address: Operational Base + offset (0x00f0)

Bit	Attr	Reset Value	Description
31:16	RO	0x0ca1	svn_revision Svn revision 16'd4536 for 6 channel timer 16'd4398 for single channel timer
15:14	RO	0x0	reserved
13:8	RO	0x00	up_down bit 8 indicates count up or count down for channel 0 timer bit 9 for channel 1 bit 10 for channel 2 bit 11 for channel 3 bit 12 for channel 4 bit 13 for channel 5 1'b1: count up 1'b0: count down if corresponding channel does not exist, then this bit is meaningless 6'b111111 for 6 channel timer Reserved for single channel timer, return 0
7:0	RO	0x06	total_channel The total channel number 8'd1: total one channel 8'd2: total two channel 8'd6 for 6 channel timer 8'd1 for single channel timer

24.5 Application Notes

24.5.1 Register Base Address

Table 24-1 Register Base Address

Module	CNT	Base Addr
TIMER0	CH0_BASE	0x40000000
	CH1_BASE	0x40000020
	CH2_BASE	0x40000040
	CH3_BASE	0x40000060
	CH4_BASE	0x40000080
	CH5_BASE	0x400000a0
TIMER1	CH0_BASE	0x41030000

24.5.2 Clock and Enable

In the chip, the timer_clk is from XIN24M, asynchronous to the pclk. When user disables the timer enables bit (bit 0 of TIMERn_CONTROLREG ($0 \leq n \leq 5$)), 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-configure registers when timer_en is low.



Fig. 24-3 Timing between timer_en and timer_clk

Please refer to function description section for the timer usage flow.

Chapter 25 WDT

25.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that caused by conflicting parts or programs in a SOC. The WDT will generate interrupt or reset signal when its counter reaches zero, then a reset controller will reset the system.

WDT supports the following features:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
 - Generate a system reset
 - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period
- There are three WDTs in the power domain MCU:
 - WDT0 for Cortex-M4
 - WDT1 for Cortex-M0
 - WDT2 for HiFi3
- All these WDTs can drive CRU to generate global software reset

25.2 Block Diagram

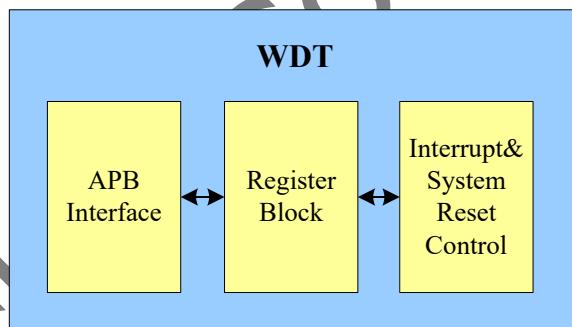


Fig. 25-1 WDT Block Diagram

WDT is comprised of:

- 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.

25.3 Function Description

25.3.1 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.

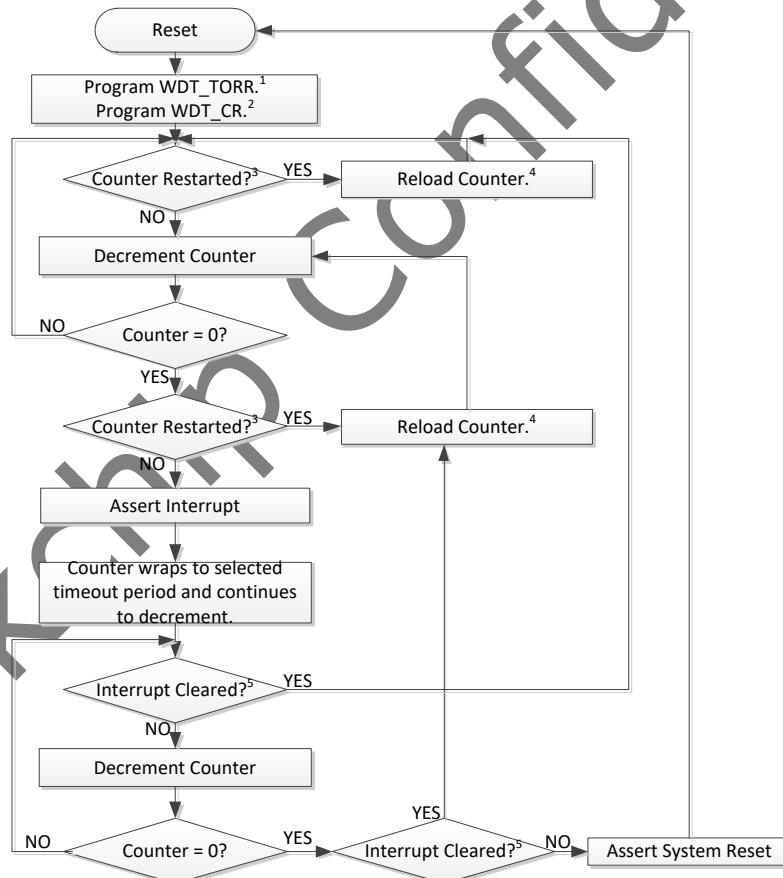
System Resets

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.

25.3.2 Programming Sequence



1. Select required timeout period.

2. Set reset pulse length, response mode, and enable WDT.

3. Write 0x76 to WDT_CRR.

4. Starts back to selected timeout period.

5. Can clear by reading WDT_EOI or restarting (kicking) the counter by writing 0x76 to WDT_CRR.

Fig. 25-2 WDT Operation Flow (RMOD=1)

25.4 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 three WDTs (WDT0 ~ WDT2), and each of them has same register group. Therefore, three WDTs' register groups have three different base addresses.

25.4.1 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: **S**-ize: **B**- Byte (8 bits) access, **H****W**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

25.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	<p>rst_pulse_length Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. The range of values available is 2 to 256 pclk cycles.</p> <p>3'b000: 2 pclk cycles 3'b001: 4 pclk cycles 3'b010: 8 pclk cycles 3'b011: 16 pclk cycles 3'b100: 32 pclk cycles 3'b101: 64 pclk cycles 3'b110: 128 pclk cycles 3'b111: 256 pclk cycles</p>
1	RW	0x1	<p>resp_mode Response mode. Selects the output response generated to a timeout. 1'b0: Generate a system reset 1'b1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset</p>

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>wdt_en WDT enable.</p> <p>This bit is used to enable and disable the WDT. When disabled, the counter does not decrement. Thus, no interrupts or system resets are generated. The WDT is used to prevent system lock-up. To prevent a software bug from disabling the WDT, once this bit has been enabled, it can be cleared only by a system reset.</p> <p>1'b0: WDT disabled 1'b1: WDT enabled</p>

WDT_TORR

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>timeout_period Timeout period.</p> <p>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).</p> <p>The range of values available for a 32-bit watchdog counter are:</p> <p>4'b0000: Time out of 64K Clocks 4'b0001: Time out of 128K Clocks 4'b0010: Time out of 256K Clocks 4'b0011: Time out of 512K Clocks 4'b0100: Time out of 1M Clocks 4'b0101: Time out of 2M Clocks 4'b0110: Time out of 4M Clocks 4'b0111: Time out of 8M Clocks 4'b1000: Time out of 16M Clocks 4'b1001: Time out of 32M Clocks 4'b1010: Time out of 64M Clocks 4'b1011: Time out of 128M Clocks 4'b1100: Time out of 256M Clocks 4'b1101: Time out of 512M Clocks 4'b1110: Time out of 1G Clocks 4'b1111: Time out of 2G Clocks</p>

WDT_CCVR

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:0	RO	0x0000ffff	<p>cur_cnt Current counter value register.</p> <p>This register, when read, is the current value of the internal counter.</p>

WDT_CRR

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	WO	0x00	<p>cnt_restart Counter restart register. 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.</p> <p>8'h76: Watchdog timer restart command</p>

WDT_STAT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	<p>int_status Interrupt status register. This register shows the interrupt status of the WDT. 1'b0: Interrupt is inactive 1'b1: Interrupt is active</p>

WDT_EOI

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	<p>int_clr Interrupt clear register. Clears the watchdog interrupt by reading this register. This can be used to clear the interrupt without restarting the watchdog counter.</p>

25.5 Application Notes

- Configure CRU_GLB_RST_CON to enable WDT triggered global software reset. Please refer to Chapter 3 for more information.

Chapter 26 Pulse Width Modulation (PWM)

26.1 Overview

The pulse-width modulator (PWM) feature is very common in embedded systems. It provides a way to generate a pulse periodic waveform for motor control or can act as a digital-to-analog converter with some external components.

The PWM module supports the following features:

- 4-built-in PWM channels
- Support capture mode
 - Measures the high/low polarity effective cycles of the 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 switch channel IO between channel 3 and channel0/1/2
 - Support a input filter to remove glitch
- Support continuous mode or one-shot mode
 - 32-bit period counter
 - 32-bit duty register
 - 32-bit current value register
 - PWM output polarity in inactive state and duty cycle polarity can be configured
 - Period and duty cycle are shadow buffered. Change takes effect when the end of the effective period is reached or when the channel is disabled
 - Programmable center or left aligned outputs, and change takes effect when the end of the effective period is reached or when the channel is disabled
 - 8-bit repeat counter for one-shot operation. One-shot operation will produce $N + 1$ periods of the waveform, where N is the repeat counter value, and generates a single interrupt at the end of operation
 - Continuous mode generates the waveform continuously, and does not generates any interrupts
- Support 2 main clock input, one is from crystal oscillator and the frequency is fixed, the other one is from PLL and the frequency can be configured. Each channel can select one of the clocks according to requirement.
- Support two-level frequency division.
- Available low-power mode to reduce power consumption when the channel is inactive.

26.2 Block Diagram

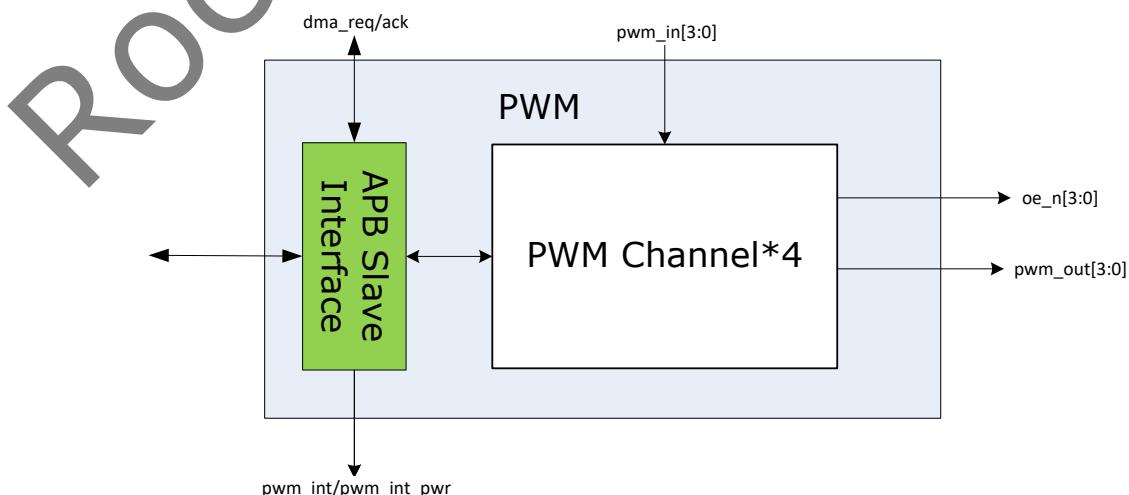


Fig. 26-1 PWM Block Diagram

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.

26.3 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.

26.3.1 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.

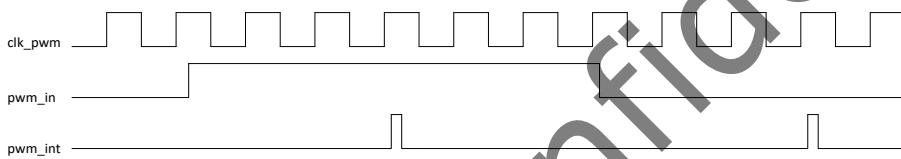


Fig. 26-2 PWM Capture Mode

The capture result can also 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.

26.3.2 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.

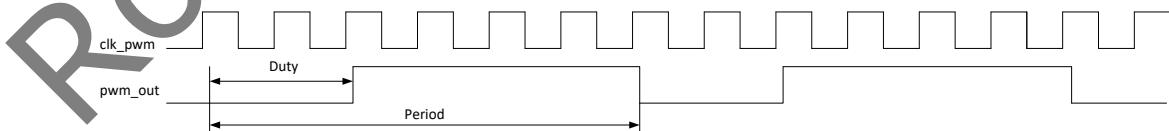


Fig. 26-3 PWM Continuous Left-aligned Output Mode

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.

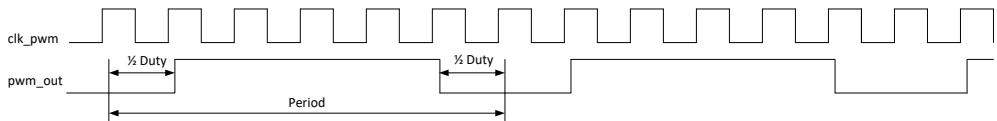


Fig. 26-4 PWM Continuous Center-aligned Output Mode

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).

26.3.3 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.

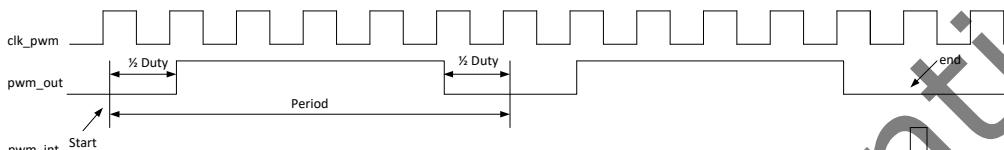


Fig. 26-5 PWM One-shot Center-aligned Output Mode

26.4 Register Description

26.4.1 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_VALUE_0	0x0098	W	0x00000000	Power Key Match Value 0 Register
PWM_PWRMATCH_VALUE_1	0x009c	W	0x00000000	Power Key Match Value 1 Register
PWM_PWRMATCH_VALUE_2	0x00a0	W	0x00000000	Power Key Match Value 2 Register
PWM_PWRMATCH_VALUE_3	0x00a4	W	0x00000000	Power Key Match Value 3 Register
PWM_PWRMATCH_VALUE_4	0x00a8	W	0x00000000	Power Key Match Value 4 Register
PWM_PWRMATCH_VALUE_5	0x00ac	W	0x00000000	Power Key Match Value 5 Register
PWM_PWRMATCH_VALUE_6	0x00b0	W	0x00000000	Power Key Match Value 6 Register
PWM_PWRMATCH_VALUE_7	0x00b4	W	0x00000000	Power Key Match Value 7 Register
PWM_PWRMATCH_VALUE_8	0x00b8	W	0x00000000	Power Key Match Value 8 Register

Name	Offset	Size	Reset Value	Description
PWM_PWRMATCH_VALUE_9	0x00bc	W	0x00000000	Power Key Match Value 9 Register
PWM_PWM3_PWRCAPTUR_E_VALUE	0x00cc	W	0x00000000	Channel 3 Power Key Capture Value Register
PWM_CHANNEL_IO_CTRL	0x00d0	W	0x00000000	Channel IO Control Register

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

26.4.2 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.
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	RO	0x0	reserved
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 1'b0: Non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source. 1'b1: Scaled clock is selected as PWM clock source.
8	RW	0x0	force_clk_en 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: Enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 1'b0: Disable lock 1'b1: Enable lock
5	RW	0x0	output_mode 1'b0: Left aligned mode 1'b1: Center aligned mode

Bit	Attr	Reset Value	Description
4	RW	0x0	<p>inactive_pol</p> <p>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.</p> <p>1'b0: Negative 1'b1: Positive</p>
3	RW	0x0	<p>duty_pol</p> <p>This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle.</p> <p>1'b0: Negative 1'b1: Positive</p>
2:1	RW	0x0	<p>pwm_mode</p> <p>2'b00: One shot mode, PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt.</p> <p>2'b01: Continuous mode, PWM produces the waveform continuously.</p> <p>2'b10: Capture mode, PWM measures the cycles of high/low polarity of input waveform.</p> <p>2'b11: Reserved</p>
0	RW	0x0	<p>pwm_en</p> <p>1'b0: Disabled 1'b1: Enabled</p> <p>If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.</p>

PWM_PWM1_CNT

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>CNT</p> <p>The 32-bit indicates current value of PWM Channel 1 counter. The counter runs at the rate of PWM clock.</p> <p>The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM1_PERIOD_HPR

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERIOD_HPR</p> <p>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.</p> <p>If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM PWM1 DUTY_LPR

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR</p> <p>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.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

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^{19}).
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	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 1'b0: Non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source. 1'b1: Scaled clock is selected as PWM clock source.
8	RW	0x0	force_clk_en 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: Enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 1'b0: Disable lock 1'b1: Enable lock
5	RW	0x0	output_mode 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: Negative 1'b1: Positive
3	RW	0x0	duty_pol This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: Negative 1'b1: Positive
2:1	RW	0x0	pwm_mode 2'b00: One shot mode, PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt. 2'b01: Continuous mode, PWM produces the waveform continuously. 2'b10: Capture mode, PWM measures the cycles of high/low polarity of input waveform. 2'b11: Reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en 1'b0: Disabled 1'b1: Enabled If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.

PWM_PWM2_CNT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CNT The 32-bit indicates current value of PWM Channel 2 counter. The counter runs at the rate of PWM clock. The value ranges from 0 to ($2^{32}-1$).

PWM_PWM2_PERIOD_HPR

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERIOD_HPR If PWM is operated at the continuous mode or one-shot mode, this value defines the period of the output waveform. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the bit [31:1] is taken into account and bit [0] always considered as 0. If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).

PWM_PWM2_DUTY_LPR

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DUTY_LPR If PWM is operated at the continuous mode or one-shot mode, this value defines the duty cycle of the output waveform. The PWM starts the output waveform with duty cycle. Note that, if the PWM is operated at the center-aligned mode, the period should be an even one, and therefore only the [31:1] is taken into account. If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform. This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).

PWM_PWM2_CTRL

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This fields defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2*N. If N is 0, it means that the clock is divided by 512(2*256).
15	RO	0x0	reserved
14:12	RW	0x0	prescale This field defines the prescale factor applied to input clock. The value N means that the input clock is divided by 2^N.
11	RO	0x0	reserved
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 1'b0: Non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source. 1'b1: Scaled clock is selected as PWM clock source.
8	RW	0x0	force_clk_en 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: Enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 1'b0: Disable lock 1'b1: Enable lock
5	RW	0x0	output_mode 1'b0: Left aligned mode 1'b1: Center aligned mode

Bit	Attr	Reset Value	Description
4	RW	0x0	<p>inactive_pol</p> <p>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.</p> <p>1'b0: Negative 1'b1: Positive</p>
3	RW	0x0	<p>duty_pol</p> <p>This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle.</p> <p>1'b0: Negative 1'b1: Positive</p>
2:1	RW	0x0	<p>pwm_mode</p> <p>2'b00: One shot mode, PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt.</p> <p>2'b01: Continuous mode, PWM produces the waveform continuously.</p> <p>2'b10: Capture mode. PWM measures the cycles of high/low polarity of input waveform.</p> <p>2'b11: Reserved</p>
0	RW	0x0	<p>pwm_en</p> <p>1'b0: Disabled 1'b1: Enabled</p> <p>If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.</p>

PWM_PWM3_CNT

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	<p>CNT</p> <p>The 32-bit indicates current value of PWM Channel 3 counter. The counter runs at the rate of PWM clock.</p> <p>The value ranges from 0 to ($2^{32}-1$).</p>

PWM_PWM3_PERIOD_HPR

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERIOD_HPR</p> <p>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.</p> <p>If PWM is operated at the capture mode, this value indicates the effective high polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM PWM3 DUTY_LPR

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>DUTY_LPR</p> <p>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.</p> <p>If PWM is operated at the capture mode, this value indicates the effective low polarity cycles of input waveform.</p> <p>This value is based on the PWM clock. The value ranges from 0 to ($2^{32}-1$).</p>

PWM PWM3 CTRL

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:24	RW	0x00	rpt This field defines the repeated effective periods of output waveform in one-shot mode. The value N means N+1 repeated effective periods.
23:16	RW	0x00	scale This field defines the scale factor applied to prescaled clock. The value N means the clock is divided by 2^N . If N is 0, it means that the clock is divided by 512(2^{19}).
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	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x0	clk_src_sel 1'b0: Select clk_pwm as root clock source, Clock is from PLL and the frequency can be configured. 1'b1: Select clk_pwm_capture as root clock source, Clock is from crystal oscillator and the frequency is fixed.
9	RW	0x0	clk_sel 1'b0: Non-scaled clock is selected as PWM clock source. It means that the prescale clock is directly used as the PWM clock source. 1'b1: Scaled clock is selected as PWM clock source.
8	RW	0x0	force_clk_en 1'b0: Disabled, when PWM channel is inactive state, the clk_pwm to PWM Clock prescale module is blocked to reduce power consumption. 1'b1: Enabled, the clk_pwm to PWM Clock prescale module is always enabled.
7	RW	0x0	ch_cnt_en 1'b0: Disabled 1'b1: Enabled
6	RW	0x0	conlock pwm period and duty lock to previous configuration 1'b0: Disable lock 1'b1: Enable lock
5	RW	0x0	output_mode 1'b0: Left aligned mode 1'b1: Center aligned mode
4	RW	0x0	inactive_pol This defines the output waveform polarity when PWM channel is in inactive state. The inactive state means that PWM finishes the complete waveform in one-shot mode or PWM channel is disabled. 1'b0: Negative 1'b1: Positive
3	RW	0x0	duty_pol This defines the polarity for duty cycle. PWM starts the output waveform with duty cycle. 1'b0: Negative 1'b1: Positive
2:1	RW	0x0	pwm_mode 2'b00: One shot mode, PWM produces the waveform within the repeated times defined by PWMx_CTRL_rpt 2'b01: Continuous mode, PWM produces the waveform continuously. 2'b10: Capture mode, PWM measures the cycles of high/low polarity of input waveform. 2'b11: Reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	pwm_en 1'b0: Disabled 1'b1: Enabled If the PWM is worked in the one-shot mode, this bit will be cleared at the end of operation.

PWM INTSTS

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:12	RO	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.
8	RO	0x0	CH0_Pol This bit is used in capture mode in order to identify the transition of the input waveform when interrupt is generated. When bit is 1, please refer to PWM0_PERIOD_HPR to know the effective high cycle of Channel 0 input waveform. Otherwise, please refer to PWM0_PERIOD_LPR to know the effective low cycle of Channel 0 input waveform. Write 1 to CH0_IntSts will clear this bit.
7	W1C	0x0	CH3_pwr_IntSts 1'b0: Channel 3 power key Interrupt not generated 1'b1: Channel 3 power key Interrupt generated
6	W1C	0x0	CH2_pwr_IntSts 1'b0: Channel 2 power key Interrupt not generated 1'b1: Channel 2 power key Interrupt generated

Bit	Attr	Reset Value	Description
5	W1 C	0x0	CH1_pwr_Intsts 1'b0: Channel 1 power keyInterrupt not generated 1'b1: Channel 1 power key Interrupt generated
4	W1 C	0x0	CH0_pwr_Intsts 1'b0: Channel 0 power key Interrupt not generated 1'b1: Channel 0 power key Interrupt generated
3	W1 C	0x0	CH3_Intsts 1'b0: Channel 3 Interrupt not generated 1'b1: Channel 3 Interrupt generated
2	W1 C	0x0	CH2_Intsts 1'b0: Channel 2 Interrupt not generated 1'b1: Channel 2 Interrupt generated
1	W1 C	0x0	CH1_Intsts 1'b0: Channel 1 Interrupt not generated 1'b1: Channel 1 Interrupt generated
0	W1 C	0x0	CH0_Intsts 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated

PWM INT EN

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	CH3_pwr_Int_en 1'b0: Channel 3 power key Interrupt disabled 1'b1: Channel 3 power key Interrupt enabled
6	RW	0x0	CH2_pwr_Int_en 1'b0: Channel 2 power key Interrupt disabled 1'b1: Channel 2 power key Interrupt enabled
5	RW	0x0	CH1_pwr_Int_en 1'b0: Channel 1 power key Interrupt disabled 1'b1: Channel 1 power key Interrupt enabled
4	RW	0x0	CH0_pwr_Int_en 1'b0: Channel 0 power key Interrupt disabled 1'b1: Channel 0 power key Interrupt enabled
3	RW	0x0	CH3_Int_en 1'b0: Channel 3 Interrupt disabled 1'b1: Channel 3 Interrupt enabled
2	RW	0x0	CH2_Int_en 1'b0: Channel 2 Interrupt disabled 1'b1: Channel 2 Interrupt enabled
1	RW	0x0	CH1_Int_en 1'b0: Channel 1 Interrupt disabled 1'b1: Channel 1 Interrupt enabled

Bit	Attr	Reset Value	Description
0	RW	0x0	CH0_Int_en 1'b0: Channel 0 Interrupt disabled 1'b1: Channel 0 Interrupt enabled

PWM FIFO CTRL

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:14	RO	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	W1 C	0x0	timieout_intsts Timeout interrupt

Bit	Attr	Reset Value	Description
2	W1 C	0x0	fifo_watermark_full_intsts This bit indicates the FIFO is Watermark Full.
1	W1 C	0x0	fifo_overflow_intsts This bit indicates the FIFO is overflow.
0	W1 C	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 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 Minor version
15:0	RW	0xb34	svn_version SVN version

PWM FIFO

Address: Operational Base + offset (0x0060)

Bit	Attr	Reset Value	Description
31	RO	0x0	pol This bit indicates the polarity of the lower 31-bit counter. 1'b0: Low 1'b1: High
30:0	RO	0x00000000	cycle_cnt This 31-bit counter indicates the effective cycles of high/low waveform.

PWM PWRMATCH CTRL

Address: Operational Base + offset (0x0080)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15	RW	0x0	CH3_pwrkey_int_ctrl 1'b0: Assert interrupt after key capture with power key match 1'b1: Assert interrupt after key capture without power key match
14:12	RO	0x0	reserved
11	RW	0x0	CH3_pwrkey_capture_ctrl 1'b0: Capture the value after interrupt 1'b1: Capture the value directly
10:8	RO	0x0	reserved
7	RW	0x0	CH3_pwrkey_polarity 1'b0: PWM in polarity is positive 1'b1: PWM in polarity is negative
6:4	RO	0x0	reserved
3	RW	0x0	CH3_pwrkey_enable 1'b0: Disabled 1'b1: Enabled
2:0	RO	0x0	reserved

PWM PWRMATCH LPRE

Address: Operational Base + offset (0x0084)

Bit	Attr	Reset Value	Description
31:16	RW	0x238c	cnt_max The maximum counter value
15:0	RW	0x22c4	cnt_min The minimum counter value

PWM PWRMATCH HPRE

Address: Operational Base + offset (0x0088)

Bit	Attr	Reset Value	Description
31:16	RW	0x11f8	cnt_max The maximum counter value
15:0	RW	0x1130	cnt_min The minimum counter value

PWM PWRMATCH LD

Address: Operational Base + offset (0x008c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM PWRMATCH HD ZERO

Address: Operational Base + offset (0x0090)

Bit	Attr	Reset Value	Description
31:16	RW	0x0294	cnt_max The maximum counter value
15:0	RW	0x01cc	cnt_min The minimum counter value

PWM_PWRMATCH_HD_ONE

Address: Operational Base + offset (0x0094)

Bit	Attr	Reset Value	Description
31:16	RW	0x06fe	cnt_max The maximum counter value
15:0	RW	0x0636	cnt_min The minimum counter value

PWM_PWRMATCH_VALUE0

Address: Operational Base + offset (0x0098)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value0 Power key match value 0

PWM_PWRMATCH_VALUE1

Address: Operational Base + offset (0x009c)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value1 Power key match value 1

PWM_PWRMATCH_VALUE2

Address: Operational Base + offset (0x00a0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value2 Power key match value 2

PWM_PWRMATCH_VALUE3

Address: Operational Base + offset (0x00a4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value3 Power key match value 3

PWM_PWRMATCH_VALUE4

Address: Operational Base + offset (0x00a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value4 Power key match value 4

PWM_PWRMATCH_VALUES

Address: Operational Base + offset (0x00ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value5 Power key match value 5

PWM_PWRMATCH_VALUE6

Address: Operational Base + offset (0x00b0)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value6 Power key match value 6

PWM_PWRMATCH_VALUE7

Address: Operational Base + offset (0x00b4)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value7 Power key match value 7

PWM_PWRMATCH_VALUE8

Address: Operational Base + offset (0x00b8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value8 Power key match value 8

PWM_PWRMATCH_VALUE9

Address: Operational Base + offset (0x00bc)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	pwrkey_match_value9 Power key match value 9

PWM_PWM3_PWRCAPTURE_VALUE

Address: Operational Base + offset (0x00cc)

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	pwrkey_capture_value Power key capture value

PWM_CHANNEL_IO_CTRL

Address: Operational Base + offset (0x00d0)

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved
18	RW	0x0	CH2_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled
17	RW	0x0	CH1_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled

Bit	Attr	Reset Value	Description
16	RW	0x0	CH0_and_CH3_switch_en 1'b0: Disabled 1'b1: Enabled
15:13	RO	0x0	reserved
12:4	RW	0x000	filter_number Filter window number
3	RW	0x0	CH3_input_filter_enable 1'b0: Disabled 1'b1: Enabled
2	RW	0x0	CH2_input_filter_enable 1'b0: Disabled 1'b1: Enabled
1	RW	0x0	CH1_input_filter_enable 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	CH0_input_filter_enable 1'b0: Disabled 1'b1: Enabled

26.5 Interface Description

Table 26-1 PWM Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
PWM0CH0	I/O	IO_SARADCch0_PWM0ch0m0_UART1ctsnm0_SPI0csn0m1_I2Ssdo1m0_SDMMCclkout_TOUCHkey9_VCCIO1gpio0a0	GRF_GPIO0C_IOMUX_SEL_L[2:0]=3'b10
		IO_PWM0ch0m1_UART0ctsnm0_SPI0csn0m0_I2C0sdam0_TOUCHkey0m0_TOUCHdrivem0_PWMaudiom0_I2C2sdam1_VCCIO0gpio-1d4	GRF_GPIO0B_IOMUX_SEL_H[3:0]=4'b1
PWM0CH1	I/O	IO_SARADCch1_PWM0ch1m0_UART1rtsnm0_SPI0clk1_I2C1sdam1_I2Sclkrxm0_SDMMCcmd_TOUCHkey10_VCCIO1gpio0a1	GRF_GPIO0C_IOMUX_SEL_L[6:4]=3'b10
		IO_PWM0ch1m1_UART0rtsnm0_SPI0clk0_I2C0sclm0_TOUCHkey1m0_PWMaudiorm0_I2C2sclm1_VCCIO0gpio-1d5	GRF_GPIO0B_IOMUX_SEL_H[6:4]=3'b1
PWM0CH2	I/O	IO_SARADCch2_PWM0ch2m0_UART1rxm0_SPI0mosim1_I2C1sclm1_I2Sclkrxm0_SDMMCd0_TOUCHkey11_VCCIO1gpio0a2	GRF_GPIO0C_IOMUX_SEL_L[10:8]=3'b10
		IO_PWM0ch2m1_UART0rxm0_SPI0mosim0_I2C1sdam0_TOUCHkey2m0_VCCIO0gpio-1d6	GRF_GPIO0B_IOMUX_SEL_H[10:8]=3'b1
PWM0CH3	I/O	IO_SARADCch3_PWM0ch3m0_UART1txm0_SPI0mosim1_PDMclksm1_I2Smclk0_TOUCHkey4m0_TOUCHdrivem1_VCCIO1gpio0a3	GRF_GPIO0C_IOMUX_SEL_L[14:12]=3'b10

Module Pin	Direction	Pin Name	IOMUX Setting
		IO_PWM0ch3m1_UART0txm0_SPI0misom0_I2C1sclm0_TOUCHkey3m0_TOUCHdrivem2_VCCIO0gpio-1d7	GRF_GPIO0B_IOMUX_SEL_H[14:12]=3'b1
PWM1CH0	I/O	IO_SARADCch4_PWM1ch0m0_UART0ctsnm1_SPI1csn0m0_PDMclk1_I2Ssclktxm0_TOUCHkey5m0_VCCIO1gpio0a4	GRF_GPIO0C_IOMUX_SEL_H[2:0]=3'b10
		IO_CODECclk0_PWM1ch0m1_TOUCHkey0m1_UART1rxm2_TOUCHdrivem4_I2C0sdam3_PWMaudiolm2_VCCIO2gpio0b0	GRF_GPIO0D_IOMUX_SEL_L[2:0]=3'b10
PWM1CH1	I/O	IO_SARADCch5_PWM1ch1m0_UART0rtsnm1_SPI1clk0_PDMsdim1_I2Slrcktxm0_TOUCHkey6m0_VCCI0gpio0a5	GRF_GPIO0C_IOMUX_SEL_H[6:4]=3'b10
		IO_CODECsyncm0_PWM1ch1m1_TOUCHkey1m1_UART1txm2_SPI1csn0m2_I2C0sclm3_PWMaudiorm2_VCCIO2gpio0b1	GRF_GPIO0D_IOMUX_SEL_L[6:4]=3'b10
PWM1CH2	I/O	IO_SARADCch6_PWM1ch2m0_UART0rxm1_SPI1mosim0_I2C0sdam1_I2Ssdo0m0_TOUCHkey7_PWMaudiolm1_VCCIO1gpio0a6	GRF_GPIO0C_IOMUX_SEL_H[10:8]=3'b10
		IO_CODECADCdatam0_PWM1ch2m1_TOUCHkey2m1_UART2ctsnm0_SPI1clk2_I2Smdclk1_VCCIO2gpio0b2	GRF_GPIO0D_IOMUX_SEL_L[10:8]=3'b10
PWM1CH3	I/O	IO_SARADCch7_PWM1ch3m0_UART0txm1_SPI1misom0_I2C0sclm1_I2Ssdim0_TOUCHkey8_PWMaudiorm1_PMICintm1_VCCIO1gpio0a7	GRF_GPIO0C_IOMUX_SEL_H[14:12]=3'b10
		IO_PWM1ch3m1_SPI0csn1_SPI1csn1_VCCIO0gpio-1d6	GRF_GPIO1D_IOMUX_SEL_L[2:0]=3'b1
PWM2CH0	I/O	IO_CODECADCdatam0_PWM2ch0_TOUCHkey3m1_UART2rtsnm0_SPI1mosim2_I2Ssclktxm1_VCCIO2gpio0b3	GRF_GPIO0D_IOMUX_SEL_L[14:12]=3'b10
PWM2CH1	I/O	IO_PMICintm0_PWM2ch1_TOUCHkey4m1_UART2rxm0_SPI1misom2_I2Slrcktxm1_VCCIO2gpio0b4	GRF_GPIO0D_IOMUX_SEL_H[2:0]=3'b10
PWM2CH2	I/O	IO_I2C2sdam0_PWM2ch2_TOUCHkey5m1_UART2txm0_I2C1sdam3_I2Ssdo0m1_PWMaudiolm3_VCCIO2gpio0b5	GRF_GPIO0D_IOMUX_SEL_H[6:4]=3'b10
PWM2CH3	I/O	IO_I2C2sclm0_PWM2ch3_TOUCHkey6m1_TOUCHdrivem5_I2C1sclm3_I2Ssdim1_PWMaudiorm3_VCCIO2gpio0b6	GRF_GPIO0D_IOMUX_SEL_H[10:8]=3'b10

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

26.6 Application Notes

26.6.1 PWM Capture Mode Standard Usage Flow

- Set PWM_PWMx_CTRL.pwm_en to '0' to disable the PWM channel.
- 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 and PWM_PWMx_CTRL.clk_src_sel.
- 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_CHANNEL_IO_CTRL.filter_number, then Enable the PWM_CHANNEL_IO_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.

26.6.2 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 and PWM_PWMx_CTRL.clk_src_sel.
3. Configure the channel 3 to work in the capture mode.
4. Configure the PWM_FIFO_CTRL.dma_mode_en and PWM_FIFO_CTRL fifo_mode_sel to enable the DMA mode. Configure PWM_FIFO_CTRL.almost_full_watermark at appropriate value.
5. Configure DMAC_BUS to transfer data from PWM to DDR.
6. Set PWM_CHANNEL_IO_CTRL.filter_number, then Enable the PWM_CHANNEL_IO_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.

26.6.3 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_LPREG, 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_CHANNEL_IO_CTRL.filter_number, then Enable the PWM_CHANNEL_IO_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_PWR_CAPTURE_VALUE to know the power key capture value.
11. Write '0' to PWM_PWM3_CTRL.pwm_en to disable the channel.

26.6.4 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 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.

26.6.5 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.

26.6.6 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 27 SARADC

27.1 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 its 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.

27.2 Block Diagram

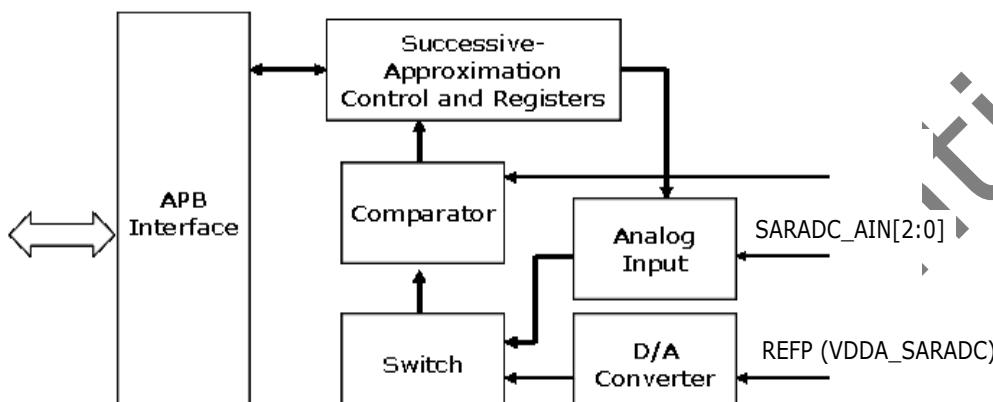


Fig. 27-1 RK2206SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[2:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

27.3 Function Description

27.3.1 APB Interface

In RK2206, SARADC works at single-sample operation mode.

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC. The detailed timing diagram will be shown in the following.

27.4 Register description

27.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.

Name	Offset	Size	Reset Value	Description
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	Delay between power up and start command

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

27.4.2 Detail Register Description

SARADC_DATA

Address: Operational Base + offset (0x0000)

This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS

Address: Operational Base + offset (0x0004)

The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

SARADC_CTRL

Address: Operational Base + offset (0x0008)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable 0: Disable 1: Enable
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down 1: ADC power up and reset. Start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up.
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 000: Input source 0 (SARADC_AIN[0]) 001: Input source 1 (SARADC_AIN[1]) 010: Input source 2 (SARADC_AIN[2]) 011: Input source 3 (SARADC_AIN[3]) 100: Input source 4 (SARADC_AIN[4]) 101: Input source 5 (SARADC_AIN[5]) Others: Reserved

SARADC_DLY_PU_SOC

Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	DLY_PU_SOC Delay between power up and start command The start signal will be asserted (DLY_PU_SOC + 2) sclk clock period later after power up.

27.5 Timing Diagram

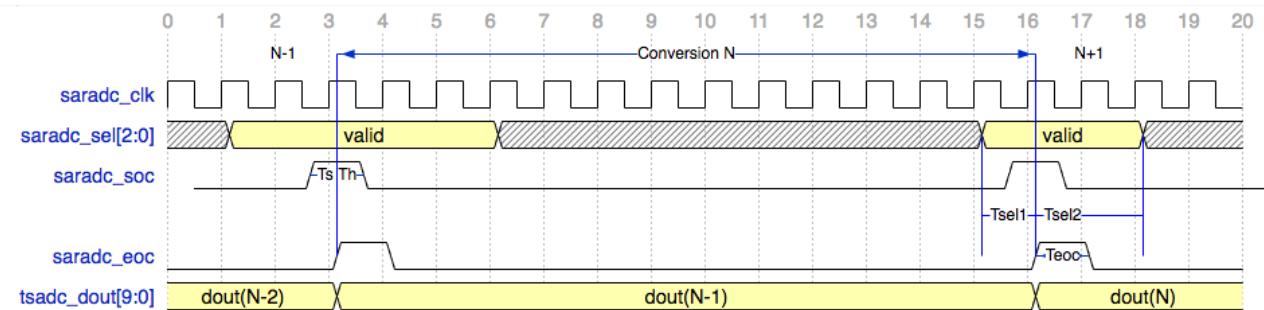


Fig. 27-2 SAR-ADC timing diagram in single-sample conversion mode

The following table shows the detailed value for timing parameters in the above diagram.

Table 27-1 RK2206 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	°C
Saradc Performance						
Resolution				10		bit
Effective Number of Bit	ENOB			9		bit
Differential Nonlinearity	DNL		-1		1	LSB
Integral Nonlinearity	INL		-2		2	LSB
Input Voltage Range	Vin		0		1	AVDD
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

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Digital Supply Current	I_{VDD}	Power Down		1		uA
		$f_s=1MS/s$		50		uA
		Power Down		1		uA

27.6 Application Notes

Steps of adc conversion:

- Write SARADC_CTRL[3] as 0 to power down ADC converter.
- Write SARADC_CTRL[2:0] as n to select adc channel(n).
- Write SARADC_CTRL[5] as 1 to enable adc interrupt.
- Write SARADC_CTRL[3] as 1 to power up adc converter.
- Wait for adc interrupt or poll SARADC_STAS register to assert whether the conversion is completed.
- Read the conversion result from SARADC_DATA[9:0].
- Note: The A/D converter was designed to operate at maximum 1MHz.

Rockchip Confidential

Chapter 28 Temperature-Sensor ADC(TSADC)

28.1 Overview

TSADC Controller module supports user-defined mode and automatic mode. User-defined mode refers, all the control signals of TSADC are entirely by software writing to register for direct control. Automatic mode refers to the module automatically polling TSADC output, and the results were checked. If the temperature is high within a period, an interrupt is generated to the processor; if the temperature is higher than TSADC_COMP_SHUT within a period, the resulting TSHUT will give to CRU module to reset the entire chip.

TSADC 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 can also be generated by hardware
- In Automatic Mode, the temperature of alarm(high/low temperature) interrupt is configurable
- In Automatic Mode, the temperature of system reset can be configurable
- Support only 1 channel TSADC
- 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
- 10-bit SARADC up to 50KS/s sampling rate

28.2 Block Diagram

TSADC controller comprises with:

- APB Interface
- TSADC control logic

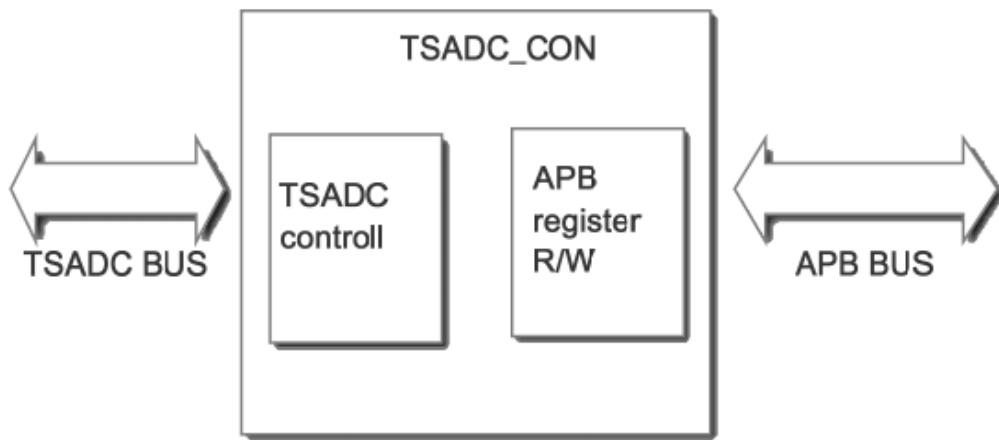


Fig. 28-1 TSADC Controller Block Diagram

28.3 Function Description

28.3.1 APB Interface

There is an APB Slave interface in TSADC Controller, which is used to configure the TSADC Controller registers and look up the temperature from the temperature sensor.

28.3.2 TSADC 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.

28.4 Register description

28.4.1 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	Interrupt enable control register
TSADC_INT_PD	0x000c	W	0x00000000	Interrupt pending status register
TSADC_DATA0	0x0020	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_DATA1	0x0024	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_COMPO_INT	0x0030	W	0x00000000	TSADC high temperature level for source 0
TSADC_COMPO_SHUT	0x0040	W	0x00000000	TSADC high temperature level for source 0
TSADC_HIGHT_INT_DEBOUNCE	0x0060	W	0x00000003	High temperature debounce
TSADC_HIGHT_TSHUT_DEBOUNCE	0x0064	W	0x00000003	High temperature debounce
TSADC_AUTO_PERIOD	0x0068	W	0x00010000	TSADC auto access period
TSADC_AUTO_PERIOD_H	0x006c	W	0x00010000	TSADC auto access period when temperature is high
TSADC_COMPO_LOW_INT	0x0080	W	0x00000000	TSADC low temperature level for source 0

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

28.4.2 Detail Register Description

TSADC_USER_CON

Address: Operational Base + offset (0x0000)

The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RO	0x0	adc_status ADC status (EOC) 0: ADC stop 1: Conversion in progress

Bit	Attr	Reset Value	Description
11:6	RW	0x08	inter_pd_soc Interleave between power down and start of conversion
5	RW	0x0	start When software writes 1 to this bit, start_of_conversion will be assert. This bit will be cleared after TSADC access finishing. When TSADC_USER_CON[4] = 1'b1 take effect.
4	RW	0x0	start_mode start mode 0: tsadc controller will assert start_of_conversion after "inter_pd_soc" cycles. 1: the start_of_conversion will be controlled by TSADC_USER_CON[5].
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down 1: ADC power up and reset
2:0	RW	0x0	adc_input_src_sel ADC input source selection (CH_SEL[2:0]). 000: Input source 0 (SARADC_AIN[0]) 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.
24:18	RO	0x0	reserved
17	RO	0x0	sample_dly_sel 0: AUTO_PERIOD is used 1: AUTO_PERIOD_HT is used
16	RO	0x0	auto_status 0: Auto mode stop 1: Auto mode in progress
15:14	RO	0x0	reserved
13	RW	0x0	src1_lt_en 0: Do not care low temperature of source 0 1: Enable the low temperature monitor of source 0

Bit	Attr	Reset Value	Description
12	RW	0x0	src0_lt_en 0: Do not care low temperature of source 0 1: Enable the low temperature monitor of source 0
11:9	RO	0x0	reserved
8	RW	0x0	tshut_polarity 0: Low active 1: High active
7:5	RO	0x0	reserved
4	RW	0x0	src0_en 0: Do not care the temperature of source 0 1: If the temperature of source 0 is too high, TSHUT will be valid
3:2	RO	0x0	reserved
1	RW	0x0	tsadc_q_sel temperature coefficient 1'b0: Use tsadc_q as output(positive temperature coefficient) 1'b1: Use(1024 - tsadc_q) as output (negative temperature coefficient) RK2206 is negative temprature coefficient, so please set this bit as 1'b1
0	RW	0x0	auto_en 0: TSADC controller works at user-define mode 1: TSADC controller works at auto mode

TSADC_INT_EN

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_en eoc_interrupt enable eoc_interrupt enable in user defined mode 0: Disable 1: Enable
15:13	RO	0x0	reserved
12	RW	0x0	lt_inten_src0 low temperature interrupt enable for src0 0: Disable 1: Enable
11:9	RO	0x0	reserved
8	RW	0x0	tshut_2cru_en_src0 0: TSHUT output to cru disabled. TSHUT output will always keep low. 1: TSHUT output works.
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	tshut_2gpio_en_src0 0: TSHUT output to gpio disabled. TSHUT output will always keep low. 1: TSHUT output works.
3:1	RO	0x0	reserved
0	RW	0x0	ht_inten_src0 high temperature interrupt enable for src0 0: Disable 1: Enable

TSADC_INT_PD

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	eoc_int_pd Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
15:13	RO	0x0	reserved
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.
11:5	RO	0x0	reserved
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.
3:1	RO	0x0	reserved
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[9:0]).

TSADC_COMP0_INT

Address: Operational Base + offset (0x0030)

TSADC high temperature level for source 0

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	tsadc_comp_src0 TSADC high temperature level. TSADC output is 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_HEIGHT_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_HEIGHT_TSHUT_DEBOUNCE

Address: Operational Base + offset (0x0064)

high temperature debounce

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x03	debounce TSADC controller will only generate interrupt or TSHUT when temperature is higher than COMP_SHUT for "debounce" times.

TSADC_AUTO_PERIOD

Address: Operational Base + offset (0x0068)

TSADC auto access period

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period when auto mode is enabled, this register controls the interleave between every two accessing of TSADC.

TSADC_AUTO_PERIOD_HT

Address: Operational Base + offset (0x006c)

TSADC auto access period when temperature is high

Bit	Attr	Reset Value	Description
31:0	RW	0x00010000	auto_period This register controls the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT

TSADC_COMP0_LOW_INT

Address: Operational Base + offset (0x0080)

TSADC low temperature level for source 0

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

28.5 Application Notes

28.5.1 Channel Select

The system has only one Temperature Sensor (channel 0).

28.5.2 Single-sample conversion

- The start timing

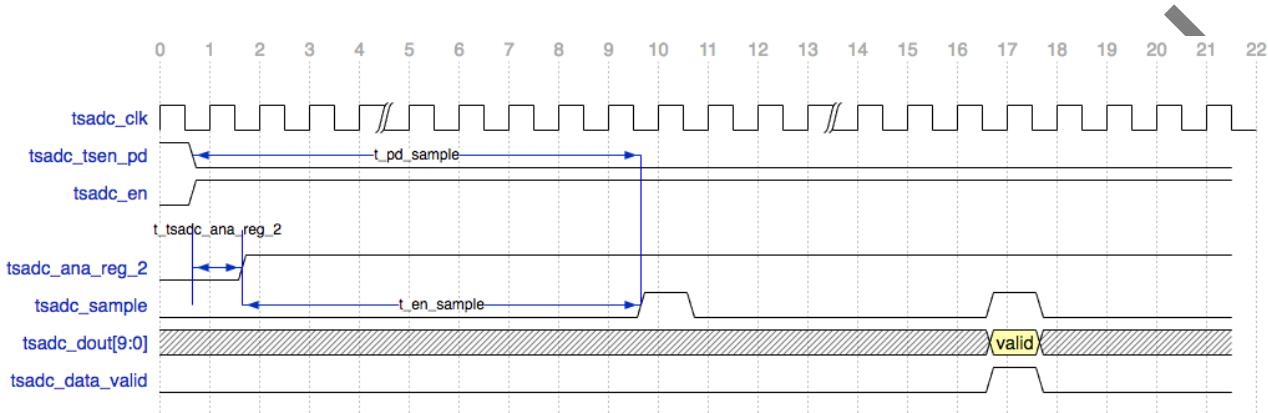


Fig. 28-2 the start flow to enable the sensor and adc

- Bypass mode($\text{grf_sco_con30[1]} = 1'b1$)

When the ADC bypass mode is enabled ($\text{tsadc_dig_bypass} = 1'b1$), the ADC will cost 14 clock cycles to complete the conversion. The tsadc_dout will keep the valid data output unchanged until the next clock cycles when the tsadc_sample is valid.

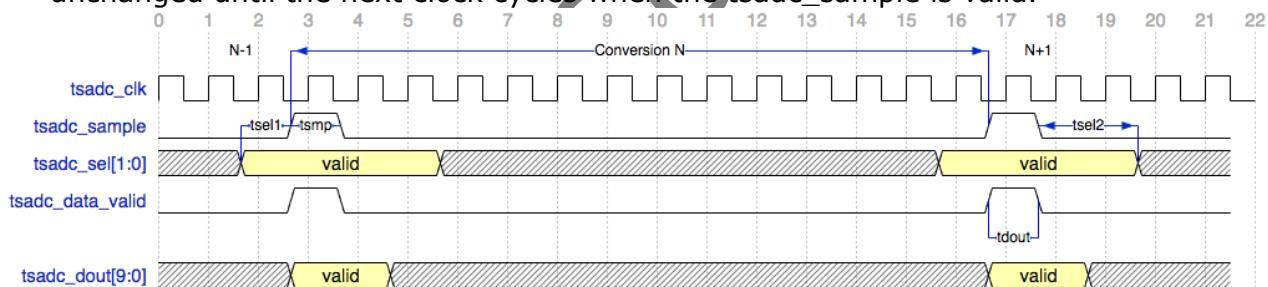


Fig. 28-3 tsadc timing diagram in bypass mode

- Normal mode

- $\text{Tsadc_clk_sel} = 1'b0$ ($\text{grf_sco_con30[0]} = 1'b0$)

The ADC will cost 15 clock cycles to complete the conversion. The tsadc_dout will keep the valid data output unchanged until the next two clock cycles when the tsadc_sample is valid.

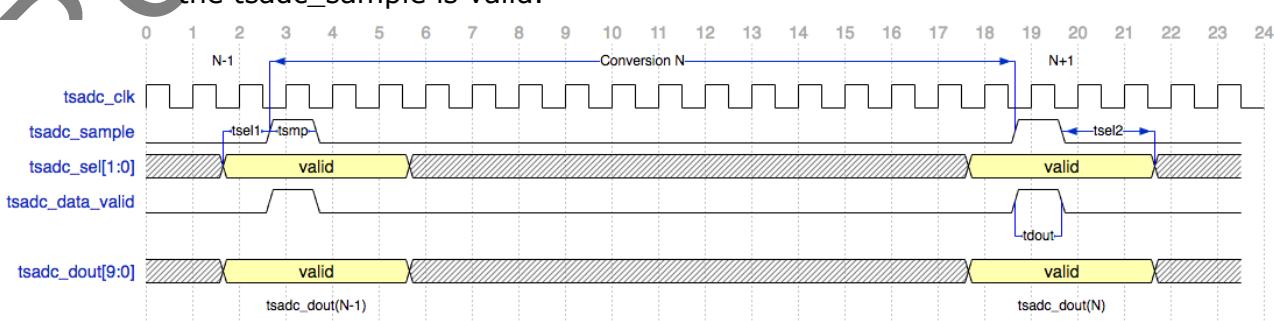


Fig. 28-4 tsadc timing diagram in normal mode with $\text{tsadc_clk_sel} = 1'b0$

- $\text{Tsadc_clk_sel} = 1'b1$ ($\text{grf_sco_con30[0]} = 1'b1$)

The ADC will cost 16 clock cycles to complete the conversion. The tsadc_dout will keep the valid data output unchanged until the next three clock cycles when the tsadc_sample is valid.

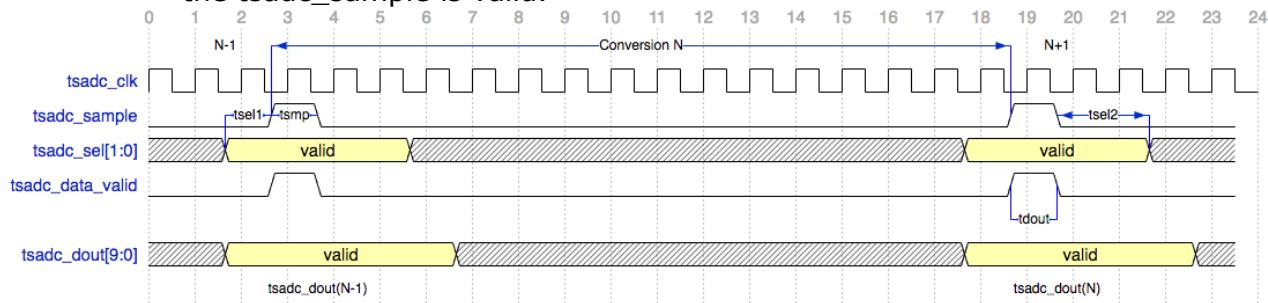


Fig. 28-5 tsadc timing diagram in normal mode with tsadc_clk_sel = 1'b1

28.5.3 Temperature-to-code mapping

Table 28-1 Temperature Code Mapping

Temperature/°C	ADC Output Data AUTO_CON[1] = 1'b0			ADC Output Data AUTO_CON[1] = 1'b1		
	Min	Typ	Max	Min	Typ	Max
-40	-	635	-	-	389	-
-35	-	626	-	-	398	-
-30	-	618	-	-	406	-
-25	-	609	-	-	415	-
-20	-	601	-	-	423	-
-15	-	592	-	-	432	-
-10	-	583	-	-	441	-
-5	-	575	-	-	449	-
0	-	566	-	-	458	-
5	-	557	-	-	467	-
10	-	549	-	-	475	-
15	-	540	-	-	484	-
20	-	531	-	-	493	-
25	-	522	-	-	502	-
30	-	514	-	-	510	-
35	-	505	-	-	519	-
40	-	496	-	-	528	-
45	-	487	-	-	537	-
50	-	479	-	-	545	-
55	-	470	-	-	554	-
60	-	461	-	-	563	-
65	-	452	-	-	572	-
70	-	443	-	-	581	-
75	-	435	-	-	589	-
80	-	426	-	-	598	-
85	-	417	-	-	607	-
90	-	408	-	-	616	-
95	-	399	-	-	625	-

Temperature/°C	ADC Output Data AUTO_CON[1] = 1'b0			ADC Output Data AUTO_CON[1] = 1'b1		
	Min	Typ	Max	Min	Typ	Max
100	-	390	-	-	634	-
105	-	381	-	-	643	-
110	-	372	-	-	652	-
115	-	363	-	-	661	-
120	-	354	-	-	670	-
125	-	345	-	-	679	-

Note:

Code to Temperature mapping of the Temperature sensor is a piece wise linear curve. Any temperature, code falling between to 2 give temperatures can be linearly interpolated.

Code to Temperature mapping should be updated based on silicon results.

28.5.4 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 temperature from TSADC_DATA.

28.5.5 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_PERIODDHT, configure the interleave between every two accessing of TSADC after the temperature is higher than COMP_SHUT or COMP_INT.
- Set TSADC_COMPn_INT(n=0,1), configure the high temperature level. If tsadc output is smaller than the value, which means the temperature is high, tsadc_int will be asserted.
- Set TSADC_COMPn_SHUT(n=0), configure the super high temperature level. If tsadc output is smaller than the value, which means the temperature is too high, TSHUT will be asserted.
- Set TSADC_INT_EN, you can enable the high temperature interrupt for all channels; 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 29 eFuse

29.1 Overview

In this chip, there is one eFuse controller and memory. It is organized as 128 bits by 8 one-time programmable electrical fuses with random access interface.

It is a type of non-volatile memory fabricated in standard CMOS logic process. The main features are as follows:

- Program condition : $VDDQ = 2.5V \pm 10\%$
- Program time : $10\mu s \pm 1\mu s$
- Read condition : $VDDQ = 0V$
- Provide standby mode

29.2 Block Diagram

In the following diagram, all the signals except power supply $VDDQ$ are controlled by registers. For detailed description, please refer to detailed register descriptions.

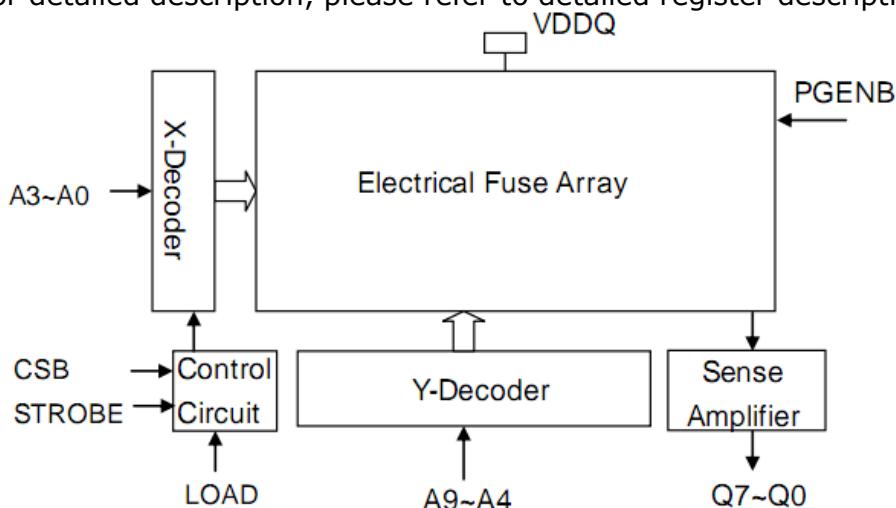


Fig. 29-1 eFuse Block Diagram

29.3 Function Description

eFuse has three operation modes. They are defined as program, read and standby.

● Program Mode

In order to enter programming mode, the following conditions need to be satisfied: $VDDQ$ is at high voltage, $LOAD$ signal is low, $PGENB$ signal is low, and CSB signal is low. All bits can be individually programmed (one at a time) with the proper address selected, the $STROBE$ signal assert high and the address bits satisfying setup and hold time with respect to $STROBE$.

● Read Mode

In order to enter read mode the following conditions need to be satisfied: $VDDQ$ is at ground, the $LOAD$ signal is high, the $PGENB$ signal is high, and the CSB is low. An entire 8-bit word of data can be read in one read operation with $STROBE$ being high and a proper address selected (address signals $A[7] \sim A[9]$ are "don't cares").

Table 29-1 1024-bits Dout Format

A[6]~A[0]	D[0]	D[1]	D[6]	D[7]
0000000	Fuse[0]	Fuse[128]	Fuse[768]	Fuse[896]
0000001	Fuse[1]	Fuse[129]	Fuse[769]	Fuse[897]
0000010	Fuse[2]	Fuse[130]	Fuse[770]	Fuse[770]
...
...
1111111	Fuse[127]	Fuse[131]	Fuse[895]	Fuse[1023]

● Standby Mode

Standby is defined when the macro is not being programmed or read. The conditions for

standby mode are: the LOAD signal is low, the STROBE signal is low, the CSB signal is high and PGEBN is high.

Table 29-2 list of allowed modes

Mode	CSB	STROBE	LOAD	PGENB	VQPS
Program mode	L	H	L	L	2.5V+/- 10%
Read mode	L	H	H	H	0V or floating
Standby mode	H	X	X	H	0V or floating

29.4 Register Description

29.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
<u>EFUSE MOD</u>	0x0000	W	0x00000006	Mode control register
<u>EFUSE RD MASK</u>	0x0004	W	0x00000000	Read mask control register
<u>EFUSE PG MASK</u>	0x0008	W	0x00000000	Program mask control register
<u>EFUSE INT CON</u>	0x0014	W	0x00000000	Interrupt control register
<u>EFUSE INT STATUS</u>	0x0018	W	0x00000000	Interrupt status register
<u>EFUSE USER CTRL</u>	0x001c	W	0x00000009	User mode control register
<u>EFUSE DOUT</u>	0x0020	W	0x00000000	Data output register
<u>EFUSE AUTO CTRL</u>	0x0024	W	0x00000000	Auto mode control register
<u>EFUSE T CSB P</u>	0x0028	W	0x000f0000	CSB program timing register
<u>EFUSE T PGEBN P</u>	0x002c	W	0x00000000	PGEBN program timing register
<u>EFUSE T LOAD P</u>	0x0030	W	0x00000000	LOAD program timing register
<u>EFUSE T ADDR P</u>	0x0034	W	0x00000000	ADDR program timing register
<u>EFUSE T STROBE P</u>	0x0038	W	0x00000000	STROBE program timing register
<u>EFUSE T CSB R</u>	0x003c	W	0x00000000	CSB read timing register
<u>EFUSE T PGEBN R</u>	0x0040	W	0x00000000	PGEBN read timing register
<u>EFUSE T LOAD R</u>	0x0044	W	0x00000000	LOAD read timing register
<u>EFUSE T ADDR R</u>	0x0048	W	0x00000000	ADDR read timing register
<u>EFUSE T STROBE R</u>	0x004c	W	0x00000000	STROBE read timing register
<u>EFUSE REVISION</u>	0x0050	W	0x00000011	Revision 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

EFUSE MOD

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:7	RO	0x0	Reserved
6	RW	0x0	rd_enb_user 1'b0: User mode read disable 1'b1: User mode read enable
5	RW	0x0	pg_en_user 1'b0: User mode program disable 1'b1: User mode program enable

Bit	Attr	Reset Value	Description
4	RW	0x0	strobe_pol 1'b0: Active high 1'b1: Active low
3	RW	0x0	load_pol 1'b0: Active high 1'b1: Active low
2	RW	0x0	pgenb_pol 1'b0: Active high 1'b1: Active low
1	RW	0x1	csb_pol 1'b0: Active high 1'b1: Active low
0	RW	0x0	work_mod 1'b0: Auto mode, software only configures the timing register and set enb to 1, hardware will auto do program and read until it's done. 1'b1: User mode, software can configure pins of eFuse as his desired to do program or read.

EFUSE INT CON

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	Reserved
12	RW	0x0	user_pg_mask_int_en 1'b0: Disable 1'b1: Enable
11	RW	0x0	user_rd_mask_int_en 1'b0: Disable 1'b1: Enable
10:9	RO	0x0	Reserved
8	RW	0x0	auto_pg_mask_int_en 1'b0: Disable 1'b1: Enable
7	RW	0x0	auto_rd_mask_int_en 1'b0: Disable 1'b1: Enable
6:1	RO	0x0	reserved
0	RW	0x0	finish_int_en 1'b0: Disable 1'b1: Enable

EFUSE INT STATUS

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:13	RO	0x0	Reserved
12	W1 C	0x0	user_pg_mask_int 1'b1: User mode read for current whole word can't be program.
11	W1 C	0x0	user_rd_mask_int 1'b1: User mode read for current word can't be read.
10:9	RO	0x0	Reserved
8	W1 C	0x0	auto_pg_mask_int 1'b1: Auto mode read for current whole word can't be program.
7	W1 C	0x0	auto_rd_mask_int 1'b1: Auto mode read for current word can't be read.
6:1	RO	0x0	Reserved
0	W1 C	0x0	finish_int 1'b1: Auto mode program or read operation is done.

EFUSE USER CTRL

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	Reserved
25:16	RW	0x000	addr_user User mode address
15:4	RO	0x0	Reserved
3	RW	0x0	pgenb_user User mode pgenb
2	RW	0x0	load_user User mode load
1	RW	0x0	strobe_user User mode strobe
0	RW	0x1	csb_user User mode csb

EFUSE DOUT

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	Reserved
7:0	RO	0x00	dout Read data register

EFUSE AUTO CTRL

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:26	RO	0x0	Reserved
25:16	RW	0x000	addr_auto Auto mode address
15:2	RO	0x0	Reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	pg_r 1'b0: Auto mode program operation 1'b1: Read mode read operation
0	R/W SC	0x0	enb 1'b0: Auto mode disable 1'b1: Auto mode enable Note, this bit is clear after program or read operation is done.

EFUSE T CSB P

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RO	0xf	t_csb_p_s Csb program start time. Unit: efuse_clk.
15:11	RO	0x0	Reserved
10:0	RW	0x000	t_csb_p_l Csb program last time. Unit: efuse_clk.

EFUSE T PGENB P

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_pgenb_p_s Pgenb program start time. Unit: efuse_clk.
15:11	RO	0x0	Reserved
10:0	RW	0x000	t_pgenb_p_l Pgenb program last time. Unit: efuse_clk.

EFUSE T LOAD P

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_load_p_s Load program start time. Unit: efuse_clk.
15:11	RO	0x0	Reserved
10:0	RW	0x000	t_load_p_l Load program last time. Unit: efuse_clk.

EFUSE T ADDR P

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_addr_p_s Address program start time. Unit: efuse_clk.

Bit	Attr	Reset Value	Description
15:11	RO	0x0	Reserved
10:0	RW	0x000	t_addr_p_l Address program last time. Unit: efuse_clk.

EFUSE T STROBE P

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	Reserved
23:16	RW	0x00	t_strobe_p_s Strobe program start time. Unit: efuse_clk.
15:11	RO	0x0	Reserved
10:0	RW	0x000	t_strobe_p_l Strobe program last time. Unit: efuse_clk.

EFUSE T CSB R

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_csb_r_s Cs read start time. Unit: efuse_clk.
15:10	RO	0x0	Reserved
9:0	RW	0x000	t_csb_r_l Cs read last time. Unit: efuse_clk.

EFUSE T PGEB R

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_pgenb_r_s Pgenb read start time. Unit: efuse_clk.
15:10	RO	0x0	Reserved
9:0	RW	0x000	t_pgenb_r_l Pgenb read last time. Unit: efuse_clk.

EFUSE T LOAD R

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_load_r_s Load read start time. Unit: efuse_clk.
15:10	RO	0x0	Reserved
9:0	RW	0x000	t_load_r_l Load read last time. Unit: efuse_clk.

EFUSE T ADDR R

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_addr_r_s Address read start time. Unit: efuse_clk.
15:10	RO	0x0	Reserved
9:0	RW	0x000	t_addr_r_l Address read last time. Unit: efuse_clk.

EFUSE T STROBE R

Address: Operational Base + offset (0x004c)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	Reserved
19:16	RW	0x0	t_strobe_r_s Strobe read start time. Unit: efuse_clk.
15:10	RO	0x0	Reserved
9:0	RW	0x000	t_strobe_r_l Strobe read last time. Unit: efuse_clk.

EFUSE REVISION

Address: Operational Base + offset (0x0050)

Bit	Attr	Reset Value	Description
31:8	RO	0x0	Reserved
7:0	RW	0x11	revision Revision 1.1

29.5 Timing Diagram

- When efuse128×8 is in program(PGM) mode.

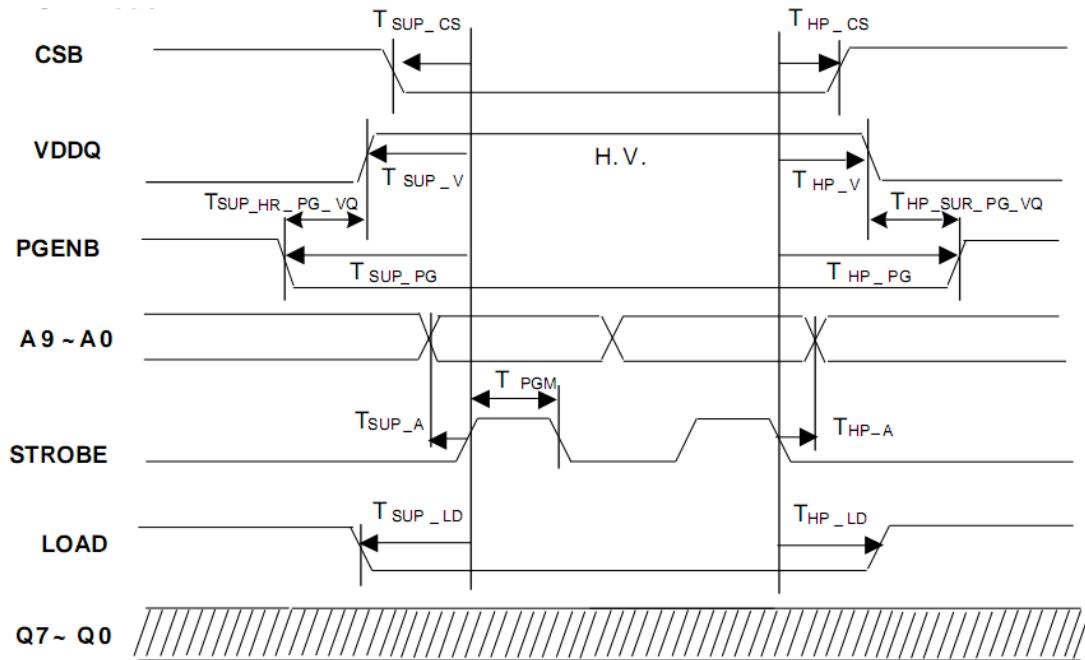


Fig. 29-2 efuse128×8 timing diagram in program mode

- When efuse128×8 is in read mode.

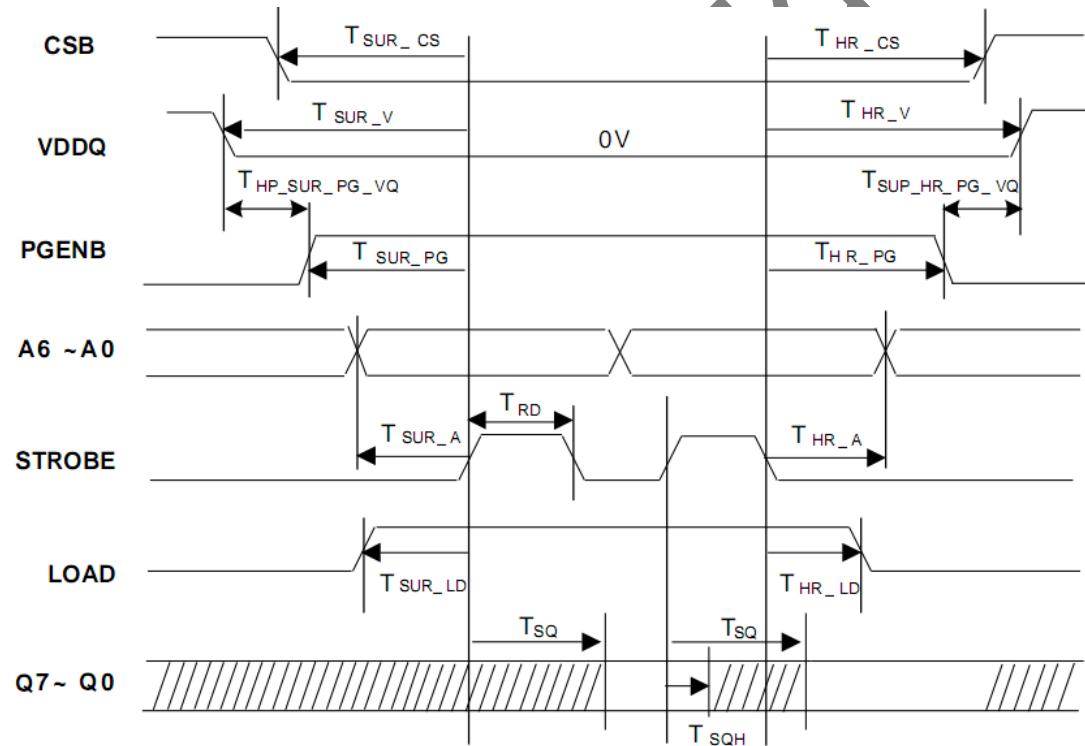


Fig. 29-3 efuse128×8 timing diagram in read mode

The min time of T_PGM is 10 us, the max time of T_PGM is 11 us.
Other timing parameters in above figure just need to be more than 40 ns.

29.6 Application Notes

During usage of eFuse, customers must pay more attention to the following items:

- In condition of program(PGM) mode, VDDQ= 2.5V±10%.
- Q0~Q7 will be reset to "0" once CSB at high.
- No data access allowed at the rising edge of CSB.
- User can select Auto mode or user mode.

Chapter 30 Touch Key Controller

30.1 Overview

The Touch Key Controller is used to determine whether a touch event is recognized. The Touch Key Controller supports following features:

- Support AMBA APB slave interface
- Support up to 20 channel touch events be detected at the same time
- Support two detect interrupts, one interrupt called posedge interrupt goes high when touch event is recognized, and the other interrupt called negedge interrupt goes high when touch event is disappeared.
- support configurable detect threshold
- support configurable detect out clock frequency

30.2 Block Diagram

The Touch Key Controller consists of the following main functional blocks.

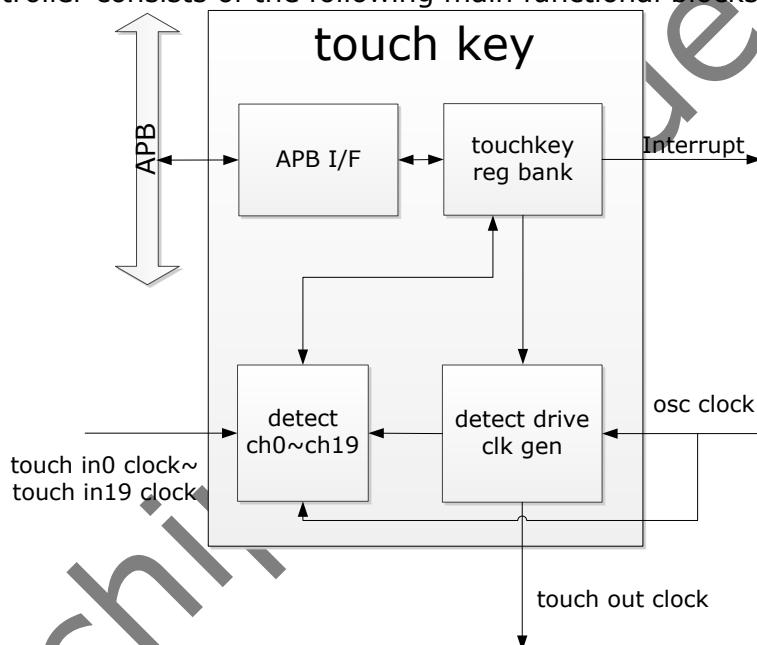


Fig. 30-1 Host Controller Block Diagram

- APB I/F: PB Slave Interface.
- touchkey reg bank: Register bank.
- touch drive clk gen: generate touch out clock, the clock frequency can be cnfigured.
- detect ch0~ch19: 20 channels for touch event detection, one channel for one touch event.

30.3 Function Description

30.3.1 touch drive clk gen

Once the processor starts touch key controller, touch drive clk gen block starts to generate touch out clock which is output to pad.

Touch out clock frequency can be cnfigured.

30.3.2 detect ch0~ch19

Detect ch0 detects the change phase between touch clock in0 and touch clock out. The phase difference change into OSC clock cycle. This cycle counter passes a low pass filter first, and then goes to a high pass filter. The filter result is compare to the threshold which setting by processor.

When the filter result exceed the threshold value, a posedge_irq generates. And once the posedge_irq goes high, the filter result start to do auto-decreasing operators, and once goes to zero, a negedge_irq generates. A posedge_irq means touch event is recognized, and a negedge_irq means touch event is gone. Both IRQ need to be cleared by CPU.

Interrupt can be generated in the other way. If the user wants to use one IRQ only, it is suggested to use posedge_irq only, and set the register CH_IRQ_SEL[0] to 1'b1. This will generate an interrupt every cycle when there is a touch event. And this IRQ can be self-clear every osc clock cycle.

Detect ch1~ch19 is similar to detect ch0, except detect ch0 is designed for touch clock in0, detect ch1 is designed for touch clock in1, and so on.

The cycle counter and the result of low pass/high pass filter can be accessed by processor.

30.4 Register Description

30.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TOUCH_DETECT_CH_STA_RT	0x0000	W	0x00000000	Touch Sensor Controller Signal
TOUCH_DETECT_CH_ENA_BLE0	0x0004	W	0x00000000	Touch Sensor Controller Signal
TOUCH_DETECT_CH_ENA_BLE1	0x0008	W	0x00000000	Touch Sensor Controller Signal
TOUCH_DETECT_CH_DIV	0x000c	W	0x000fffff	Drive Pulse Frequency Setting
TOUCH_DETECT_CH_IRQ_EN0	0x0010	W	0x00000000	Interrupt Enable Signal
TOUCH_DETECT_CH_IRQ_EN1	0x0014	W	0x00000000	Interrupt Enable Signal
TOUCH_DETECT_CH_IRQ_EN2	0x0018	W	0x00000000	Interrupt Enable Signal
TOUCH_DETECT_CH_IRQ_EN3	0x001c	W	0x00000000	Interrupt Enable Signal
TOUCH_DETECT_CH_IRQ_ST_POS	0x0020	W	0x00000000	Interrupt Status Of Key Touch Is Pressing
TOUCH_DETECT_CH_IRQ_ST_NEG	0x0024	W	0x00000000	Interrupt Status Once Key Touch Is Releasing
TOUCH_DETECT_CH_IRQ_RAW	0x0028	W	0x00000000	Interrupt Raw Status, Once CPU Get A Interrupt, Cpu Need To Read Bit To Check Whether There Is A Shorter Touch Or Longer Touch
TOUCH_DETECT_CH_IRQ_CLEAR0	0x002c	W	0x00000000	Interrupt Status Clear

Name	Offset	Size	Reset Value	Description
TOUCH_DETECT_CH_IRQ_CLEAR1	0x0030	W	0x00000000	Interrupt Status Clear
TOUCH_DETECT_CH_CHARGE_THRESHOLD	0x0034	W	0x00ffff	Charge Threshold
TOUCH_DETECT_CH_FILTER_STABLE_TIME	0x0038	W	0x000001ff	Time For Filter To Be Stable
TOUCH_DETECT_CH_IRQ_SEL	0x003c	W	0x00000000	IRQ Select
TOUCH_DETECT_CH_LOCK	0x0040	W	0x00000000	Lock Mode
TOUCH_DETECT_CH_RC_TYPE_SEL	0x0044	W	0x00000000	Rc Type Sel
TOUCH_DETECT_CH_RC_SPEED_STEP_CNT	0x0048	W	0x00000000	Rc Speed Cnt
TOUCH_DETECT_CH0_CNT	0x0100	W	0x00000000	Pulse Count From Channel 0
TOUCH_DETECT_CH0_CNT_DC	0x0104	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 0
TOUCH_DETECT_CH0_CNT_DO	0x0108	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 0
TOUCH_DETECT_CH0_CNT_FILTER	0x010c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 0
TOUCH_DETECT_CH1_CNT	0x0200	W	0x00000000	Pulse Count From Channel 1
TOUCH_DETECT_CH1_CNT_DC	0x0204	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 1
TOUCH_DETECT_CH1_CNT_DO	0x0208	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 1
TOUCH_DETECT_CH1_CNT_FILTER	0x020c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 1
TOUCH_DETECT_CH2_CNT	0x0300	W	0x00000000	Pulse Count From Channel 2
TOUCH_DETECT_CH2_CNT_DC	0x0304	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 2
TOUCH_DETECT_CH2_CNT_DO	0x0308	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 2
TOUCH_DETECT_CH2_CNT_FILTER	0x030c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 2
TOUCH_DETECT_CH3_CNT	0x0400	W	0x00000000	Pulse Count From Channel 3
TOUCH_DETECT_CH3_CNT_DC	0x0404	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 3
TOUCH_DETECT_CH3_CNT_DO	0x0408	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 3

Name	Offset	Size	Reset Value	Description
TOUCH_DETECT_CH3_CN_T_FILTER	0x040c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 3
TOUCH_DETECT_CH4_CN_T	0x0500	W	0x00000000	Pulse Count From Channel 4
TOUCH_DETECT_CH4_CN_T_DC	0x0504	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 4
TOUCH_DETECT_CH4_CN_T_DO	0x0508	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 4
TOUCH_DETECT_CH4_CN_T_FILTER	0x050c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 4
TOUCH_DETECT_CH5_CN_T	0x0600	W	0x00000000	Pulse Count From Channel 5
TOUCH_DETECT_CH5_CN_T_DC	0x0604	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 5
TOUCH_DETECT_CH5_CN_T_DO	0x0608	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 5
TOUCH_DETECT_CH5_CN_T_FILTER	0x060c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 5
TOUCH_DETECT_CH6_CN_T	0x0700	W	0x00000000	Pulse Count From Channel 6
TOUCH_DETECT_CH6_CN_T_DC	0x0704	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 6
TOUCH_DETECT_CH6_CN_T_DO	0x0708	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 6
TOUCH_DETECT_CH6_CN_T_FILTER	0x070c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 6
TOUCH_DETECT_CH7_CN_T	0x0800	W	0x00000000	Pulse Count From Channel 7
TOUCH_DETECT_CH7_CN_T_DC	0x0804	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 7
TOUCH_DETECT_CH7_CN_T_DO	0x0808	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 7
TOUCH_DETECT_CH7_CN_T_FILTER	0x080c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 7
TOUCH_DETECT_CH8_CN_T	0x0900	W	0x00000000	Pulse Count From Channel 8
TOUCH_DETECT_CH8_CN_T_DC	0x0904	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 8
TOUCH_DETECT_CH8_CN_T_DO	0x0908	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 8
TOUCH_DETECT_CH8_CN_T_FILTER	0x090c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 8
TOUCH_DETECT_CH9_CN_T	0x1000	W	0x00000000	Pulse Count From Channel 9

Name	Offset	Size	Reset Value	Description
TOUCH_DETECT_CH9_CN_T_DC	0x1004	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 9
TOUCH_DETECT_CH9_CN_T_DO	0x1008	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 9
TOUCH_DETECT_CH9_CN_T_FILTER	0x100c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 9
TOUCH_DETECT_CH10_CNT	0x1100	W	0x00000000	Pulse Count From Channel 10
TOUCH_DETECT_CH10_CNT_DC	0x1104	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 10
TOUCH_DETECT_CH10_CNT_DO	0x1108	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 10
TOUCH_DETECT_CH10_CNT_FILTER	0x110c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 10
TOUCH_DETECT_CH11_CNT	0x1200	W	0x00000000	Pulse Count From Channel 11
TOUCH_DETECT_CH11_CNT_DC	0x1204	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 11
TOUCH_DETECT_CH11_CNT_DO	0x1208	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 11
TOUCH_DETECT_CH11_CNT_FILTER	0x120c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 11
TOUCH_DETECT_CH12_CNT	0x1300	W	0x00000000	Pulse Count From Channel 12
TOUCH_DETECT_CH12_CNT_DC	0x1304	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 12
TOUCH_DETECT_CH12_CNT_DO	0x1308	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 12
TOUCH_DETECT_CH12_CNT_FILTER	0x130c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 12
TOUCH_DETECT_CH13_CNT	0x1400	W	0x00000000	Pulse Count From Channel 13
TOUCH_DETECT_CH13_CNT_DC	0x1404	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 13
TOUCH_DETECT_CH13_CNT_DO	0x1408	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 13
TOUCH_DETECT_CH13_CNT_FILTER	0x140c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 13
TOUCH_DETECT_CH14_CNT	0x1500	W	0x00000000	Pulse Count From Channel 14
TOUCH_DETECT_CH14_CNT_DC	0x1504	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 14
TOUCH_DETECT_CH14_CNT_DO	0x1508	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 14

Name	Offset	Size	Reset Value	Description
TOUCH_DETECT_CH14_CNT_FILTER	0x150c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 14
TOUCH_DETECT_CH15_CNT	0x1600	W	0x00000000	Pulse Count From Channel 15
TOUCH_DETECT_CH15_CNT_DC	0x1604	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 15
TOUCH_DETECT_CH15_CNT_DO	0x1608	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 15
TOUCH_DETECT_CH15_CNT_FILTER	0x160c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 15
TOUCH_DETECT_CH16_CNT	0x1700	W	0x00000000	Pulse Count From Channel 16
TOUCH_DETECT_CH16_CNT_DC	0x1704	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 16
TOUCH_DETECT_CH16_CNT_DO	0x1708	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 16
TOUCH_DETECT_CH16_CNT_FILTER	0x170c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 16
TOUCH_DETECT_CH17_CNT	0x1800	W	0x00000000	Pulse Count From Channel 17
TOUCH_DETECT_CH17_CNT_DC	0x1804	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 17
TOUCH_DETECT_CH17_CNT_DO	0x1808	W	0x00000000	Pulse Count Only With Offset After Low Pass Filter From Channel 17
TOUCH_DETECT_CH17_CNT_FILTER	0x180c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 17
TOUCH_DETECT_CH18_CNT	0x1900	W	0x00000000	Pulse Count From Channel 18
TOUCH_DETECT_CH18_CNT_DC	0x1904	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 18
TOUCH_DETECT_CH18_CNT_DO	0x1908	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 18
TOUCH_DETECT_CH18_CNT_FILTER	0x190c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 18
TOUCH_DETECT_CH19_CNT	0x2000	W	0x00000000	Pulse Count From Channel 19
TOUCH_DETECT_CH19_CNT_DC	0x2004	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 19
TOUCH_DETECT_CH19_CNT_DO	0x2008	W	0x00000000	Pulse Count With Offset After Low Pass Filter From Channel 19
TOUCH_DETECT_CH19_CNT_FILTER	0x200c	W	0x00000000	Pulse Count After RC Low Pass Filter From Channel 19

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

30.4.2 Detail Register Description

TOUCH_DETECT_CH_START

Address: Operational Base + offset (0x0000)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0	reserved
0	RW	0x0	start Start enable. 1'b0: touch detect keep idle state 1'b1: start touch detect

TOUCH_DETECT_CH_ENABLE0

Address: Operational Base + offset (0x0004)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	enable Enable signal. Channel enable for channel0~channel15, each bit is individual. 1'b0: Channel disable 1'b1: Channel enable

TOUCH_DETECT_CH_ENABLE1

Address: Operational Base + offset (0x0008)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x0	reserved
3:0	RW	0x0	enable Enable signal. Channel enable for channel16~channel19, each bit is individual. 1'b0: Channel disable 1'b1: Channel enable

TOUCH_DETECT_CH_DIV

Address: Operational Base + offset (0x000c)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0xfffffff	drive_pulse_div Pulse out divider register, divide value equal to $2^{\text{charge_pulse_div}}$, charge_pulse_div is equal to $2*2^{16-1}$.

TOUCH DETECT CH IRQ EN0

Address: Operational Base + offset (0x0010)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:0	RW	0x0000	charge_irq_pos_en Touch event detected interrupt enable. Interrupt enable for channel0~channel15, each bit is individual. 1'b0: Disable 1'b1: Enable

TOUCH DETECT CH IRQ EN1

Address: Operational Base + offset (0x0014)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x0	reserved
3:0	RW	0x0	charge_irq_pos_en Touch event detected interrupt enable. Interrupt enable for channel16~channel19, each bit is individual. 1'b0: Disable 1'b1: Enable

TOUCH DETECT CH IRQ EN2

Address: Operational Base + offset (0x0018)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	charge_irq_neg_en Touch event disappeared interrupt enable. Interrupt enable for channel0~channel15, each bit is individual. 1'b0: Disable 1'b1: Enable

TOUCH DETECT CH IRQ EN3

Address: Operational Base + offset (0x001c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:4	RO	0x0	reserved
3:0	RW	0x0	charge_irq_neg_en Touch event disappeared interrupt enable. Interrupt enable for channel16~channel19, each bit is individual. 1'b0: Disable 1'b1: Enable

TOUCH DETECT CH IRQ ST POS

Address: Operational Base + offset (0x0020)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	charge_irq Touch event detected interrupts for channel0~channel19.

TOUCH DETECT CH IRQ ST NEG

Address: Operational Base + offset (0x0024)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	charge_irq Touch event disappeared interrupts for channel0~channel19

TOUCH DETECT CH IRQ RAW

Address: Operational Base + offset (0x0028)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	charge_irq Charge counter bigger than charge_threshold.

TOUCH DETECT CH IRQ CLEAR0

Address: Operational Base + offset (0x002c)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	charge_irq_pos_clear Write 1'b1 clear the corresponding interrupt.

TOUCH DETECT CH IRQ CLEAR1

Address: Operational Base + offset (0x0030)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	charge_irq_neg_clear Write 1'b1 clear the corresponding interrupt.

TOUCH DETECT CH CHARGE THRESHOLD

Address: Operational Base + offset (0x0034)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0xfffff	charge_threshold Charge threshold value.

TOUCH DETECT CH FILTER STABLE TIME

Address: Operational Base + offset (0x0038)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x01ff	filter_threshold Filter threshold value, setting the minimum value.

TOUCH DETECT CH IRQ SEL

Address: Operational Base + offset (0x003c)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:1	RO	0x0	reserved
0	RW	0x0	irq_sel IRQ select signal. 1'b0: IRQ generate according to value after RC filter 1'b1: IRQ generate every drive clock cycle

TOUCH DETECT CH LOCK

Address: Operational Base + offset (0x0040)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0	reserved
1	RW	0x0	ch_kick Kick the lock signal.
0	RW	0x0	ch_lock Enable signal. 1'b0: normal operate 1'b1: lock mode, clk_drv only active once Note: This bit is self controll when lock mode is enable.

TOUCH DETECT CH RC TYPE SEL

Address: Operational Base + offset (0x0044)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable Write enable for low 16 bits, each bit is individual. 1'b0: Write access disable 1'b1: Write access enable
15:2	RO	0x0	reserved
1:0	RW	0x0	rc_type_sel Select RC mode. 2'b00: normal RC mode, every cycle chx_cnt_filter equal to chx_cnt_filter plus ~(chx_cnt_filter>>5) 2'b01: RC atspeed mode, when chx_cnt_filter is less than 'd32, then every cycle chx_cnt_filter equal to (chx_cnt_filter >>1) 2'b10: high speed mode, every cycle chx_cnt_filter equal to (chx_cnt_filter >>1) 2'b11: user define mode, every cycle chx_cnt_filter equal to chx_cnt_filter minus rc_speed_step_cnt

TOUCH DETECT CH RC SPEED STEP CNT

Address: Operational Base + offset (0x0048)

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	rc_speed_step_cnt User define the speed step of RC filter, only used when rc_type_sel == 2'b11.

TOUCH DETECT CH0 CNT

Address: Operational Base + offset (0x0100)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch0_cnt Pulse count value from channel 0.

TOUCH DETECT CH0 CNT DC

Address: Operational Base + offset (0x0104)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch0_cnt_dc Pulse count with offset after low pass filter from channel 0.

TOUCH DETECT CH0 CNT DO

Address: Operational Base + offset (0x0108)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch0_cnt_do Pulse count only with offset after low pass filter from channel 0.

TOUCH DETECT CH0 CNT FILTER

Address: Operational Base + offset (0x010c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch0_cnt_filter Pulse count after RC low pass filter from channel 0.

TOUCH DETECT CH1 CNT

Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch1_cnt Pulse count value from channel 1.

TOUCH DETECT CH1 CNT DC

Address: Operational Base + offset (0x0204)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch1_cnt_dc Pulse count with offset after low pass filter from channel 1

TOUCH DETECT CH1 CNT DO

Address: Operational Base + offset (0x0208)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch1_cnt_do Pulse count only with offset after low pass filter from channel 1.

TOUCH DETECT CH1 CNT FILTER

Address: Operational Base + offset (0x020c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch1_cnt_filter Pulse count after RC low pass filter from channel 1.

TOUCH DETECT CH2 CNT

Address: Operational Base + offset (0x0300)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch2_cnt Pulse count value from channel 2.

TOUCH DETECT CH2 CNT DC

Address: Operational Base + offset (0x0304)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch2_cnt_dc Pulse count with offset after low pass filter from channel 2.

TOUCH DETECT CH2 CNT DO

Address: Operational Base + offset (0x0308)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch2_cnt_do Pulse count only with offset after low pass filter from channel 2.

TOUCH DETECT CH2 CNT FILTER

Address: Operational Base + offset (0x030c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch2_cnt_filter Pulse count after RC low pass filter from channel 2.

TOUCH DETECT CH3 CNT

Address: Operational Base + offset (0x0400)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch3_cnt Pulse count value from channel 3.

TOUCH DETECT CH3 CNT DC

Address: Operational Base + offset (0x0404)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch3_cnt_dc Pulse count with offset after low pass filter from channel 3.

TOUCH DETECT CH3 CNT DO

Address: Operational Base + offset (0x0408)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch3_cnt_do Pulse count only with offset after low pass filter from channel 3.

TOUCH DETECT CH3 CNT FILTER

Address: Operational Base + offset (0x040c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch3_cnt_filter Pulse count after RC low pass filter from channel 3.

TOUCH DETECT CH4 CNT

Address: Operational Base + offset (0x0500)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch4_cnt Pulse count value from channel 4.

TOUCH DETECT CH4 CNT DC

Address: Operational Base + offset (0x0504)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch4_cnt_dc Pulse count with offset after low pass filter from channel 4.

TOUCH DETECT CH4 CNT DO

Address: Operational Base + offset (0x0508)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch4_cnt_do Pulse count only with offset after low pass filter from channel 4.

TOUCH DETECT CH4 CNT FILTER

Address: Operational Base + offset (0x050c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch4_cnt_filter Pulse count after RC low pass filter from channel 4.

TOUCH DETECT CH5 CNT

Address: Operational Base + offset (0x0600)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch5_cnt Pulse count value from channel 5.

TOUCH DETECT CH5 CNT DC

Address: Operational Base + offset (0x0604)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch5_cnt_dc Pulse count with offset after low pass filter from channel 5.

TOUCH DETECT CH5 CNT DO

Address: Operational Base + offset (0x0608)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch5_cnt_do Pulse count only with offset after low pass filter from channel 5.

TOUCH DETECT CH5 CNT FILTER

Address: Operational Base + offset (0x060c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch5_cnt_filter Pulse count after RC low pass filter from channel 5.

TOUCH DETECT CH6 CNT

Address: Operational Base + offset (0x0700)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch6_cnt Pulse count value from channel 6.

TOUCH DETECT CH6 CNT DC

Address: Operational Base + offset (0x0704)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch6_cnt_dc Pulse count with offset after low pass filter from channel 6.

TOUCH DETECT CH6 CNT DO

Address: Operational Base + offset (0x0708)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch6_cnt_do Pulse count only with offset after low pass filter from channel 6.

TOUCH DETECT CH6 CNT FILTER

Address: Operational Base + offset (0x070c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch6_cnt_filter Pulse count after RC low pass filter from channel 6.

TOUCH DETECT CH7 CNT

Address: Operational Base + offset (0x0800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch7_cnt Pulse count value from channel 7.

TOUCH DETECT CH7 CNT DC

Address: Operational Base + offset (0x0804)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch7_cnt_dc Pulse count with offset after low pass filter from channel 7.

TOUCH DETECT CH7 CNT DO

Address: Operational Base + offset (0x0808)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch7_cnt_do Pulse count only with offset after low pass filter from channel 7.

TOUCH DETECT CH7 CNT FILTER

Address: Operational Base + offset (0x080c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch7_cnt_filter Pulse count after RC low pass filter from channel 7.

TOUCH DETECT CH8 CNT

Address: Operational Base + offset (0x0900)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch8_cnt Pulse count value from channel 8.

TOUCH DETECT CH8 CNT DC

Address: Operational Base + offset (0x0904)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch8_cnt_dc Pulse count with offset after low pass filter from channel 8.

TOUCH DETECT CH8 CNT DO

Address: Operational Base + offset (0x0908)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch8_cnt_do Pulse count only with offset after low pass filter from channel 8.

TOUCH DETECT CH8 CNT FILTER

Address: Operational Base + offset (0x090c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch8_cnt_filter Pulse count after RC low pass filter from channel 8.

TOUCH DETECT CH9 CNT

Address: Operational Base + offset (0x1000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch9_cnt Pulse count value from channel 9.

TOUCH DETECT CH9 CNT DC

Address: Operational Base + offset (0x1004)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch9_cnt_dc Pulse count with offset after low pass filter from channel 9.

TOUCH DETECT CH9 CNT DO

Address: Operational Base + offset (0x1008)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch9_cnt_do Pulse count only with offset after low pass filter from channel 9.

TOUCH DETECT CH9 CNT FILTER

Address: Operational Base + offset (0x100c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch9_cnt_filter Pulse count after RC low pass filter from channel 9.

TOUCH DETECT CH10 CNT

Address: Operational Base + offset (0x1100)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch10_cnt Pulse count value from channel 10.

TOUCH DETECT CH10 CNT DC

Address: Operational Base + offset (0x1104)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch10_cnt_dc Pulse count with offset after low pass filter from channel 10.

TOUCH DETECT CH10 CNT DO

Address: Operational Base + offset (0x1108)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch10_cnt_do Pulse count only with offset after low pass filter from channel 10.

TOUCH DETECT CH10 CNT FILTER

Address: Operational Base + offset (0x110c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch10_cnt_filter Pulse count after RC low pass filter from channel 10.

TOUCH DETECT CH11 CNT

Address: Operational Base + offset (0x1200)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch11_cnt Pulse count value from channel 11.

TOUCH DETECT CH11 CNT DC

Address: Operational Base + offset (0x1204)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch11_cnt_dc Pulse count with offset after low pass filter from channel 11.

TOUCH DETECT CH11 CNT DO

Address: Operational Base + offset (0x1208)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch11_cnt_do Pulse count only with offset after low pass filter from channel 11.

TOUCH DETECT CH11 CNT FILTER

Address: Operational Base + offset (0x120c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch11_cnt_filter Pulse count after RC low pass filter from channel 11.

TOUCH DETECT CH12 CNT

Address: Operational Base + offset (0x1300)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch12_cnt Pulse count value from channel 12.

TOUCH DETECT CH12 CNT DC

Address: Operational Base + offset (0x1304)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	ch12_cnt_dc Pulse count with offset after low pass filter from channel 12.

TOUCH DETECT CH12 CNT DO

Address: Operational Base + offset (0x1308)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch12_cnt_do Pulse count only with offset after low pass filter from channel 12.

TOUCH DETECT CH12 CNT FILTER

Address: Operational Base + offset (0x130c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch12_cnt_filter Pulse count after RC low pass filter from channel 12.

TOUCH DETECT CH13 CNT

Address: Operational Base + offset (0x1400)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch13_cnt Pulse count value from channel 13.

TOUCH DETECT CH13 CNT DC

Address: Operational Base + offset (0x1404)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch13_cnt_dc Pulse count with offset after low pass filter from channel 13.

TOUCH DETECT CH13 CNT DO

Address: Operational Base + offset (0x1408)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch13_cnt_do Pulse count only with offset after low pass filter from channel 13.

TOUCH DETECT CH13 CNT FILTER

Address: Operational Base + offset (0x140c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch13_cnt_filter Pulse count after RC low pass filter from channel 13.

TOUCH DETECT CH14 CNT

Address: Operational Base + offset (0x1500)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch14_cnt Pulse count value from channel 14.

TOUCH DETECT CH14 CNT DC

Address: Operational Base + offset (0x1504)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch14_cnt_dc Pulse count with offset after low pass filter from channel 14.

TOUCH DETECT CH14 CNT DO

Address: Operational Base + offset (0x1508)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch14_cnt_do Pulse count only with offset after low pass filter from channel 14.

TOUCH DETECT CH14 CNT FILTER

Address: Operational Base + offset (0x150c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch14_cnt_filter Pulse count after RC low pass filter from channel 14.

TOUCH DETECT CH15 CNT

Address: Operational Base + offset (0x1600)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch15_cnt Pulse count value from channel 15.

TOUCH DETECT CH15 CNT DC

Address: Operational Base + offset (0x1604)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch15_cnt_dc Pulse count with offset after low pass filter from channel 15.

TOUCH DETECT CH15 CNT DO

Address: Operational Base + offset (0x1608)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch15_cnt_do Pulse count only with offset after low pass filter from channel 15.

TOUCH DETECT CH15 CNT FILTER

Address: Operational Base + offset (0x160c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch15_cnt_filter Pulse count after RC low pass filter from channel 15.

TOUCH DETECT CH16 CNT

Address: Operational Base + offset (0x1700)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch16_cnt Pulse count value from channel 16.

TOUCH DETECT CH16 CNT DC

Address: Operational Base + offset (0x1704)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch16_cnt_dc Pulse count with offset after low pass filter from channel 16.

TOUCH DETECT CH16 CNT DO

Address: Operational Base + offset (0x1708)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch16_cnt_do Pulse count only with offset after low pass filter from channel 16.

TOUCH DETECT CH16 CNT FILTER

Address: Operational Base + offset (0x170c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch16_cnt_filter Pulse count after RC low pass filter from channel 16.

TOUCH DETECT CH17 CNT

Address: Operational Base + offset (0x1800)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch17_cnt Pulse count value from channel 17.

TOUCH DETECT CH17 CNT DC

Address: Operational Base + offset (0x1804)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch17_cnt_dc Pulse count with offset after low pass filter from channel 17.

TOUCH DETECT CH17 CNT DO

Address: Operational Base + offset (0x1808)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch17_cnt_do Pulse count only with offset after low pass filter from channel 17.

TOUCH DETECT CH17 CNT FILTER

Address: Operational Base + offset (0x180c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch17_cnt_filter Pulse count after RC low pass filter from channel 17.

TOUCH DETECT CH18 CNT

Address: Operational Base + offset (0x1900)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch18_cnt Pulse count value from channel 18.

TOUCH DETECT CH18 CNT DC

Address: Operational Base + offset (0x1904)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch18_cnt_dc Pulse count with offset after low pass filter from channel 18.

TOUCH DETECT CH18 CNT DO

Address: Operational Base + offset (0x1908)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch18_cnt_dc Pulse count only with offset after low pass filter from channel 18.

TOUCH DETECT CH18 CNT FILTER

Address: Operational Base + offset (0x190c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch18_cnt_filter Pulse count after RC low pass filter from channel 18.

TOUCH DETECT CH19 CNT

Address: Operational Base + offset (0x2000)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch19_cnt Pulse count value from channel 19.

TOUCH DETECT CH19 CNT DC

Address: Operational Base + offset (0x2004)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch19_cnt_dc Pulse count with offset after low pass filter from channel 19.

TOUCH DETECT CH19 CNT DO

Address: Operational Base + offset (0x2008)

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	ch19_cnt_do Pulse count with offset after low pass filter from channel 19.

TOUCH DETECT CH19 CNT FILTER

Address: Operational Base + offset (0x200c)

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	ch19_cnt_filter Pulse count after RC low pass filter from channel 19.

30.5 Interface Description

The interface and IOMUX setting for touch key are shown as follows.

Table 30-1 Touch key Interface Description

Module Pin	Direction	Pin Name	IOMUX Setting
touch out clock m0	O	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GRF_GPIO0B_IOMUX_H[3:0] =4'b0110
touch out clock m1	O	SRADC3/PWM3_M0/UART1_RX_M0/SPI0_MISO_M1/PDM_CLK_S_M1/I2S_MCLK_M0/TKEY4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GRF_GPIO0C_IOMUX_L[15:12] =4'b1000
touch out clock m2	O	PWM3_M1/UART0_RX_M0/SPI0_MISO_M0/I2C1_SCL_M0/TKEY3_M0/TKEY_DRIVE_M2/GPIO0_B7_d	GRF_GPIO0B_IOMUX_H[14:12] =3'b110
touch out clock m3	O	LCD_RS/CIF_HREF/I2C1_SDA_M2/TKEY14/TKEY_DRIVE_M3/PMU_STATE0/AONJTAG_TDI/DSPTAG_TDI/GPIO0_A2_u	GRF_GPIO0A_IOMUX_L[11:7] =4'b0101
touch out clock m4	O	CODEC_CLK_M0/PWM4_M1/TKEY0_M1/UART1_RX_M2/TKEY_DRIVE_M4/I2C0_SDA_M3/PWM_AUDIO_L_M2/GPIO0_D0_u	GRF_GPIO0D_IOMUX_L[2:0] =3'b101
touch out clock m5	O	I2C2_SCL_M0/PWM11/TKEY6_M1/TKEY_DRIVE_M5/I2C1_SCL_M3/I2S_SDI_M1/PWM_AUDIO_R_M3/GPIO0_D6_u	GRF_GPIO0D_IOMUX_H[10:8] =3'b100
touch in0_clock m0	I	PWM0_M1/UART0_CTSN_M0/SPI0_CS0n_M0/I2C0_SDA_M0/TKEY0_M0/TKEY_DRIVE_M0/PWM_AUDIO_L_M0/I2C2_SDA_M1/GPIO0_B4_d	GRF_GPIO0B_IOMUX_H[3:0] =4'b0101
touch in0_clock m1	I	CODEC_CLK_M0/PWM4_M1/TKEY0_M1/UART1_RX_M2/TKEY_DRIVE_M4/I2C0_SDA_M3/PWM_AUDIO_L_M2/GPIO0_D0_u	GRF_GPIO0D_IOMUX_L[2:0] =3'b011
touch in1_clock m0	I	PWM1_M1/UART0_RTSN_M0/SPI0_CLK_M0/I2C0_SCL_M0/TKEY1_M0/PWM_AUDIO_R_M0/I2C2_SCL_M1/GPIO0_B5_d	GRF_GPIO0B_IOMUX_H[6:4] =3'b101
touch in1_clock	I	CODEC_SYNC_M0/PWM5_M1/T	GRF_GPIO0D_IOMUX_L[6:4]

Module Pin	Direction	Pin Name	IOMUX Setting
m1		KEY1_M1/UART1_TX_M2/SPI1_CS0n_M2/I2C0_SCL_M3/PWM_AUDIO_R_M2/GPIO0_D1_u	=3'b011
touch in2 clock m0	I	PWM2_M1/UART0_RX_M0/SPI0_MOSI_M0/I2C1_SDA_M0/TKE_Y2_M0/GPIO0_B6_d	GRF_GPIO0B_IOMUX_H[10:8] =3'b101
touch in2 clock m1	I	CODEC_ADC_D_M0/PWM6_M1/TKEY2_M1/UART2_CTSN_M0/SPI1_CLK_M2/I2S_MCLK_M1/GPIO0_D2_u	GRF_GPIO0D_IOMUX_L[10:8] =3'b011
touch in3 clock m0	I	PWM3_M1/UART0_TX_M0/SPI0_MISO_M0/I2C1_SCL_M0/TKEY3_M0/TKEY_DRIVE_M2/GPIO0_B7_d	GRF_GPIO0B_IOMUX_H[14:12] =3'b101
touch in3 clock m1	I	CODEC_DAC_DL_M0/PWM8/TKEY3_M1/UART2_RTSN_M0/SPI1_MOSI_M2/I2S_SCLK_TX_M1/GPIO0_D3_u	GRF_GPIO0D_IOMUX_L[14:12] =3'b011
touch in4 clock m0	I	SRADC3/PWM3_M0/UART1_TX_M0/SPI0_MISO_M1/PDM_CLK_S_M1/I2S_MCLK_M0/TKEY4_M0/TKEY_DRIVE_M1/GPIO0_C3_d	GRF_GPIO0C_IOMUX_L[15:12] =4'b0111
touch in4 clock m1	I	PMIC_INT_M0/PWM9/TKEY4_M1/UART2_RX_M0/SPI1_MISO_M2/I2S_LRCK_TX_M1/GPIO0_D4_u	GRF_GPIO0D_IOMUX_H[2:0] =3'b011
touch in5 clock m0	I	SRADC4/PWM4_M0/UART0_CTSN_M1/SPI1_CS0n_M0/PDM_CLK_M1/I2S_SCLK_TX_M0/TKEY5_M0/GPIO0_C4_u	GRF_GPIO0C_IOMUX_H[2:0] =3'b111
touch in5 clock m1	I	I2C2_SDA_M0/PWM10/TKEY5_M1/UART2_TX_M0/I2C1_SDA_M3/I2S_SDO0_M1/PWM_AUDIO_L_M3/GPIO0_D5_u	GRF_GPIO0D_IOMUX_H[6:4] =3'b011
touch in6 clock m0	I	SRADC5/PWM5_M0/UART0_RSTN_M1/SPI1_CLK_M0/PDM_SD1_M1/I2S_LRCK_TX_M0/TKEY6_M0/GPIO0_C5_u	GRF_GPIO0C_IOMUX_H[6:4] =3'b111
touch in6 clock m1	I	I2C2_SCL_M0/PWM11/TKEY6_M1/TKEY_DRIVE_M5/I2C1_SCL_M3/I2S_SDI_M1/PWM_AUDIO_R_M3/GPIO0_D6_u	GRF_GPIO0D_IOMUX_H[10:8] =3'b011
touch in7 clock	I	SRADC6/PWM6_M0/UART0_RX_M1/SPI1_MOSI_M0/I2C0_SDA_M1/I2S_SDO_M0/TKEY7/PWM_AUDIO_L_M1/GPIO0_C6_u	GRF_GPIO0C_IOMUX_H[11:8] =4'b0111
touch in8 clock	I	SRADC7/PWM7_M0/UART0_TX_M1/SPI1_MISO_M0/I2C0_SCL_M1/I2S_SDI_M0/TKEY8/PWM_AUDIO_R_M1/PMIC_INT_M1/GPIO0_C7_d	GRF_GPIO0C_IOMUX_H[15:12] =4'b0111
touch in9 clock	I	SRADC0/PWM0_M0/UART1_CTSN_M0/SPI0_CS0n_M1/I2S_SD01_M0/SDMMC_CLKOUT/TKEY	GRF_GPIO0C_IOMUX_L[2:0] =3'b111

Module Pin	Direction	Pin Name	IOMUX Setting
		9/GPIO0_C0_u	
touch in10 clock	I	SRADC1/PWM1_M0/UART1_RT SN_M0/SPI0_CLK_M1/I2C1_SD A_M1/I2S_SCLK_RX_M0/SDMM C_CMD/TKEY10(GPIO0_C1_u)	GRF_GPIO0C_IOMUX_L[7:4] =4'b1000
touch in11 clock	I	SRADC2/PWM2_M0/UART1_RX _M0/SPI0_MOSI_M1/I2C1_SCL _M1/I2S_LRCK_RX_M0/SDMMC _D0/TKEY11(GPIO0_C2_u)	GRF_GPIO0C_IOMUX_L[11:8] =4'b1000
touch in12 clock	I	LCD_D0/CIF_D0/I2C0_SDA_M2 /TKEY12/M4F_WFI/M4F_JTAG _TCK/M0_JTAG_TCK/AONJTAG _TCK/DSPJTAG_TCK GPIO0_A0_u	GRF_GPIO0A_IOMUX_L[3:0] =4'b0100
touch in13 clock	I	LCD_D1/CIF_D1/I2C0_SCL_M2 /TKEY13/M0_WFI/M4F_JTAG_T MS/M0_JTAG_TMS/AONJTAG_T MS/DSPJTAG_TMS GPIO0_A1_u	GRF_GPIO0A_IOMUX_L[7:4] =4'b0100
touch in14 clock	I	LCD_RS/CIF_HREF/I2C1_SDA _M2/TKEY14/TKEY_DRIVE_M3/P MU_STATE0/AONJTAG_TDI/DS PJTAG_TDI GPIO0_A2_u	GRF_GPIO0A_IOMUX_L[11:7] =4'b0100
touch in15 clock	I	LCD_CSn/CIF_VSYNC/I2C1_SC L_M2/TKEY15/PMU_DEBUG/PM U_STATE1/AONJTAG_TDO/DSP JTAG_TDO GPIO0_A3_u	GRF_GPIO0A_IOMUX_L[15:12] =4'b0100
touch in16 clock	I	LCD_RDN/CIF_CLKOUT/UART1 _CTSN_M1/TKEY16/PMU_SLEE P/PMU_STATE2/CODEC_CLK_M 1/AONJTAG_TRSTn/DSPJTAG_T RSTn GPIO0_A4_u	GRF_GPIO0A_IOMUX_H[3:0] =4'b0100
touch in17 clock	I	LCD_WRN/CIF_CLKIN/UART1 _RTSN_M1/PDM_CLK_M0/TKEY1 7/PMU_STATE3/CODEC_SYNC_M 1 GPIO0_A5_d	GRF_GPIO0A_IOMUX_H[7:4] =4'b0101
touch in18 clock	I	LCD_D2/CIF_D2/UART1_RX_M 1/PDM_SDI_M0/TKEY18/PMU STATE4/CODEC_ADC_D_M1/GP IO0_A6_d	GRF_GPIO0A_IOMUX_H[10:8] =3'b101
touch in19 clock	I	LCD_D3/CIF_D3/UART1_TX_M 1/PDM_CLK_S_M0/TKEY19/TE S_CLKOUT/CODEC_DAC_DL_M 1 GPIO0_A7_d	GRF_GPIO0A_IOMUX_H[14:2] =3'b101

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

30.6 Application Notes

30.6.1 Touch key controller usage flow

The touch key controller core operation flow chart below is to describe how the software configures and performs a touch event. Users are strongly advised to follow.

If user wants to use source data to calculate touch event by processor, the usage flow is suggested below.

Set corresponding IOMUX.
 Set corresponding channel enable signal.
 Set drive clock out frequency, suggested frequency is 1KHz to 10Khz, depends on the product.
 Set corresponding posedge_irq enable.
 Set TOUCH_DETECT_CH_IRQ_SEL[0] to 1'b1.
 Set threshold, this may be different according to product.
 Set touchkey controller start bit, this bit must be configured at last.

Waiting for posedge_irq, once receive a interrupt, processor need to get data from touch key controller as soon as possible.
 Processor does some calculation.

Fig. 30-2 Processor Detect mode

If user wants to let touch key controller do some calculation, the usage flow is suggested below.

Set corresponding iomux.
 Set corresponding channel enable signal.
 Set drive clock out frequency, suggested frequency is 1KHz to 10Khz, depends on the product.
 Set corresponding channel posedge_irq and negedge_irq enable
 Set threshold, this may be different according to product.
 Set touchkey controller start bit, this bit must be configured at last.

Normally, once a touch event happens, processor will get a posedge_irq first, and process need to checkout which channel recognizes touch event. And processor also need to record the system time.
 Processor clears corresponding interrupt.
 When the negedge_irq comes out, processor need to checkout which channel triggers and records the system time.
 Processor clears corresponding interrupt.
 Process uses record time to determine there is a long time touch event or a short time touch event.

Fig. 30-3 Touch key controller Detect mode